HYPERSPECTRAL DATA PROCESSING

HYPERSPECTRAL DATA PROCESSING

Algorithm Design and Analysis

Chein-I Chang

University of Maryland, Baltimore County (UMBC), Maryland, USA

WILEY

A JOHN WILEY & SONS, INC., PUBLICATION

Published by John Wiley & Sons, Inc., Hoboken, NJ
Published simultaneously in Canada

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data:

Chang, Chein-I.
Hyperspectral data processing : algorithm design and analysis / Chein-I Chang.
p. cm.
Includes bibliographical references and index.
ISBN 978-0-471-69056-6 (hardback)
1. Image processing–Digital techniques. 2. Spectroscopic imaging. 3. Signal processing. I. Chang, Chein-I. Hyperspectral imaging. II. Title.
TA1637.C4776 2012
621.39’94–dc23
2011043896

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

*This book is dedicated to members of my family, specifically
my mother who provided me with her timeless support
and encouragement during the course of preparing
this book. It is also dedicated to all of my students
who have contributed to this book.*

Contents

Preface

Hyperspectral imaging has witnessed tremendous growth over the past few years. Still its applications to new areas are yet to be explored. Many hyperspectral imaging techniques have been developed and reported in various venues. My first book, *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, referenced as Chang (2003a), was written in an attempt to summarize the research conducted at that time in my laboratory (remote sensing signal and image processing laboratory, RSSIPL) and to provide readers with a peek of this fascinating and exciting area. With rapid advancement in this area many signal processing techniques have been developed for hyperspectral signal and image processing. This book has been written with four goals in mind. One is to continuously explore new statistical signal processing algorithms in this area for various applications. Many results in this book are new, particularly some in Chapters 2, 4, 5–6, 11, 16, 18–19, 23, 24, 29, 30–31, and 33. A second goal is to supplement Chang (2003a), where many potential research efforts were only briefly mentioned (in Chapter 18 of the book). A third goal is to distinguish this book from Chang (2003a) in many ways. Unlike Chang (2003a) where the main theme was hyperspectral target detection and classification from a viewpoint of subpixel and mixed pixel analysis, this book is focused on more in-depth treatment of hyperspectral signal and image processing from a statistical signal processing point of view. A fourth and last goal is to focus on several unsettled but very important issues that have been avoided and never addressed in the past.

One issue is "how many spectral signatures are required to unmix data?" arising in linear hyperspectral unmixing. This has been a long-standing and unresolved issue in remote sensing image processing, specifically hyperspectral imaging, since the number of signatures to be used for data unmixing has a significant impact on image analysis while its accurate number is never known in real applications. Another is "how many pure spectral signatures, referred to as endmembers, are supposed to be present in the data to be processed?" It is common practice to assume that the number of signatures used for spectral unmixing is the same number of endmembers. Unfortunately, such a claim, which has been widely accepted by the community, is not always true in practical applications (see Chapter 17). The issue of endmembers has not received much interest in multispectral image analysis because of its low spectral and spatial resolutions that generally result in mixed data sample vectors. However, due to recent advances in hyperspectral imaging sensors with hundreds of contiguous spectral bands endmember extraction has become increasingly important since endmembers provide crucial "nonliteral" information in spectral interpretation, characterization, and analysis. Interestingly, this issue has never been seriously addressed until recently when it has been investigated by a series of papers (Chang, 2006ab; Chang and

Plaza, 2006; Chang *et al.*, 2006; Plaza and Chang 2006) by introducing a new concept of virtual dimensionality (VD). Besides, some controversial issues result from misinterpreting VD. Therefore, one of the major chapters in this book is Chapter 5, which revisits VD to explore its utility in various applications. Unlike the intrinsic dimensionality (ID), also known as effective dimensionality (ED), which is somewhat abstract and defined as the minimum number of parameters to represent general high-dimensional multivariate data, VD is more practical and realistic. It is defined as the number of "*spectrally*" distinct signatures particularly developed for hyperspectral data in which the non-literal (spectral) information is more crucial and vital than information provided by other dimensions such as spatial information. In particular, an issue arises in how to define the spectral distinction among signatures in VD estimation. Furthermore, unlike ID that is a one-size-fits-all definition for all data sets, VD should adapt to data sets used for different applications as well as vary with the techniques used to estimate VD. In order to address this issue, Chapter 5 explores two types of VD criteria, data characterization-driven criteria and data representation-driven criteria, to define spectrally distinct signatures, and further decouples the concept of VD from the techniques used to estimate VD. Consequently, when VD is poorly estimated by one technique for a particular data set, it is not the definition of VD to be blamed, but rather the technique used for VD estimation that is not applicable to this particular data set. In addition, an issue related to VD is "characterization of pixel information." For example, an anomaly is not necessarily an endmember and vice versa. So, the issues "what is the distinction between these two?" and "how do we characterize these two?" become interesting issues in hyperspectral data exploitation to be discussed in Chapter 18.

Another interesting topic presented in this book is a new concept of "hyperspectral information compression" introduced in Chapters 19–23. It is different from the commonly used so-called hyperspectral data compression in the sense that hyperspectral information compression is generally performed based on the information required to be retained rather than the size of hyperspectral data to be compressed. Therefore, a more appropriate term to be used is "exploitation-based lossy hyperspectral data compression." Nevertheless, it should be noted that the definitions and terminologies used in these chapters are by no means standard.

Finally, an issue of "multispectral imagery versus hyperspectral imagery" is also investigated. It seems that there is no cut-and-dried definition to distinguish these two terminologies. A general understanding of distinguishing these two is that a hyperspectral image is acquired by hundreds of *contiguous* spectral channels/bands with very high spectral resolution, while a multispectral image is collected by tens of *discrete* spectral channels/bands with low spectral resolution. If this interpretation is used, we run into a dilemma, "how many spectral channels/bands are enough for a remotely sensed image to be called a hyperspectral image?" or "how fine the spectral resolution should be for a remote sensing image to be considered as a hyperspectral image?" For example, if we take a small set of hyperspectral band images with spectral resolution 10 nm, say five spectral band images, to form a five-dimensional image cube, do we still consider this new-formed five-dimensional image cube as a hyperspectral image or simply a multispectral image? If we adopt the former definition based on the number of bands, this five-dimensional image cube should be viewed as a multispectral image. On the other hand, if we adopt the latter definition based on spectral resolution, the five-dimensional image cube should be considered as a hyperspectral image. Thus far, it seems that there is no general consensus on this issue. In Chapter 31, an attempt is made to address this issue from a viewpoint of how two versions of independent component analysis (ICA), over-complete ICA, and under-complete ICA can be used to resolve this long-debated issue in the context of linear spectral mixture analysis (LSMA). After all, some of these issues may never be settled or standardized for years to come. Many researchers can always argue differently at their discretion and provide their own versions of interpretation. I have no intention of disputing any of them, but rather respect their opinions.

Since processing hyperspectral signatures as one-dimensional signals and processing hyperspectral images as three-dimensional image cubes are rather different, this book makes a distinction by treating hyperspectral image processing and hyperspectral signal processing in two separate categories to avoid confusion. To this end, three categories are specifically outlined in this book: Category A: hyperspectral image processing; Category B: hyperspectral signal processing; and Category C: applications.

For better understanding, a set of six chapters is included in PART I as preliminaries that cover fundamentals and provide a basic background required for readers to follow algorithm design and development discussed in this book. Category A is made up of 15 chapters (Chapters 7–23) treated separately in four different parts, Part II to Part V. Category B consists of six chapters (Chapters 24–29) in two separate parts, Part VI and Part VII. Finally, applications make up Category C.

It is worth noting that many materials presented in this book have been only available after Chang (2003a). Theses include endmember extraction (Chapters 7–11), algorithm design using different levels of information (supervised linear hyperspectral mixture analysis in Chapters 12–15), pixel characterization and analysis (unsupervised hyperspectral analysis in Chapters 16–18), exploitation-based hyperspectral information compression (Chapters 19–23), hyperspectral signature coding and characterization (Chapters 24–29), and applications (Chapters 30–32) in Category C.

There are three unique features in this book that cannot be found in Chang (2003a): (1) Part I: preliminaries (Chapters 2–6); (2) extensive studies of synthetic image-based experiments for performance evaluation; and (3) an appendix on algorithm compendium that compiles recently developed signal processing algorithms developed in the RSSIPL, all of which are believed to be useful and beneficial to those who design and develop algorithms for hyperspectral signal/image processing. Because this book also addresses many issues that were not explored in Chang (2003a), it can be used in conjunction with Chang (2003a) without much overlap, where the latter provides necessary basic background in design and development of statistical signal processing algorithms for hyperspectral image analysis, especially for subpixel detection and mixed pixel classification. Therefore, on one end, those who have been involved in hyperspectral imaging and are familiar with hyperspectral imaging techniques will find this book useful as reference material. On the other end, those who are new will find this book a good and valuable guide on the topics that may interest them.

I would like to thank the Spectral Information Technology Applications Center (SITAC) that provides its HYDICE data to be used for experiments in this book. I would also like to acknowledge the use of Purdue's Indiana Indian Pine test site and the AVIRIS Cuprite image data website.

I owe my sincere gratitude and deepest appreciation to my former Ph.D. students, Drs. Sumit Chakravarty, Hsian-Min Chen, Yingzi Du, Qian Du, Mingkai Hsueh, Baohoing Ji, Xiaoli Jiao, Keng-Hao Liu, Weimin Liu, Bharath Ramakrishna, Hsuan Ren, Haleh Safavi, Chiun-Mu Wang, Jianwei Wang, Jing Wang, Su Wang, Englin Wong, Chao-Cheng Wu, Wei Xiong, and MS student, Ms. Farzeen Chaudhary as well as my current Ph.D. student, Shih-Yu Chen. My appreciation is also extended to my colleagues, Professor Chinsu Lin with the Department of Forestry and Natural Resources at National Chiayi University, Dr. Ching Wen Yang who is the Director of Computer Center, Taichung Veterans General Hospital, and Professor Ching Tsorng Tsai with the Computer Science Department at Tunghai University. I would like to thank particularly my former Ph.D. students, Dr. Chao-Cheng Wu who carried out most of the experiments presented in Chapters 7–11, Dr. Ken-Hao Liu who performed many experiments described in Chapters 21–23, Dr. Su Wang who did all the work mentioned in Chapter 29, Dr. Englin Wong who performed all the experiments described in Chapter 32, and Professor Antonio J. Plaza who contributed to some part of Chapter 18 when he was on sabbatical leave from the Computer Science Department, University of Extremadura, Spain, in 2004 to visit my laboratory. This book could not have been completed without their contributions.

I would also like to thank the Ministry of Education in Taiwan for supporting me as a Distinguished Lecture Chair within the Department of Electrical Engineering from 2005 to 2006, a Chair Professorship of Reduction Technology within the Environmental Restoration and Disaster Reduction Research Center and Department of Electrical Engineering from 2006 to 2009, and a Chair Professorship of Remote Sensing Technology within the Department of Electrical Engineering from 2009 to 2012, at National Chung Hsing University where Professor Yen-Chieh Ouyang of Electrical Engineering has been a very supportive host during my visit. In particular, during the period 2009–2010, I was on sabbatical leave from UMBC to visit National Chung Hsing University where my appointment as a distinguished visiting fellow/fellow professor was supported and funded by the National Science Council in Taiwan under projects of NSC 98-2811-E-005-024 and NSC 98-2221-E-005-096. All their support is highly appreciated.

Last but not least, I would also like to thank my friends, Dr. San-Kan Lee (Deputy Superintendent of Taichung Veterans General Hospital (TCVGH)), Dr. Clayton Chi-Chang Chen (Chairman of Radiology at TCVGH), Dr. Jyh-Wen Chai (Section Chief of Radiology at TCVGH), and Dr. Yong Kie Wong (Head of Dental Department at TCVGH) who have selflessly provided their expertise and resources, especially an excellent testbed environment to help me use hyperspectral imaging techniques in magnetic resonance imaging (MRI). Chapter 32 is indeed a culmination of such a great working relationship.

As a final note, I would like to share that this book was supposed to be delivered by 2008. The most important factor that caused the delay is the urge to include the latest reports on hyperspectral data analysis. It is very difficult and challenging to keep a track of such new developments. Nevertheless, this book has grown three times larger than what I had originally proposed. Those who are interested in my forthcoming 2013 book can have a quick peek of these topics briefly discussed in Chapter 33, which includes a new development of target-characterized virtual dimensionality (VD), real-time and progressive processing of endmember extraction, unsupervised target detection, anomaly detection, as well as their field programmable gate array (FPGA) implementation.

CHEIN-I CHANG (張建禕)

Professor of Electrical Engineering
Remote Sensing Signal and Image Processing Laboratory (RSSIPL)
University of Maryland, Baltimore County
Baltimore, Maryland
USA

Chair Professor of Remote Sensing Technology
National Chung Hsing University
Taichung, Taiwan
Republic of China

Distinguished Professor
Providence University
Taichung, Taiwan
Republic of China

International Chair Professor
National Taipei University of Technology
Taipei, Taiwan
Republic of China

Technical Advisor
Center for QUantitative Imaging in Medicine (*CQUIM*)
Taichung Veterans General Hospital
Taichung, Taiwan
Republic of China
Fall 2012

1

Overview and Introduction

The past few years have witnessed tremendous advances in hyperspectral imaging where statistical signal processing has played a pivotal role in driving algorithm design and development for hyperspectral data exploitation. It has attracted attention of those who come from different disciplinary areas by exploring new applications and making connections between remote sensing and other engineering fields. In recent years, there has been a significant increase in participation in various conferences and venues related to this area, which in turn has provided evidence that hyperspectral signal and image processing has broken away from traditional spatial domain analysis–based remote sensing and successfully branched out to stand alone as a single research topic, similar to signal processing that evolved as a separate area from communications in the late 1970s. On the contrary issues related to high spectral resolution provided by hyperspectral imaging sensors have also changed the ways in which algorithms are designed and developed. As a consequence, many problems such as subpixels and mixed pixels that are generally encountered in hyperspectral image processing, but do not occur in classical two-dimensional (2D) image processing, have become major issues for traditional spatial domain-based techniques. This is because the concept of "*seeing-is-believing*" by visual inspection, which has been widely used in image processing, cannot resolve issues of targets that are completely embedded in a single pixel or partially but do not fully occupy a single pixel, in which case only spectral properties can be used to characterize such targets for data analysis. Therefore, to distinguish such spectral characterization-based analysis from the traditional spatial domain–based analysis, the former is referred to as *nonliteral* analysis as opposed to the latter termed as *literal* analysis.

Due to complicated environments in real-world problems many uncontrollable parameters are also beyond our grip. In order to explore insights into algorithm design, the use of synthetic images to simulate various scenarios to substantiate designed algorithms for performance analysis becomes an effective proof-of-concept evaluation tool. Such synthetic images can be simulated by either real image scenes or laboratory data sets for various applications. Unfortunately, such synthetic image-based computer simulations have received little attention in the past. Accordingly, one of the major features that readers will find in this book is an extensive use of synthetic image-based experiments in algorithm design and analysis for qualitative as well as quantitative performance evaluation. Another unique feature of this book is that the algorithms derived and developed in this book can be implemented with little difficulty via the MATLAB, a widely accepted software package developed by the MATHWORK for engineering applications. This advantage allows readers to implement their algorithms. To further facilitate this benefit MATLAB codes of many

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

popular algorithms developed in this book are also made available in the appendix at the end of this book. Most importantly, this book has also expanded its use of images in Chang (2003a) to include two more popular image scenes, Purdue's Indian Pine test site in Indiana and Cuprite image scene in Nevada, both of which are available on web site so that they can be used by those who develop new algorithms, to validate and evaluate their designed algorithm for performance analysis as well as to conduct their own comparative study. Last but not the least, this book includes an appendix that compiles many algorithms developed in the Remote Sensing Signal and Image Processing Laboratory (RSSIPL) at the University of Maryland, Baltimore County (UMBC). Such an algorithm compendium should serve as a valuable guide for people who are interested in applications of hyperspectral data processing.

1.1 Overview

Hyperspectral signal and image processing has become a fast-growing area that bridges communities of remote sensing and signal/image processing due to the fact that many problems arising in the former can be reformatted and solved by the latter. A good example is an increasing number of conferences in both communities and a wide range of publications in both signal and image processing journals and traditional remote sensing journals. Particularly, in recent years significant research and development in hyperspectral imaging has resulted in at least hundreds, if not thousands, of publications in various journals. Many new findings have been reported in many annual meetings and venues such as IEEE International Geoscience and Remote Sensing Symposia; SPIE conferences on Defense and Security (previously known as AeroSense); Algorithms and Technologies for Multispectral, Hyperspectral and Ultraspectral Imagery (annually held in April); SPIE International Symposium on Optical Science & Technology (*Remote Sensing Symposium*, specifically Conferences on *Satellite Data Compression, Communication, and Processing* and Conferences on *Imaging Spectrometry*, annually held in August); Conference on Imaging Spectrometry; EOS/SPIE Symposium on Remote Sensing; IEEE GRSS Workshop on Hyperspectral Image and Signal Processing–Evolution in Remote Sensing (WHISPERS); and so forth. Accordingly, any attempt to cover in a single book all possible areas in this field would be impossible and unrealistic. Keeping this reality in mind the book is written based on personal preference and is only focused on recent works that have been done mostly in RSSIPL at UMBC, but have not been covered in my previous book (Chang, 2003a). Therefore, there is not much overlap between this book and Chang (2003a). Hence this book can be considered as a sequel of Chang (2003a). Specifically, this book takes a rather different yet unique approach compared with that adopted in Chang (2003a), by treating hyperspectral image processing and hyperspectral signal processing as two separate subjects where the former processes a hyperspectral image as an image cube and the latter considers a hyperspectral signature as a one-dimensional signal so that no sample correlation such as spectral correlation among pixels in a hyperspectral image cube can be taken into account and used for algorithm design. Within this context the topics presented in this book are organized in the following order.

I. Preliminaries
A. Hyperspectral Image Processing
II. Endmember Extraction
III. Supervised Linear Hyperspectral Mixture Analysis
IV. Unsupervised Hyperspectral Image Analysis
V. Hyperspectral Information Compression
B. Hyperspectral Signal Processing

VI. Hyperspectral Signal Coding
VII. Hyperspectral Signal Characterization
C. Applications
Appendix: Algorithm Compendium

Several recent books on hyperspectral imaging are also available in the public domain (Chang. 2003a; 2006b; 2007a; Plaza and Chang, 2007a; Varshney and Arora, 2004) and the subjects covered in these books are somewhat selective. For example, the book by Varshney and Arora (2004) is focused on some specific techniques, for example, independent component analysis, support vector machines, and Markov random field. The book by Chang (2003a) is primarily devoted to nonliteral statistical signal processing techniques developed for subpixel detection and mixed pixel classification. The two books edited by Chang (2006b) and (2007a) are collections of most recent results contributed by researchers who are experts and currently active in hyperspectral imaging. Another book edited by Plaza and Chang (2007a) intends to target specific topics in high-performance computing, which has grown rapidly due to the need of processing enormous data volumes and has also become increasingly important in remote sensing data processing. Unlike the above-mentioned books this book explores many interesting research topics resulting from issues that are generally either overlooked or neglected in multispectral imagery as well as issues that were not addressed or fully explored in Chang (2003a).

1.2 Issues of Multispectral and Hyperspectral Imageries

Because of its low spectral resolution a multispectral image pixel vector usually does not have information as rich as a hyperspectral image pixel vector does. In this case, multispectral image processing must rely on image spatial information and correlation to make up insufficient spectral information resulting from a few discrete spectral bands. Therefore, an early development of multispectral image processing has focused on spatial domain-based techniques. However, with recent advent of very high-spectral resolution hyperspectral imaging sensors many material substances that cannot be resolved by multispectral imaging sensors can now be uncovered by hyperspectral imagers for data analysis. As a consequence, targets or objects of interest for multispectral and hyperspectral image analyses are quite different. In multispectral image analysis land cover/land use is often of major interest. Therefore, the developed techniques generally perform pattern classification and analysis in the sense that every single pixel of an image must be classified into one of a number of pattern classes, each of which corresponds to one particular spatial class. On the contrary, the objects of interest in hyperspectral image analysis are usually targets with particular spectral characteristics such as man-made targets, anomalies, or rare targets. The targets of these types generally appear either in a form mixed with a number of material substances or at subpixel level with targets embedded in a single pixel vector due to their size that is smaller than the ground sampling distance (GSD). Besides, these types of targets usually appear unexpectedly and their probabilities of occurrence are also low. Most importantly, their sample pool may also be relatively small and their sizes may only have limited spatial extent. As a consequence, such targets may not be easy to be visually identified or inspected with prior knowledge; thus, they can be considered as insignificant targets but are indeed of major interest from an intelligence or information point of view. For example, these targets may include special spices in agriculture and ecology, toxic wastes in environmental monitoring, rare minerals in geology, drug/smuggler trafficking in law enforcement, military vehicles in combat, abnormality in battlefields, landmines in war zones, chemical/biological agents in bioterrorism, weapon concealment and mass graves in intelligence gathering, and so on. Under such circumstances, they can be only detected at mixed or subpixel level and traditional spatial domain (i.e., literal)-based image processing techniques may not be suitable or effective. So, the extraction of such targets must rely on their spectral profiles and the techniques developed for hyperspectral image analysis should perform *target*-based detection, discrimination, classification, identification, recognition, and quantification as opposed to *pattern*-based multispectral imaging techniques. As a result, a direct extension of multispectral imaging techniques to hyperspectral imagery may not be effective in hyperspectral data exploitation because pattern class information and correlation provided by these targets may be too little to be used for performing hyperspectral image analysis. In order to address this issue the techniques in Chang (2003a) are developed directly from a hyperspectral imagery point of view for spectral detection and classification. This book expands the scope of Chang (2003a) to cover a wider range of applications including endmember extraction, unsupervised target detection, information compression, and hyperspectral signal coding and characterization, none of which is studied in Chang (2003a).

1.3 Divergence of Hyperspectral Imagery from Multispectral Imagery

The hyperspectral imagery has changed the way we think of multispectral imagery. This is because we now have hundreds of contiguous spectral bands available at our disposal. So, one major issue is how to effectively use and take advantage of spectral information provided by these hundreds of spectral bands for various applications in data exploitation, for example, target detection, discrimination, classification, quantification, and identification. This interesting issue can be addressed by the following two interesting examples. The first example uses real-to-complex analysis to illustrate why it is inappropriate to simply extend multispectral imaging techniques to process hyperspectral imagery. The second example uses the well-known pigeon-hole principle in discrete mathematics (Epp, 1995) to illustrate how hyperspectral imagery can be addressed by a rationale completely different from that used for multispectral imagery.

1.3.1 Misconception: Hyperspectral Imaging is a Natural Extension of Multispectral Imaging

While dealing with hyperspectral imagery there is a general consensus that hyperspectral imagery is a natural extension of multispectral imagery based on an assumption that a hyperspectral image has more spectral bands for data collection than a multispectral image does. As a result, it may lead to a misconception that hyperspectral imaging problems can be solved by multispectral imaging techniques by simply taking advantage of its expanded spectral bands. A similar misconception also occurs in hyperspectral data compression where researchers in data compression community consider a hyperspectral data as an image cube so that 3D image compression processing techniques developed for videos can be simply applied to hyperspectral imagery as a 3D image cube without extra precaution (see Part V: Chapters 19–23). Unfortunately, over the past few years these misconceptions have somewhat directed the way we design and develop hyperspectral imaging techniques.

To understand the fundamental difference between multispectral imaging and hyperspectral imaging, we use a simple mathematical example to illustrate a similar misconception, which is finding derivatives in real analysis and complex analysis. Since real variables can be considered as real parts of complex variables, this may lead to a brief that real analysis is a special case of complex analysis, which is certainly not true. One piece of clear evidence is derivatives. When a derivative is calculated in the real line, the direction with respect to which a derivative is calculated along the real axis is constrained either to the left or to the right. However, the direction along which a derivative is calculated by complex analysis can be along any curve in the complex plane. As a result, calculating a complex derivative is more sophisticated than simply extending the way derivatives are calculated in real analysis. A natural extension of real derivatives is partial derivatives in complex analysis along two axes: x-axis and y-axis. However, it is not true for any derivative calculated in the complex plane. This is because the direction along which the derivative is calculated is not only limited to x- and y-axes but it must also take into account all directions that are more likely curves instead of lines. When such a derivative occurs it is called total differentiable or analytic and must satisfy the so-called Cauchy–Riemann equation that allows a differentiable complex variable to be expanded as a power series which is much stronger than only derivatives. This simple example explains why complex analysis is not a natural extension of real analysis and a direct extension of real derivatives to complex derivatives as partial derivatives can only achieve limited success to some extent. This example sheds some light on a similar key difference between multispectral and hyperspectral images. In its early days multispectral imagery has been used in remote sensing mainly for land cover/land use classification in agriculture, disaster assessment and management, ecology, environmental monitoring, geology, geographical information system (GIS), and so on. In these applications, low spectral resolution multispectral imagery may provide sufficient information for data analysis and the techniques developed for multispectral image processing are primarily based on pattern classes that take advantage of spatial correlation to perform various tasks. Compared to multispectral imagery, hyperspectral imagery utilizes hundreds of contiguous spectral bands to perform target-class analysis. This is the major difference between hyperspectral imagery and multispectral imagery. Specifically, the objects of interest in hyperspectral imagery are no longer patterns of large areas as considered in multispectral imagery. Instead, hyperspectral image analysts are interested in those objects that cannot be visualized by inspection or with prior knowledge due to limited extent of their spatial presence. As a result, hyperspectral imaging is generally developed to perform target class–based image analysis where image background is usually of no interest. Such examples include anomaly detection, endmember extraction, man-made target detection, and so on, where the spatial information provided by these objects of interest is generally very little. So, if the hyperspectral imagery is treated as a natural extension of the multispectral imagery, its success can be very limited due to its use of spatial information to perform *pattern class–based image analysis* rather than *target class–based image analysis*, a similar dilemma that also occurs between real and complex derivatives. Accordingly, we must reinvent the wheel and re-design and develop new hyperspectral imaging techniques rather than directly derive those adopted from multispectral image techniques. One promising approach is the use of the following pigeon-hole principle described in the following section.

1.3.2 Pigeon-Hole Principle: Natural Interpretation of Hyperspectral Imaging

Suppose that there are p pigeons flying into L pigeon holes (nests) with $L < p$. According to the pigeon-hole principle, there exists at least one pigeon hole that must accommodate at least two or more pigeons. Now, assume that L is the total number of spectral bands and p is the number of targets of interest. By virtue of the pigeon-hole principle, we can interpret a pigeon hole as a spectral band while a pigeon is considered as a target (or an object) of interest. With this interpretation if $L > p$, a spectral band can be used to detect, discriminate, and classify a distinct target. Since there are hundreds of spectral bands available from hyperspectral imagery, technically speaking, hundreds of spectrally distinct targets can be accommodated by these spectral bands, namely one target by one particular spectral band. In order to materialize this idea, three issues need to be addressed. First, the number of spectral bands, L, must be greater than or equal to the number of targets of interest, p, that is, $L \geq p$. This seems always true for hyperspectral imagery, but is not necessarily valid for multispectral imagery, where $L < p$ in the latter is usually true. For example, 3-band SPOT multispectral data may have difficulty with classifying more than three target substances present in the data using the pigeon-hole principle. However, the benefit of $L \geq p$ also gives rise to a challenging issue known as "curse of dimensionality" (Duda and Hart, 1973), that is, "what is the true value of p if $L \geq p$." This has been a long-standing issue for any hyperspectral image analyst to resolve because it is nearly impossible to know the exact value of p in real-world problems. Moreover, even if the value of p can be provided by prior knowledge it may not be reliable due to many unexpected factors that cannot be known *a priori*. In multivariate data analysis, the value of p is described by the so-called intrinsic dimensionality (ID) (Fukunaga, 1990), which is defined as the minimum number of parameters used to specify the data. However, this concept is only of theoretical interest. No method has been proposed regarding how to find it in the literature. A common strategy is to estimate the p on a trial-and-error basis. A similar problem is also encountered in passive array processing where the number of signal sources, p, arriving at a linear array of sensors is of major interest. In order to estimate this number, two criteria, an information criterion (AIC) suggested by Akaike (1974) and minimum description length developed by Schwarz (1978) and Rissanen (1978), have been widely used to estimate the value of p. Unfortunately, a key assumption made on these criteria is that the noise must be independent and identically distributed, a fact that is usually not a valid assumption in hyperspectral images as shown in Chang (2003a) and Chang and Du (2004). In order to cope with this dilemma, a new concept called virtual dimensionality (VD) was coined and suggested by Chang (2003a) to estimate the number of spectrally distinct signatures in hyperspectral imagery. It is also based on the pigeon-hole principle where VD is used to estimate the number of pigeons with the total number of spectral bands interpreted as the number of pigeon holes. The last issue to be addressed is that once a spectral band is being used to accommodate one target, it cannot be used again to accommodate another distinct target. One way to do so is to perform orthogonal subspace projection (OSP) developed by Harsanyi and Chang (1994) on a space linearly spanned by the already found targets to find an orthogonal complement space from which only new targets can be generated. Equivalently speaking, the spectral bands used to accommodate previous targets cannot be used again to accommodate a new target. Through a series of such OSP operations no two distinct targets will be specified and accommodated by a single spectral band. In other words, all the found targets must be accommodated in separate mutual orthogonal subspaces. In terms of the pigeon-hole principle it implies that no two pigeons will be allowed to fly into a single pigeon hole. Here, one remark is noteworthy. When it says that one target is accommodated and specified by one spectral band, it simply means that the target can be best spectrally characterized by this particular band compared to other bands. So, this band is chosen to be its identity like its fingerprint or DNA. If two targets happen to have the same band being used for their best spectral characterization then there is no way to discriminate one from the other. In this case, it implies that two pigeons fly into the same pigeon hole. More specifically, one pigeon hole is used to accommodate two flying-in pigeons, both of which reside in a single pigeon hole.

Once these three issues, that is, (1) $L \geq p$, (2) determination of p, and (3) no two distinct target signatures to be accommodated by a single spectral band, are resolved, the idea of applying the pigeon-hole principle to hyperspectral data exploitation can be realized and becomes feasible. More specifically, using spectral bands as a means to perform detection, discrimination, classification, and identification without accounting for spatial information or correlation provides an alternative approach, to be called nonliteral analysis as opposed to the spatial domain-based approach, to be called literal analysis. Such a nonliteral analysis is particularly important for two types of targets. One is that targets are small or insignificant due to their limited spatial presence and cannot be effectively captured by spatial correlation or information. The other is that targets of the same type are spatially separated so that their spatial correlation is actually very weak and little in which case the spatial domain-based literal analysis may have difficulty in finding them spatially correlated. The only way to group them together is based on their spectral characteristics regardless of where they are spatially located.

Interestingly, the pigeon-hole principle also sheds light on differentiation of hyperspectral imagery from multispectral imagery. Through the relationship between the total number of spectral bands, L, and the number of signal sources to be accommodated, p, discussed above, a multiple-band remote sensing image can be considered as a hyperspectral image if $L \geq p$ and a multispectral image otherwise (i.e., $L < p$). More details of this interpretation can be found in Chapter 31.

Furthermore, VD can be also interpreted by the pigeon-hole principle, and its potential in hyperspectral data exploitation has been demonstrated in many applications, for example, linear spectral mixture analysis (Chang, 2006c), dimensionality reduction (Wang and Chang, 2006a, 2006b), band selection (Chang and Wang, 2006), and so on. Chapter 5 will revisit VD for more details.

1.4 Scope of This Book

While writing this book it is important to consider hyperspectral image processing and hyperspectral signal processing as two different research areas and treat them separately. When hyperspectral data are processed as image cubes, it is called hyperspectral image processing where data samples are image pixel vectors and both spectral and spatial correlation among image pixel vectors can be made available for data processing. On the other hand, when hyperspectral data are processed as signatures it is called hyperspectral signal processing where a signature is a one-dimensional signal, which represents its spectral profile over a range of wavelengths for signature characterization. In this case, only interband spectral correlation within the signature is available for data processing and no other information such as sample spatial or spectral correlation used in hyperspectral image processing is available for signature processing. Such hyperspectral signals include data obtained from laboratories, databases, and spectral libraries where no data sample spatial/spectral correlation is available. Therefore, techniques developed for hyperspectral image processing may not be directly applicable to hyperspectral signal processing and vice versa. Unfortunately, it seems that there is no concern in distinguishing one from another when it comes to algorithm design. This book is believed to be the first to do so by treating hyperspectral image processing and hyperspectral signal processing in two separate categories: Category A: Hyperspectral Image Processing treated in Parts II–V; Category B: Hyperspectral Signal Processing treated in Parts VI–VII.

In order to make this book self-contained, preliminaries are also included as Part I to cover basic knowledge that provides readers with necessary background required to read this book. In particular, it integrates many scattering results into different chapters so that readers can follow through materials easily without looking for other references. Each chapter in Part I can be read independently with little interruption while also keeping the flow and all the chapters coherent each other.

Part II is endmember extraction that is one of most crucial tasks in hyperspectral data exploitation and has recently become increasingly important due to significantly improved high spatial and spectral resolution provided by hyperspectral imaging sensors. According to the definition given by Schowengerdt (1997), an endmember is an idealized, pure signature for a class, more specifically, spectral class. For multispectral imagery, an endmember may nowhere be found since most data sample vectors may be heavily mixed due to low spatial and spectral resolution. As a result, the importance of endmember extraction has been overlooked and not been a major subject in multispectral image analysis. By contrast, with recent advances of hyperspectral imaging sensors many subtle material substances that cannot be resolved by multispectral imagery can be now revealed by hyperspectral imagery. These substances are generally not known *a priori* and can be only diagnosed by high spectral resolution. Endmembers are considered to be one of such substances. In general, their existence in image data cannot be detected visually. Most importantly, once endmembers are present, their spatial extent is relatively limited. Besides, their sample pools are also very small. Accordingly, they may appear as anomalies. In this case, spatial characteristics offer little advantage in finding endmembers. In the past, the image classification in multispectral image processing has been often performed by pattern classification (land use/land cover classification) where each image pixel must be classified into a particular class in accordance with a certain classification criterion. However, endmembers are generally rare. Unless they are treated and extracted as targets of interest, their detection and extraction is very challenging. Additionally, because of the lack of spatial patterns specified by endmembers the effectiveness of endmember detection or extraction will be very likely to be compromised by spatial-based pattern classification techniques. In order to address this issue, Chang (2003a) has focused on target classification than on pattern classification, in which case only targets of interest are of major concern where image background is only used for suppression. However, such an important issue of endmember extraction was not investigated and explored in Chang (2003a), when this subject was not mature but now will be one of the major subjects in this book studied in great detail in Part II.

Part III revisits supervised linear spectral mixture analysis (SLSMA), which was discussed in great length in Chang (2003a). This part rederives a least squares-based orthogonal subspace projection (LSOSP) from the signal-to-noise ratio (SNR)-based orthogonal subspace projection so that LSOSP and OSP essentially operate the same matched filter subject to a constant κ, which accounts for least squares estimation error. More specifically, LSOSP performs as an estimator by including the κ, while OSP operates as a detector by setting $\kappa = 1$. By using different matched signatures LSOSP and OSP can interpret many commonly used operators such as constrained energy minimization (CEM) in Chang (2003a) and RX detector developed by Reed and Yu (1990). Furthermore, OSP and LSOSP can be extended in three different directions. One is to replace the least squares error criterion with Fisher's ratio to derive Fisher's LSMA (FLSMA). Another is to impose weight constraints on spectral bands to derive weighted abundance-constrained LSMA (WAC-LSMA). Finally, a third direction introduces a nonlinear kernel into LSMA to derive kernel-based LSMNA (KLSMA).

Part IV extends SLSMA developed in Part III to unsupervised LSMA (ULSMA) where prior knowledge of signature information is not available. Under such circumstance two major issues that do not occur in supervised analysis need to be addressed. One is how many signature sources of interest to be used for LSMA and the second is how to find them. Once these issues are resolved ULSMA becomes SLSMA where approaches presented in Part III are readily applied.

Part V is hyperspectral information compression. One challenging issue in processing hyperspectral imagery is its huge data volume, which may result in high computational cost of data processing, long delay of data transmission and communications, and difficult management of data storage and archiving. Another is how to compress spectral information resulting from highly correlated spectral bands without sacrificing vital information. The first issue can be addressed by developing techniques reducing data size, referred to as data reduction/compression, while the second issue can only be addressed by developing techniques removing redundant information, referred to as information compression. These two types of compression are completely different and should be dealt with separately. Unfortunately, many hyperspectral data compression techniques have not taken this distinction into account. But, it is important to differentiate information compression from data compression since the compression ratio used in data compression is measured by data size, which does not imply compression of information. In other words, information compression is determined by various applications with specific information required to be retained during a compression process. This type of information compression can be considered as exploitation-based lossy compression. To address this issue, the commonly used terminology, hyperspectral data compression, is referred to as hyperspectral information compression in this book and can be interpreted as exploitation-based lossy hyperspectral compression, which includes two major spectral compression techniques, spectral dimensionality reduction and spectral band selection, each of which will be discussed in great detail in Part V.

Up to now the hyperspectral data considered in previous parts are image cubes where all the data sample vectors are image pixel vectors. However, in many situations the hyperspectral data may only be obtained as signature vectors by nonimage sensors or from spectral libraries or databases. In this case, the data to be dealt with is a one-dimensional hyperspectral signal as a signature vector rather than as a pixel vector in three-dimensional image cube. So, Category B in this book is primarily focused on hyperspectral signal processing, which consists of two parts, Part VI and Part VII. Part VI considers hyperspectral signal coding where information compression is performed on a hyperspectral signature vector to capture its unique spectral profile to serve its fingerprint for signature discrimination, detection, classification, and identification. In other words, instead of considering image data as a 3D image cube, the idea of hyperspectral signal coding is to explore spectral characteristics and further to capture changes in the spectral profile within a single signature vector as spectral marks so that a single signature vector can be encoded by its fingerprint as a code word to represent its identity. On the other hand, hyperspectral signal coding can also be considered as quantization that discretizes hundreds of spectral values into a finite set of discrete values. So, it can be viewed as an analog-to-digital (A/D) converter and intends to find the best possible representation of a hyperspectral signature vector for a given bit rate. For a multispectral signature vector the spectral resolution is low and only a few spectral values are available for quantization. So, signature coding may not be effective to characterize spectral signature properties. This may no longer be true for a hyperspectral signature vector where hundreds of contiguous spectral bands may provide sufficient information for spectral characterization. Interestingly, hyperspectral signature coding has never been of major interest in hyperspectral data analysis. This part investigates three types of hyperspectral signal coding: binary coding, vector coding, and progressive coding, where the binary coding can be viewed as memoryless coding as opposed to the vector coding and progressive coding, which can be regarded as memory coding. Comparing the hyperspectral signature coding in Part VI that makes hard decisions on the spectral profile of a signature vector, Part VII presents techniques that make soft decisions on a signature vector to perform hyperspectral signature analysis in the sense of hyperspectral signature characterization. In this case, the knowledge of a reference signature is generally required for a hyperspectral signature vector to be characterized. Unfortunately, hyperspectral signature analysis via spectral characterization has not received much attention in the last few years. Part VII investigates this issue by developing three different approaches: band selection for signature characterization, Kalman filter for signature estimation, and wavelets for signature representation.

The last category of this book is Category C: Applications, which show how hyperspectral imaging techniques can be applied to various problems such as size estimation of subpixel targets, concealed target detection, and how to take advantage of hyperspectral imaging techniques to resolve issues of multispectral imagery. Specifically, a new application of hyperspectral imaging to magnetic resonance imaging is included to demonstrate its utility in medical imaging.

To conclude this book, an appendix is also included for readers' reference. It is an algorithm compendium that compiles important algorithms developed in the RSSIPL at UMBC.

1.5 Book's Organization

This book is organized in accordance with the order laid out by seven parts in three categories presented in the previous section. Each part can be read independently while keeping sufficient correlation with other parts.

1.5.1 Part I: Preliminaries

The preliminaries in Part I help readers grasp sufficient knowledge to follow this book without difficulty. It consists of six chapters.

Chapter 2 is Fundamentals of Subsample and Mixed Sample Analyses. It uses a simple example to illustrate issues of subsamples and mixed samples encountered in detection and classification. It then walks through various approaches using hard and soft decisions for subsample detection and mixed sample classification. It includes many techniques currently being used and available in the literature.

Chapter 3 introduces Three-Dimensional Receiver Operating Characteristics (3D ROC) Analysis that can be used as an evaluation tool for soft decision-making performance for hyperspectral target detection and classification. An ROC curve is defined as a curve plotted based on detection probability versus false alarm probability. An analysis that uses ROC curves to evaluate the effectiveness of a Neyman–Pearson detector is called ROC analysis. A major advantage of ROC analysis is that there is no need of specifying a particular cost function. For example, least squares error or signal-to-noise ratio may be a good criterion for detection of problems in signal processing and communications, but may not be appropriate to measure image quality or classification accuracy. This is essentially true when it comes to design of computer-aided diagnostic systems where their effectiveness is measured by their end users in which case the cost function is generally human errors. Furthermore, ROC analysis is developed for detection in the context of binary hypothesis testing problems. In chemical/biological warfare (CBW) defense, estimation of chemical/biological (CB) agent abundance is more critical than CB agent detection since the lethal level of concentration of different CB agents poses different threats. The detection-based ROC curves cannot address this need. Chapter 3 is included to resolve this issue where a 3D ROC analysis is developed by creating a third dimension to specify target abundance so that a 3D-ROC curve can be generated and plotted based on three parameters, detection probability, P_{D}, false alarm probability, P_{F}, and threshold τ. Consequently, the traditional detection-based ROC curves, referred to as 2D ROC curves, become a special case of 3D ROC curves. As noted, most hyperspectral imaging techniques are actually derived from various aspects of estimation, which produce abundance fractions of signatures of interest such as linear spectral mixture analysis. In order to evaluate their performance for quantitative analysis the estimated abundance fractions must be converted to hard decisions via a threshold τ. The 3D ROC analysis provides a feasible tool for this purpose.

Chapter 4 is Design of Synthetic Image Experiments. One of major difficulties in algorithm design is how to evaluate various algorithms objectively and impartially on a fair common ground. In doing so, the first concern is the data to be used for experiments that must be available and assessable for those who are interested in comparing their designed algorithms to others. This can be done by using data sets in the public domain. A second concern is that the experiments should be repeatable for performance assessment. A third and most important one is design of experiments that should have controllable parameters to generate desired ground truth to address issues to be investigated. Chapter 4 takes advantage of real image scenes available on web site to simulate synthetic images with various scenarios that can be designed for this purpose.

Chapter 5 is Virtual Dimensionality of Hyperspectral Data that revisits a recently developed concept called virtual dimensionality defined in Chapter 17 of Chang (2003a) as the number of spectrally distinct signatures in hyperspectral imagery. VD has been found to be very useful in many applications (Chang, 2006a, 2006b) such as DR in Wang and Chang (2006a), BS in Chang and Wang (2006), and endmember extraction in Wang and Chang (2006b). Accordingly, a new way of reinterpreting VD becomes imperative. Chapter 5 is a result of such an effort where VD is explored for new interpretation and various techniques are also developed to estimate VD for different applications.

Chapter 6 is Data Dimensionality Reduction. It provides a comprehensive study and survey on many popular and commonly used dimensionality reduction (DR) techniques, which can be treated in two separate categories: dimensionality reduction by transform (DRT) and DR by band selection (DRBS). Specifically, DRT comprises two types of transforms: component analysis (CA)-based transforms, which are derived from statistics of various orders including 2nd order statistics-based principal components analysis (PCA), 3rd order statistics-based skewness, 4th order statistics-based kurtosis and statistical independency-based independent component analysis (ICA) and feature extraction (FE)-based transforms, Fisher's ratio-based linear discriminant analysis (FLDA), and linear mixture model-based OSP. As an alternative to DRT, DRBS selects an appropriate subset of bands from the original band set to replace the high-dimensional original data set with a low-dimensional data set represented by selected bands. So, technically speaking, DRBS performs data reduction, not data compression, by reducing band dimensionality without processing data in the sense that selected bands form a new data cube with all the unselected bands being discarded. While both DRT and DRBS accomplish the same goal, they present different rationales in DR. The former is developed to compact data information in low dimensions via a transform, while the latter represents the original high-dimensional data by its low-dimensional data via band selection. As a consequence, the effectiveness of DR and BS is measured by the transform used for DR and criteria used for BS. Nevertheless, DRT and DRBS do share the same fundamental issue, that is, "how many dimensions are required to be retained after DRT?" and "how many bands are needed for DRBS to faithfully represent the original data?." Interestingly, such an issue has been either overlooked or intentionally avoided in the past because finding an effective criterion for determination of the number of dimensions to be retained or bands to be selected is extremely challenging. Figure 1.1 lists six chapters in Part I to provide background knowledge for follow-up chapters.

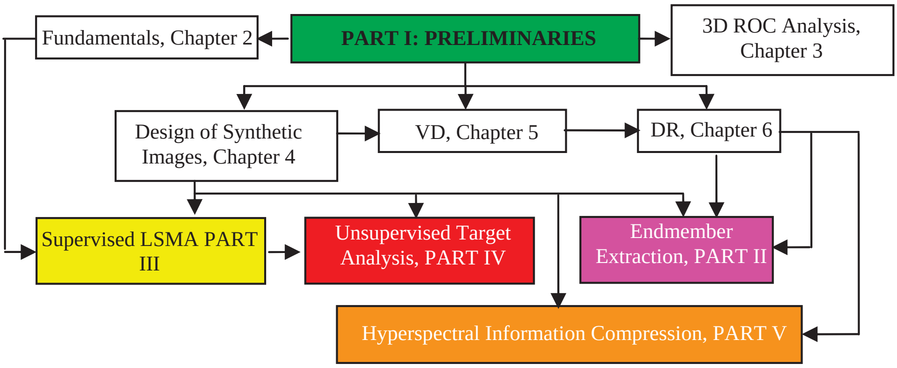

Fundamentals, Chapter 2
PART I: PRELIMINARIES
3D ROC Analysis, Chapter 3
Design of Synthetic Images, Chapter 4
VD, Chapter 5
DR, Chapter 6
Supervised LSMA PART III
Unsupervised Target Analysis, PART IV
Endmember Extraction, PART II
Hyperspectral Information Compression, PART V

Figure 1.1 Six chapters in Part I to provide background knowledge.

1.5.2 Part II: Endmember Extraction

Endmembers are probably one of most important features in hyperspectral data exploitation since they represent pure signatures used to specify distinct spectral classes. So, finding endmembers becomes a very crucial preprocessing step for hyperspectral image analysis. This is particularly true for linear spectral mixture analysis (LSMA) that requires a set of basic material constituents, referred to as image endmembers to form a linear mixing model to unmix data in terms of abundance fractions of these endmembers. However, the prior knowledge of such image endmembers is usually not available *a priori*. Therefore, endmember extraction comes to play a key role in finding such image endmembers. Unfortunately, the research in endmember extraction has not received much attention in early days until recently. This may be partly due to the fact that many research efforts in remote sensing image processing have been directed to design and development of supervised methods where the necessary prior knowledge is assumed provided *a priori*. In this case, there is no need of finding endmembers. Second, because of low spectral or spatial resolution most image pixels appear in a mixed form rather than as pure pixels. Consequently, the presence of endmembers is considered to be very rare. From a land use/land cover's point of view there may be few endmembers that have little impact on image classification. However, from a viewpoint of intelligence endmembers provide crucial and critical information since their existence is unexpected. Specifically, when they appear, only a small population will be present and cannot be identified by prior knowledge. Additionally, the low probability of their occurrence also makes their detection very difficult. Part II is devoted to this topic. Most importantly, it develops various algorithms of different forms for endmember extraction.

Basically, an endmember extraction algorithm (EEA) can be categorized into simultaneous EEA (SM-EEA) and sequential EAA (SQ-EEA) depending upon how it generates endmembers. An SM-EEA generates a required number of endmembers all together compared to an SQ-EEA, which generates one endmember at a time until it reaches a required number of endmembers. On the other hand, based on how initial conditions are used for initialization, an EEA can be also categorized into initialization-driven EEA (ID-EEA) and random EEA (REEA). These two types of EEAs adopt completely opposite philosophies. An ID-EEA selects a specific set of initial endmembers to avoid randomness caused by the use of random initial endmembers compared to an REEA, which converts the disadvantage resulting from random nature of initial endmembers to an advantage of making an EEA immune to random initial conditions. In order to treat EEAs systematically and logically, Chapter 7 first considers SM-EEAs followed by SQ-EEAs in Chapter 8, ID-EEAs in Chapter 9, and REEA in Chapter 10. Finally, Part II is concluded by Chapter 11, which explores relationships among various EEAs studied in Chapters 7–10. Figure 1.2 outlines the organization of five chapters in Part II.

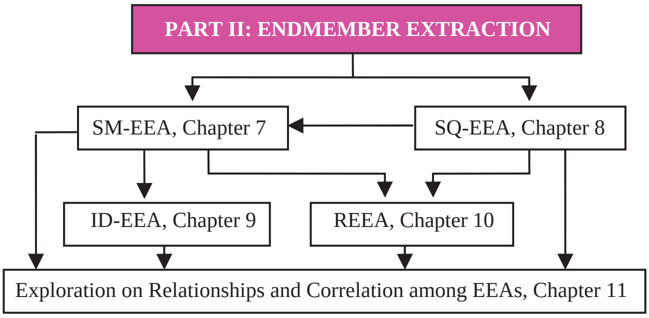

PART II: ENDMEMBER EXTRACTION
SM-EEA, Chapter 7
SQ-EEA, Chapter 8
ID-EEA, Chapter 9
REEA, Chapter 10
Exploration on Relationships and Correlation among EEAs, Chapter 11

Figure 1.2 Organization of five chapters in Part II.

1.5.3 Part III: Supervised Linear Hyperspectral Mixture Analysis

Supervised linear hyperspectral mixture analysis (SLSMA) is probably the most widely used hyperspectral imaging technique to perform various tasks for data analysis. It makes an assumption that a data sample vector can be described by a linear mixing model as a linear mixture of a finite number of known basic signature constituents called image endmembers, from which it can be unmixed via a specific linear spectral unmixing technique into abundance fractions of these image endmembers. Since SLSMA has been previously treated in the book by Chang (2003a), the five chapters, Chapters 12–15 presented in this book, can be considered as an expansion of SLSMA and complement to the LSMA discussed in Chang (2003a). Chapter 12 revisits the orthogonal subspace projection originally developed by Harsanyi and Chang (1994). In particular, when only partial knowledge such as desired target information is provided with no prior background knowledge, OSP can be implemented as the constrained energy minimization developed in Harsanyi's dissertation (1993). If no prior knowledge is available, then OSP can be implemented as RX detector (Reed and Yu, 1990) for anomaly detection. Chapter 13 presents a third approach to SLSMA, Fisher's linear spectral mixture analysis (FLSMA), which replaces the signal-to-noise ratio criterion used by OSP or least squares error (LSE) used by LSOSP with the criterion of Fisher's ratio. Chapter 14 further extends OSP and FLSMA to WAC-LSMA by replacing the commonly used LSE with weighted LSE. While Chapters 13 and 14 extend SLSMA via imposing constraints on the used linear mixing model, Chapter 15 derives kernel-based LSMA, which extends SLSMA techniques to their kernel-based counterparts via nonlinear functions. Figure 1.3 outlines the organization of four chapters in Part III.

1.5.4 Part IV: Unsupervised Hyperspectral Analysis

One of major tasks in hyperspectral imaging is target detection and classification. Due to its high spectral resolution, targets of interest are generally different from those in multispectral imagery. For example, endmembers and anomalies that generally do not contribute much to land cover/land use classification are actually crucial in hyperspectral image analysis. Other targets of interest in hyperspectral data analysis also include rare minerals in geology, special spices in agriculture and ecology, drug trafficking in law enforcement, combat vehicles in battlefield, man-made targets in intelligent analysis, and so on. Realistically, most of such targets generally appear as either mixed pixels or subpixels. So, the major goal of Part IV is to extend the SLSMA in Part III to ULSMA where two main issues that do not occur in the SLSMA need to be addressed in ULSMA. One is the number of signature sources of interest, *p*. The other is how to find these signature sources once the value of the *p* is determined. Since the first issue can be addressed by the concept of VD developed in Chapter 5, the main theme of Part IV is primarily focused on the second issue.

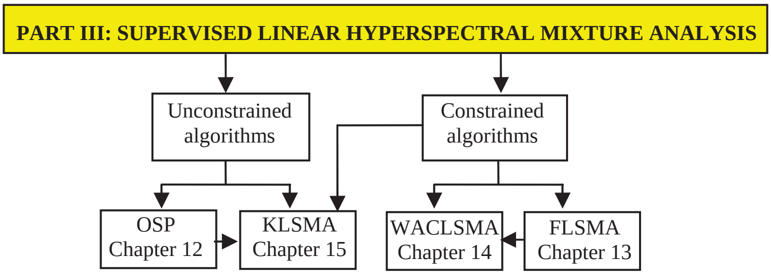


Figure 1.3 Organization of four chapters in Part III.

Chapter 16 investigates two types of hyperspectral measures: signature-based and correlation-weighted measures, both of which can be used to discriminate and identify unknown signature vectors for unsupervised data analysis. The former includes the spectral angle mapper (SAM), Euclidean distance, spectral information divergence (SID), and orthogonal projection divergence (OPD), while the latter uses the sample spectral correlation as a weighting factor to measure signature similarity for discrimination and identification.

Chapter 17 extends SLSMA to ULSMA. In doing so, two approaches are developed to find unknown image endmembers, referred to as virtual signatures (VSs). The first one is to implement LSMA techniques in an unsupervised manner on the original data and its sphered data to find two sets of VSs corresponding to background and target signatures, respectively. A second approach is to use components analysis methods where PCA and ICA are implemented to find unknown background and target signatures, respectively.

Due to substantial amount of information provided by hundreds of contiguous spectral bands it is interesting to know how much information can be extracted from a single hyperspectral image pixel vector as well as how to process the extracted pixel information for data analysis. In traditional image processing the only image pixel information is uniquely specified by its gray-level value. In multispectral image processing with only tens of discrete spectral bands in use, the spectral information provided by a multispectral image pixel is generally very limited compared to that provided by a hyperspectral image pixel. So, the issue in exploration of information extraction from a single hyperspectral image pixel vector has not received much interest as it should. Very little work has been done in the past. For example, an endmember itself provides vital information of a particular spectral class. Another example is an anomaly that provides information in identifying unknown targets. While an endmember is specifically defined, the definition of anomaly seems vague with a general understanding that an anomaly is a target whose spectral signature is distinct from those of pixels in its surrounding neighborhood. However, how large should a surrounding neighborhood be for a pixel vector to be qualified as anomalous pixel vector? So far, there is no answer to it. More generally, for a given pixel vector, how can we characterize the pixel vector as a subpixel vector or a mixed pixel vector or an anomalous pixel vector or a pixel vector of some other type? Besides, can an endmember be a pure pixel vector, in which case it is referred to as endmember pixel vector or vice versa? Can a pixel be both an anomalous pixel vector and an endmember pixel vector? As a complete opposite to anomaly, how can we view a pixel vector if the spectral signatures of pixel vectors in its proximity are very similar and close to each other? Interestingly, these issues have never been investigated on a single pixel vector basis. So, Chapter 18 investigates the issue of “what spectral information can be extracted from a single hyperspectral image pixel vector?” Figure 1.4 outlines the organization of two chapters in Part IV.

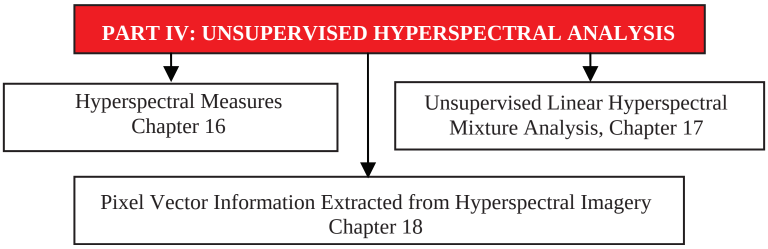

PART IV: UNSUPERVISED HYPERSPECTRAL ANALYSIS
Hyperspectral Measures Chapter 16
Unsupervised Linear Hyperspectral Mixture Analysis, Chapter 17
Pixel Vector Information Extracted from Hyperspectral Imagery Chapter 18

Figure 1.4 Organization of three chapters in Part IV.

1.5.5 Part V: Hyperspectral Information Compression

Data compression has received increasing interest in hyperspectral data analysis because of the vast amount of data volumes needed to be processed and significant redundancy resulting from high interband spectral correlation. Since a hyperspectral image can be viewed as a 3D image cube, a common practice is a direct application of 3D compression techniques available in image/video processing to hyperspectral imagery so as to achieve so-called hyperspectral data compression. Unfortunately, there are several issues arising from such an approach. One is how to deal with spectral compression from very high spectral resolution provided by a hyperspectral imaging sensor. The reason why the hyperspectral imagery is called "*hyperspectral*" is due to its wealthy spectral information, which offers unique spectral characterization that cannot be provided by spatial information, particularly, the spectral profile information provided by subpixels and mixed pixels across its acquired wavelength range by hundreds of spectral channels. Therefore, from a hyperspectral imagery point of view, spectral information is usually more important and crucial than spatial information when it comes to hyperspectral image analysts. When hyperspectral compression is performed, extra care must be taken of in order to preserve spectral characteristics and properties. For example, when targets of interest are rare such as anomalies and endmembers, their spatial extent is generally very small and limited. Thus, the spatial correlation resulting from such targets will be too little to be used for spatial compression. In this case a direct spatial compression without taking into account spectral properties of these targets may result in significant loss of information that characterizes these targets. As a consequence, blindly applying 3D compression techniques to hyperspectral data may not be able to achieve effective compression from an exploitation perspective. Accordingly, a more appropriate approach is to consider "*information*" compression rather than "*data*" compression since the compression is performed based on preservation of the information of interest instead of reduction in data size. More specifically, an effective technique in compressing data size does not necessarily imply that it is also effective in compressing information to be retained. To resolve this dilemma, an effective means of compressing hyperspectral imagery may be one that performs compression in a two-stage process that carries out spectral compression in the first stage to preserve crucial spectral information to avoid being compromised by the follow-up spatial compression in the second stage (Ramakrishna *et al.*, 2005a, 2005b). Such a two-stage compression is referred to as *hyperspectral information compression* or *exploitation-based lossy hyperspectral data compression* in this book as opposed to *lossy hyperspectral data compression*, commonly referred in the literature. Five chapters are presented in Part V and outlined in Figure 1.5.

Chapter 19 reviews issues arising in data compression commonly used in the literature and further introduces a new concept of hyperspectral information compression or exploitation-based lossy hyperspectral data compression where various approaches can be derived for different applications in data exploitation. This chapter is followed by two new approaches to hyperspectral information compression developed in Chapters 20 and 21, which develop techniques to process spectral dimensions and band dimensions in a progressive manner, referred to as progressive spectral dimensionality process (PSDP) and progressive band dimensionality process (PBDP), respectively. In order to more effectively determine spectral and band dimensionality to be used for material classification Chapter 22 presents a new idea of dynamic dimensionality allocation (DDA). By taking advantage of PBDP in Chapter 21 and DDA in Chapter 22 a new approach to band selection, called progressive band selection (PBS), is further developed and presented in Chapter 23.

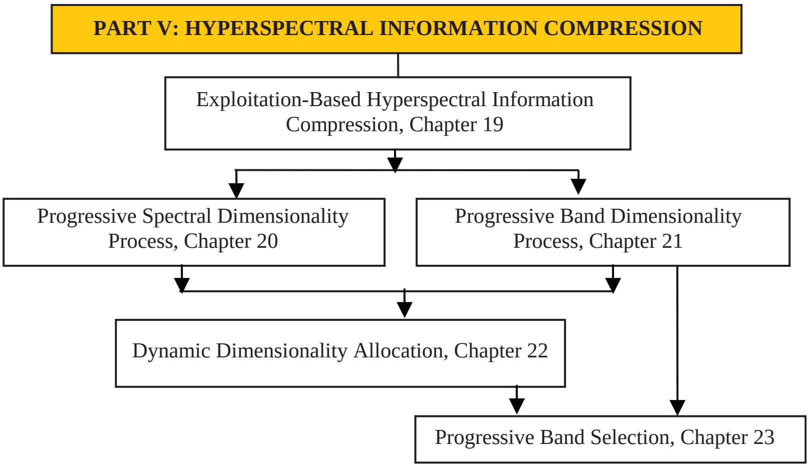

PART V: HYPERSPECTRAL INFORMATION COMPRESSION
Exploitation-Based Hyperspectral Information Compression, Chapter 19
Progressive Spectral Dimensionality Process, Chapter 20
Progressive Band Dimensionality Process, Chapter 21
Dynamic Dimensionality Allocation, Chapter 22
Progressive Band Selection, Chapter 23

Figure 1.5 Organization of three chapters in Part V.

1.5.6 Part VI: Hyperspectral Signal Coding

So far, data processing discussed in all the previous chapters, Chapters 7–23, is considered as hyperspectral image processing because the considered data are image data cubes and the techniques are developed to process hyperspectral data as an image cube with data samples treated as image pixel vectors. However, due to the use of hundreds of spectral channels a hyperspectral data sample vector already contains spectral information that can be used for data analysis without relying on sample spectral correlation provided by image structures. So, instead of considering a data sample vector as an image pixel vector in an image cube, a data sample vector can also be processed as a one-dimensional signal, referred to as a hyperspectral signal or signature vector rather than as a hyperspectral image pixel vector. In this case a hyperspectral signal is a spectral signature of a material substance specified by hundreds of spectral channels across a certain range of wavelengths. In this book, both hyperspectral signal and signature vector will be used interchangeably as appropriate. The data processing of hyperspectral signals or signature vectors is called hyperspectral signal processing to distinguish it from hyperspectral image processing discussed in previous chapters. The only difference between hyperspectral image processing and hyperspectral signal processing is that the former takes advantage of statistics resulting from spectral correlation among pixel vectors in an image cube, while the latter processes a hyperspectral signal as an individual 1D signal such as signatures from spectral libraries or databases without accounting for spectral correlation among sample signals. As a result, when a hyperspectral signal is processed, the information available for processing is only the spectral information within the signal without referencing spectral correlation with other signals. Accordingly, 1D hyperspectral signal processing is primarily used as signal discrimination, detection, classification, representation, and identification. Having this clear distinction in mind, Part VI and Part VII are devoted to hyperspectral signal processing with an understanding that no sample spectral correlation is available to be used for data processing.

The main focus of Part VI is on signal coding that encodes a hyperspectral signal as a code word for its discrete representation. How fine and accurate such discrete representation of a hyperspectral signal can be is determined by the total number of bits used for encoding. Three types of encoding methods are developed in this part. One is binary coding in Chapter 24, which performs

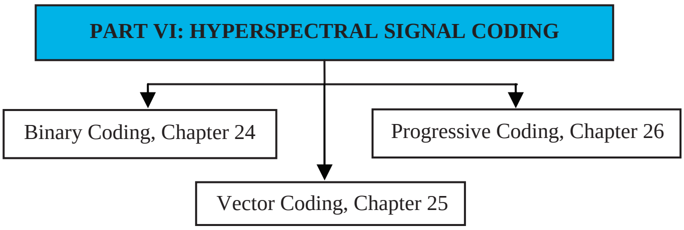

PART VI: HYPERSPECTRAL SIGNAL CODING
Binary Coding, Chapter 24
Progressive Coding, Chapter 26
Vector Coding, Chapter 25

Figure 1.6 Organization of three chapters in Part VI.

memoryless coding. Another is vector coding in Chapter 25, which takes advantage of memory to perform signature coding. A third one discussed in Chapter 26 is progressive coding, which encodes a hyperspectral signal stage by stage in a progressive manner. Figure 1.6 outlines the organization of three chapters in Part VI.

1.5.7 Part VII: Hyperspectral Signal Feature Characterization

While the hyperspectral signal coding considered in Part VI converts a hyperspectral signal to a codeword as its discrete representation so that different hyperspectral signatures can be discriminated and identified via their encoded code words, Part VII can be considered as a counterpart of Part VI to perform hyperspectral signal characterization by converting a hyperspectral signal as a continuous representation. Three major techniques are developed: OSP-based variable-number variable-band selection (VNVBS) in Chapter 27 for hyperspectral signals, Kalman filter-based techniques in Chapter 28 for hyperspectral signal estimation, and wavelet-based techniques in Chapter 29 for hyperspectral signal representation. Figure 1.7 outlines the organization of these three chapters in Part VII.

1.5.8 Applications

This book concludes with applications of hyperspectral data processing in various areas.

1.5.8.1 Chapter 30: Applications of Target Detection

The subpixel target detection discussed in Chapter 2 has major interests in many applications. Since the size of a subpixel target is smaller than pixel resolution specified by ground sampling distance, it is embedded in a single pixel vector and cannot be visualized by inspection. Therefore,

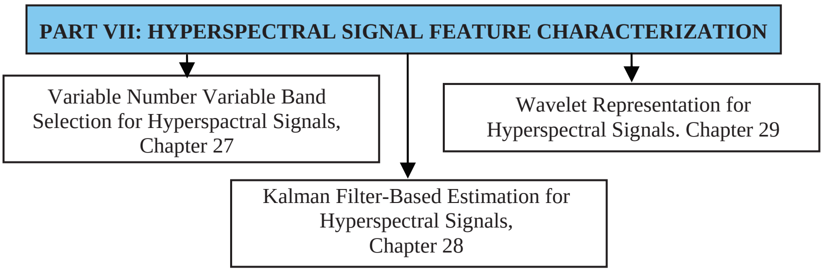

PART VII: HYPERSPECTRAL SIGNAL FEATURE CHARACTERIZATION
Variable Number Variable Band Selection for Hyperspactral Signals, Chapter 27
Wavelet Representation for Hyperspectral Signals. Chapter 29
Kalman Filter-Based Estimation for Hyperspectral Signals, Chapter 28

Figure 1.7 Organization of three chapters in Part VII.

it looks like that the best we can do for a subpixel target is detection and finding the size of a subpixel target seems out of reach. Chapter 30 provides a means of doing so. Specifically, the size of a subpixel target can be calculated by multiplying the pixel resolution with the estimated abundance fraction of the subpixel target embedded in a pixel vector. Consequently, finding the true size of a subpixel target is equivalent to accurately estimating the abundance fraction of a subpixel target.

Many problems addressed by target detection assume that the targets to be detected are exposed, in which case it makes detection easy and more effective. However, in remote sensing targets of interest may be hidden under natural environments due to terrain characteristics such as shadow and shade. On the other hand, in many military and intelligence applications, the targets of interest may be concealed weapons or combat vehicles, which are camouflaged or canvassed. Detecting such concealed targets generally presents a great challenge in an unknown image scene due to the fact that the prior knowledge about targets of interest and background is not available. The second part of Chapter 30 develops an approach to detection of unknown concealed targets. It comprises three successive stage processes: (1) band selection procedure in the first stage; (2) band ratio approach in the second stage; and (3) automatic target detection in the third stage. The objective of the band selection is to select an appropriate set of band images for the band ratio transformation and the selected bands are subsequently ratioed to form a desired set of images used for subsequent automatic target detection carried out in the third stage.

1.5.8.2 Chapter 31: Nonlinear Dimensionality Expansion to Multispectral Imagery

The data processing techniques developed in this book are mainly derived from a perspective of how to process hyperspectral imagery. Their applications to multispectral imagery may not be immediately obvious and trivial. Specifically, the pigeon-hole principle described in Section 1.3 that holds for hyperspectral imagery is no longer true for multispectral imagery and virtual dimensionality. In order for a hyperspectral imaging technique to be applied to multispectral imagery, it hinges on two key issues, how to define a hyperspectral image and a multispectral image as well as how to distinguish one from another. Interestingly, the pigeon-hole principle once again proves to be a valuable means of doing so. When there are few pigeon holes than pigeons, it implies that few spectral bands than signal sources can be used for signal discrimination in which case the image is defined as a multispectral image. Otherwise, it is a hyperspectral image. Such definitions seem controversial in the first place. As a matter of fact, similar definitions can be found in ICA. That is, if the number of data sample vectors is fewer than the number of signal sources to be separated, an ICA is defined as an over-complete ICA. Otherwise, an ICA is defined as an under-complete ICA. The definitions of over-complete ICA and under-complete ICA shed light on how to distinguish multispectral image from hyperspectral images. In ICA a data sample vector represents a linear mixture of random signal sources to be separated. This is similar to viewing a data sample vector as a linear mixture of signal sources to be present in the data. So, LSMA used to unmix a multispectral image tries to solve an over-complete linear spectral unmixing problem, while LSMA used to unmix a hyperspectral image intends to solve an under-complete linear spectral unmixing problem. By virtue of this interpretation, this chapter develops two approaches to conversion of a hyperspectral imaging technique to a multispectral imaging technique by nonlinear dimensionality expansion (NDE). One is band dimensionality expansion, which implements a band expansion process (BEP) to create new additional images from the original set of spectral images via nonlinear functions. The other is kernel-based method that kerenlizes LSMA-based techniques via nonlinear kernels to solve linear nonseparability issue arising in multispectral image analysis.

1.5.8.3 Chapter 32: Multispectral Magnetic Resonance Imaging

Recently, a new application of hyperspectral imaging techniques in multispectral imagery, magnetic resonance (MR) image analysis, has been investigated where MR images can be considered as multispectral images and each image acquired by a particular MR pulse sequence can be considered as a spectral band image. As a result, MR images are actually an image cube collected by particularly designed MR image pulse sequences. With this interpretation Chapter 32 extends results in Chapter 31 to MR image analysis.

1.6 Laboratory Data to be Used in This Book

Three sets of laboratory data will be used for experiments in this book, two of which were collected by the airborne visible infrared imaging spectrometer (AVIRIS) and the third one is a gas data set.

1.6.1 Laboratory Data

One data set to be used in this book is the one used in Harsanyi and Chang (1994). It is AVIRIS reflectance data shown in Figure 1.8, which has five field reflectance spectra, blackbrush, creosote leaves, dry grass, red soil, and sagebrush with spectral coverage from 0.4 to 2.5 μm and 158 bands after the water bands are removed.

1.6.2 Cuprite Data

Another useful laboratory data that is available on the web site http://speclab.cr.usgs.gov/ is the reflectance spectra of five USGS ground-truth mineral spectra: alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M) shown in Figure 1.9. Each of the five mineral spectral signatures is collected by 224 spectral bands at spectral resolution of 10 nm in the range of 0.4–2.5 μm.

1.6.3 NIST/EPA Gas-Phase Infrared Database

A third data set is one provided by the National Institute of Standards and Technology (NIST) and also available on the web sites http://www.nist.gov/srd/nist35.htm and webbook.nist.gov/chemistry. This data set was also used for the study in Kwan *et al.* (2006). It contains the nine

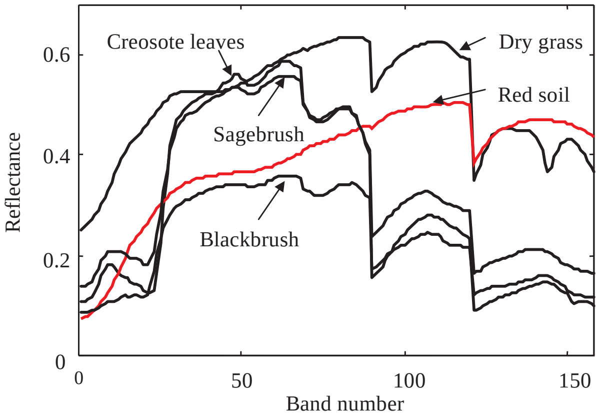


Figure 1.8 Spectra of five AVIRIS reflectances.

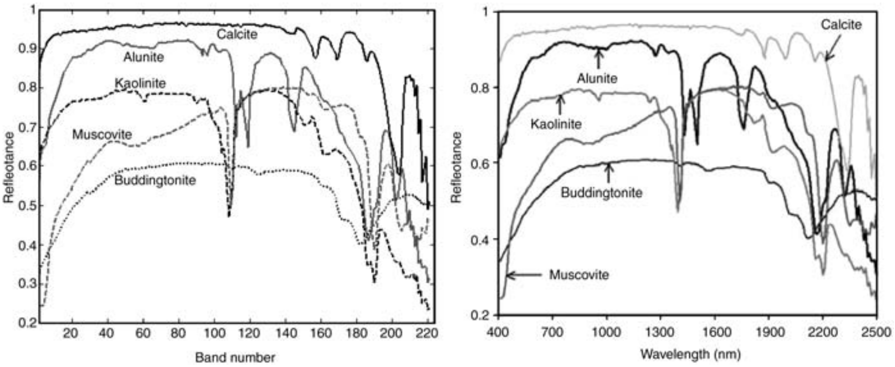


Figure 1.9 Five USGS ground-truth mineral spectra.

Table 1.1 Nine gas agent data signatures

Signature no.	Signature name
$\mathbf{s}_1$	2-Chloroethymethyl sulfide
$\mathbf{s}_2$	Diethyl ethylphosphonate
$\mathbf{s}_3$	Ethanol
$\mathbf{s}_4$	Freon 114
$\mathbf{s}_5$	*n*-Butyl bromide
$\mathbf{s}_6$	Bis-2-ethyl-1-hexyl phosphonate
$\mathbf{s}_7$	Benzyl benzoate
$\mathbf{s}_8$	Dibenzyl ether
$\mathbf{s}_9$	Piperidine

gas agents labeled by $\{\mathbf{s}_i\}_{i=1}^{9}$ listed in Table 1.1 with their spectral signatures shown in Figure 1.10. This data set is included particularly for signal processing algorithm design and development for hyperspectral signal processing to investigate hyperspectral signature analysis and characterization in Part VI and Part VII.

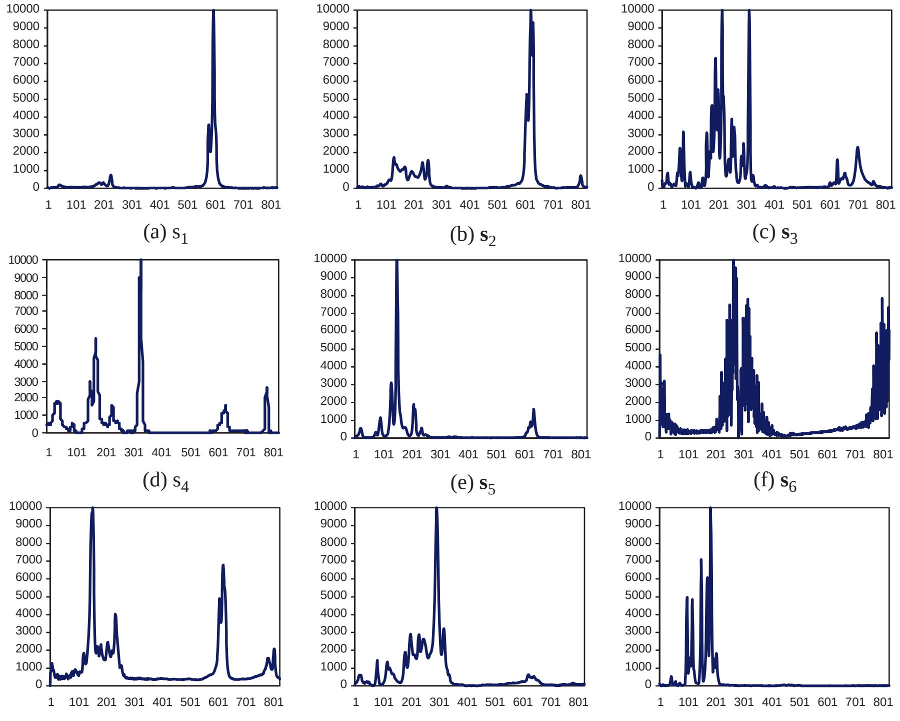
Except that the frequency range of $\mathbf{s}_1$ is 550–3846 cm^{-1} acquired by 825 bands, all the $\{\mathbf{s}_i\}_{i=2}^{9}$ has frequency range of 450–3966 cm^{-1} acquired by 880 bands, giving each signature a spectral resolution of about 4 cm^{-1} per band.

1.7 Real Hyperspectral Images to be Used in this Book

Three real hyperspectral image data sets are frequently used in this book for experiments. Two are AVIRIS real image data sets, Cuprite in Nevada and Purdue's Indian Pine test site in Indiana. A third image data set is HYperspectral Digital Imagery Collection Experiment (HYDICE) image scene. Each of these three data sets is briefly described as follows.

1.7.1 AVIRIS Data

Two AVIRIS data sets presented in this section are Cuprite data and Purdue's data, which can be used for different purposes in applications. The Cuprite data set is generally used for endmember



(a) s_1 (b) $\mathbf{s}_2$ (c) $\mathbf{s}_3$

(d) s_4 (e) $\mathbf{s}_5$ (f) $\mathbf{s}_6$

(g) s_7 (h) $\mathbf{s}_8$ (i) $\mathbf{s}_9$

Figure 1.10 Spectral signatures of nine chemical/infrared data signatures.

extraction and target detection, while the Purdue's data set is mainly used for land cover/land use classification.

1.7.1.1 Cuprite Data

One of the most widely used hyperspectral image scenes available in the public domain is Cuprite mining site, Nevada, as shown in Figure 1.11(a). It is an image scene of 20 m spatial resolution collected by 224 bands using 10 nm spectral resolution in the range of 0.4–2.5 μm. The center region shown in Figure 1.11(b), cropped from the image scene in Figure 1.10(a), has size of 350 × 350 pixel vectors.

Since it is well understood mineralogically and has reliable ground truth, this scene has been studied extensively. Two data sets for this scene, reflectance and radiance data, are also available for study. There are five pure pixels in Figure 1.11(a, b) that can be identified to be corresponding to five different minerals, alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M) labeled by A, B, C, K, and M, respectively, in Figure 1.12(b) with their corresponding reflectance and radiance spectra shown in Figure 1.12(c, d).

These five pure pixels are carefully verified using laboratory spectra provided by the USGS (available from http://speclab.cr.usgs.gov) and selected by comparing their reflectance spectra in Figure 1.12(c) against the lab reflectance data in Figure 1.9. Figure 1.12(e) also shows an alteration map for some of the minerals, which is generalized from ground map provided by the USGS

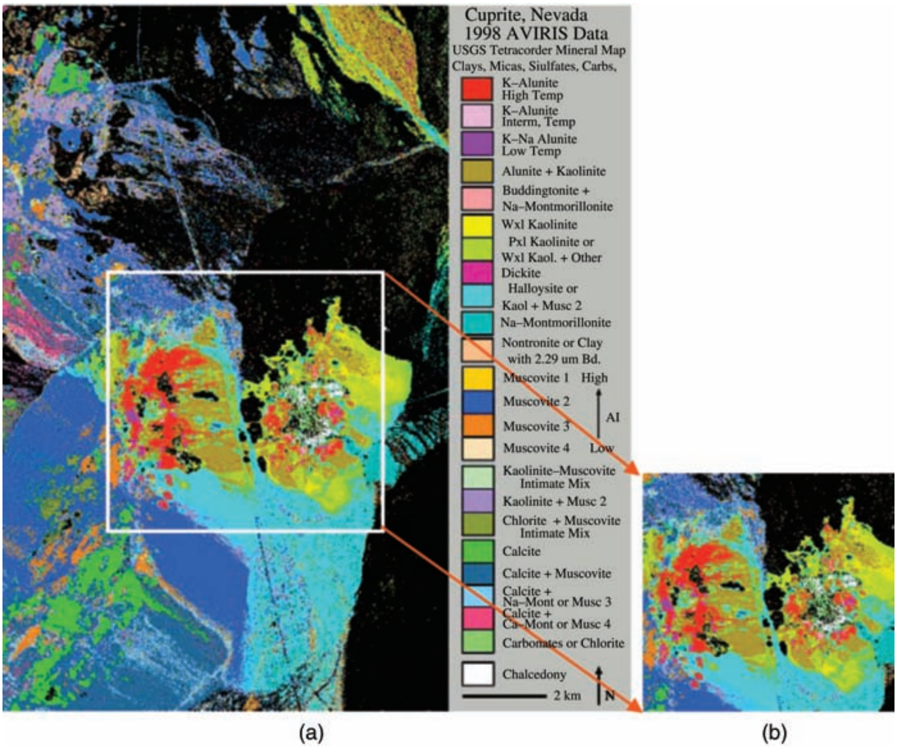
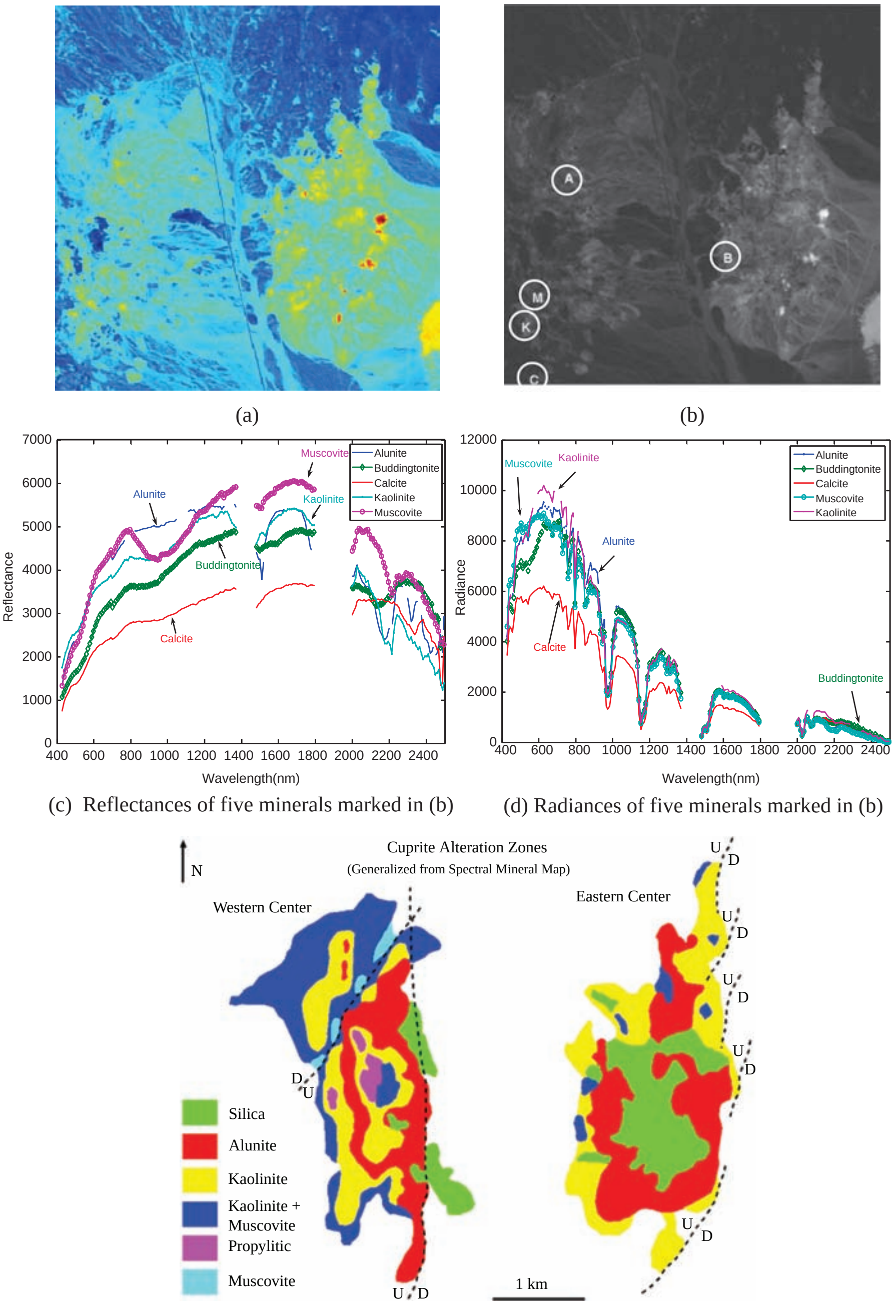

Cuprite, Nevada
1998 AVIRIS Data
USGS Tetracorder Mineral Map
Clays, Micas, Siulfates, Carbs,
K–Alunite High Temp
K–Alunite Interm, Temp
K–Na Alunite Low Temp
Alunite + Kaolinite
Buddingtonite + Na–Montmorillonite
Wxl Kaolinite
Pxl Kaolinite or Wxl Kaol. + Other
Dickite
Halloysite or Kaol + Musc 2
Na–Montmorillonite
Nontronite or Clay with 2.29 um Bd.
Muscovite 1
Muscovite 2
Muscovite 3
Muscovite 4
High
Al
Low
Kaolinite–Muscovite Intimate Mix
Kaolinite + Musc 2
Chlorite + Muscovite Intimate Mix
Calcite
Calcite + Muscovite
Calcite + Na–Mont or Musc 3
Calcite + Ca–Mont or Musc 4
Carbonates or Chlorite
Chalcedony
2 km
N
(a)
(b)

Figure 1.11 Cuprite image scene, (a) original Cuprite image scene; (b) the image cropped from the center region of the original scene in (a) (350 × 350).

and obtained by Tricorder SW version 3.3. It should be noted that this radiometrically calibrated and atmospherically corrected data set available from http://aviris.jpl.nasa.gov is provided in reflectance units with 224 spectral channels where the data has been calibrated and atmospherically rectified using the ACORN software package. It is recommended that bands 1–3, 105–115, and 150–170 be removed prior to data processing due to their low water absorption and low SNR. As a result, a total of 189 bands are used for experiments as shown in Figure 1.11(c, d). The steps to produce spectra in Figure 1.12(c, d) can be described as follows:

1. Download from http://speclab.cr.usgs.gov/ the laboratory reflectance data.
2. Use spectral angle mapper (SAM) as a spectral similarity measure to identify the five pixels in Figure 1.12(a) that correspond to the five reflectances obtained in step 1 by the following procedure:
   - Remove noisy bands from the five reflectance data.
   - Remove bands with abnormal readings from the spectral library.
   - In order to measure spectral similarity, there is still a need of removing several bands to account for compatibility.

It should be noted that the ground truth is not stored in a “file.” The locations of the five minerals are identified by comparing their reflectance spectra against their corresponding lab reflectances in the spectral library.



(a)

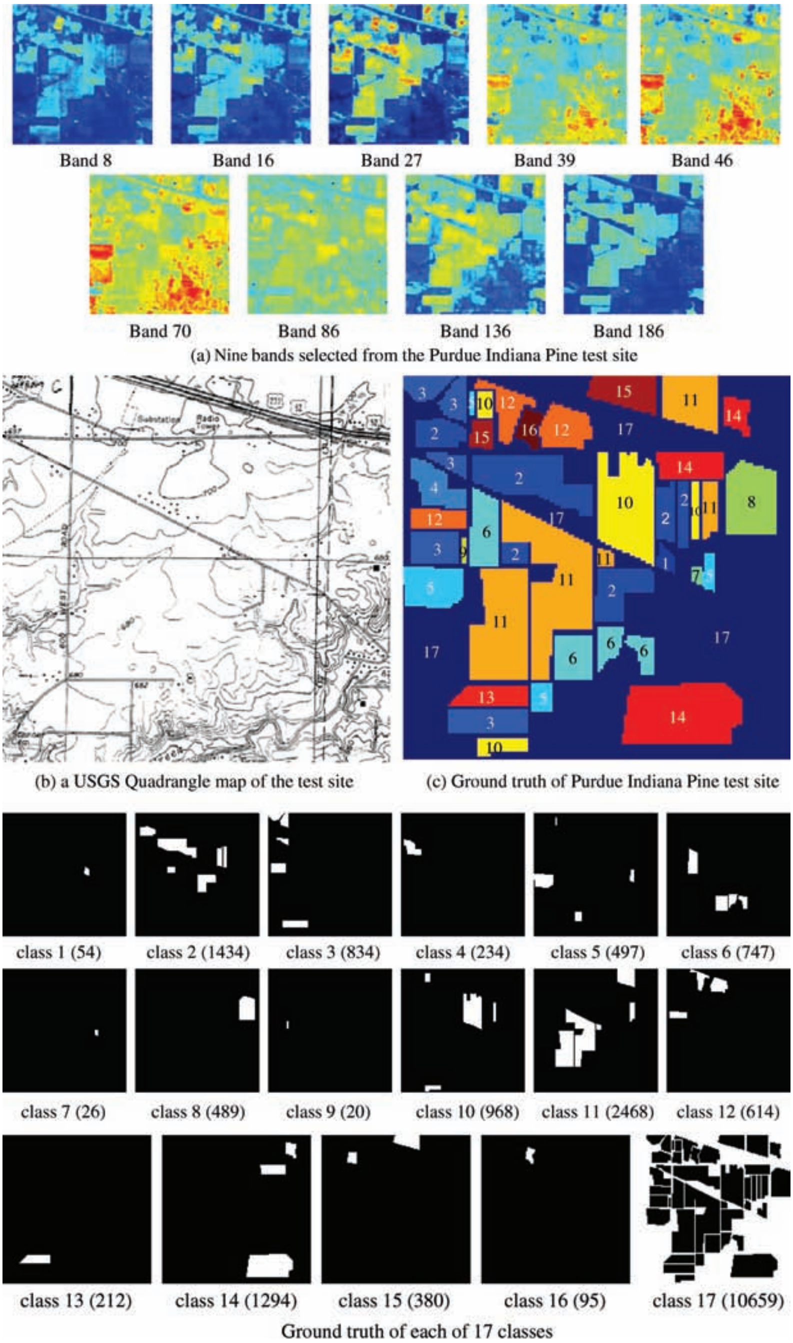
(b)

(c) Reflectances of five minerals marked in (b)

(d) Radiances of five minerals marked in (b)

(c) Alteration mineral map

Figure 1.12 (a) Spectral band number 170 of the Cuprite AVIRIS image scene; (b) spatial positions of five pure pixels corresponding to minerals: alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M); (c) reflectances of five minerals marked in (b) in wavelengths; (d) radiances of five minerals marked in (b) in bands; and (e) alteration mineral map available from USGS.



(a) Nine bands selected from the Purdue Indiana Pine test site

(b) a USGS Quadrangle map of the test site

(c) Ground truth of Purdue Indiana Pine test site

Ground truth of each of 17 classes

Figure 1.13 AVIRIS image scene: Purdue Indiana Pine test site. (a) Nine bands selected from the Purdue Indiana Pine test site; (b) a USGS quadrangle map of the test site; (c) ground truth of Purdue Indiana Pine test site; and (d) ground truth of each of 17 classes.

1.7.1.2 Purdue's Indiana Indian Pine Test Site

Another most widely used real AVIRIS image data set is Purdue's Indiana Indian Pine test site, which has 20 m spatial resolution and 10 nm spectral resolution in the range of 0.4–2.5 μm with size of 145 × 145 pixel vectors taken from an area of mixed agriculture and forestry in Northwestern Indiana, USA. The data set is available on the web site http://cobweb.ecn.purdue.edu/∼biehl/MultiSpec/documentation.html (both download link and ground truth are provided) and was recorded in June 1992 with 220 bands with water absorption bands, bands 104–108 and 150–162 removed and leaving only 202 bands. Figure 1.13(a) shows nine bands selected from the web site and a USGS quadrangle map of the test site provided in Figure 1.13(b).

According to the ground truth provided in Figure 1.13(c) there are 17 classes in this image scene shown in Figure 1.13(d) including the background labeled by class 17, which has a wide variety of targets such as highways, railroad, houses/buildings, and vegetation that may not be of interest in agricultural applications but may be of great interest in other applications such as anomaly detection. The total number of data samples in the scene is $145 \times 145 = 21,025$. Table 1.2 lists labels of each of 17 classes where the numeral in parenthesis under each of 17 classes in Figure 1.13(d) is the number of data samples in that particular class.

Due to the early season of harvest when the data were collected, some cultivated land has very little canopy cover. For example, the corn area can be divided into three classes based on how much is left on the land, which are corn-no till, -min, and corn (class 2–4). The soybean area also can be divided into soybean-no till, -min, and -clean (class 10–12). The grass is mixed with four other materials, which are classified as grass/pasture, grass/trees, grass/pasture-mowed, and bldg-grass-green-drives (class 5, 6, 7, 15). Actually, it is believed that the grass is also mixed in the background. According to Figure 1.13(c, d) (Landgrebe, 2003), the GIS map in Figure 1.13(b) provides the information of "land use" classes instead of "land cover" classes. It means that not every pixel in the map is supposed to be classified into their belonging classes. Additionally, also based on the USGS quadrangle map in Figure 1.13(b), there are dual lane highways (U.S. 52 and 231) and a railroad crossed near the top. The other is Jackson highway, which is near to the middle of the scene. All of them are in the NW–SE direction. Figure 1.13(b) also indicates some houses or buildings by small rectangular dots (Landgrebe, 1998). With this information it is believed that there are at least four classes included in the background: railroad (iron), highway (concrete), houses/buildings (concrete, painted wood, or other materials), and vegetation (grass). The number of classes for such unlabeled areas is important for the unsupervised classification when the total number of classes in the scene is assumed to be unknown.

There are many reasons to select the Purdue Indiana Indian Pine test site for experiments. First of all, it is a well-known image scene available on web site and has been studied extensively. Another is that the pixels in this image scene are heavily mixed. Many algorithms or methods claiming to work well on classification are very likely to break down for this image scene. To the author' best knowledge, most work on this image scene reported in the literature has selected

Table 1.2 Labels of 17 classes

Class 1	Alfalfa	Class 7	Grass/pasture-mowed	Class 13	Wheat
Class 2	Corn-no till	Class 8	Hay-windrowed	Class 14	Woods
Class 3	Corn-min	Class 9	Oats	Class 15	Bldg-grass-green-drives
Class 4	Corn	Class 10	Soybean-no till	Class 16	Stone-steel towers
Class 5	Grass/pasture	Class 11	Soybean-min	Class 17	Background
Class 6	Grass/trees	Class 12	Soybean-clean		

particular areas for study and also supervised based on the provided ground truth. Very little has been done in classification of the entire scene either supervisedly or unsupervisedly. Most interestingly, according to our detailed analysis on the scene, we have found that it is almost impossible to classify all the 17 classes in the image scene even though the complete knowledge of the ground truth provided in Figure 1.13(c, d) is used for classification. This is because pixels in the same class are mixed so badly that values among their spectral signatures measured by any spectral similarity measure vary in a relatively wide range in which pixels in the same class may be classified into different classes and pixels in different classes may be considered to belong to the same class.

From the ground truth provided in Table 1.2, it can be expected that the signatures of three subclasses of corn are close to each other, so are the four subclasses of grass and three subclasses of soybean. However, the relationships among other pairs are still not known. In order to know how much mixing is involved, the signature for each class is calculated by averaging all samples with the same label according to the ground-truth map in Figure 1.12(c). Then the SAM is used to measure how close one class is similar to the other. It has been shown in Liu (2005) that corn and soybean classes (2–4, 10–12) are similar, which account for 6552 pixels, 63% of 10,366 labeled pixels. Similarity also appears in two sets of classes: class 1, 7, 8 and class 6, 9, 13. Surprisingly, the four classes of grass (5, 6, 7, 15), which account for 1650 pixels, are not similar to each other. Additionally, using SAM to measure spectral similarity among 16 classes, it is found that classes 5, 14, 16 seem to be the three most distinct classes and can be classified very easily. It is reasonable and makes sense because class 5 contains chlorophyll, class 14 is wood, and class 16 comprises man-made objects.

With our tremendous experience of working on this image scene, excluding two classes (class 17 that is considered to be the background and class 9 that is considered to be too small) it is found that the spectral signatures of the pixels in the six classes (class 2, class 3, class 4, class 7, class 9, and class 11) are very close in terms of SAM or SID (spectral information divergence in Chang (2003a)). Similarly, the pixels in the three classes (class 8, class 10, and class 15) also have very similar spectral signatures. Hence distinguishing one from another is very difficult. The pixels in the three classes (class 13, class 5, and class 14) have less similar signatures but still present some difficulty with classification. The most dissimilar classes are class 1, class 6, and class 12 that are considered to be easy to classify. By taking into account all the things considered above, we can expect that the classification of this image scene is a great challenge to any hyperspectral imaging algorithm.

1.7.2 HYDICE Data

The HYDICE image scene shown in Figure 1.14(a) has a size of 200 × 74 pixel vectors along with its ground truth provided in Figure 1.14(b) where the center and boundary pixels of objects are highlighted by red and yellow, respectively. The upper part contains fabric panels with size 3, 2, and 1 m^2 from the first column to the third column. Since the spatial resolution of the data is 1.56 m^2, the panels in the third column are considered as subpixel objects. The lower part contains different vehicles with sizes of 4 m × 8 m (the first four vehicles in the first column) and 6 m × 3 m (the bottom vehicle in the first column) and three objects in the second column (the first two have size of 2 pixels and the bottom one has size of 3 pixels, respectively). In this particular scene, there are three types of targets with different sizes, small-size targets (panels of three different sizes, 3, 2, and 1 m^2), and large-size targets (vehicles of two different sizes, 4 m × 8 m and 6 m × 3 m and three objects of 2-pixel and 3-pixel sizes) that can be used to validate and test anomaly detection performance.

Figure 1.14(c) shows an enlarged HYDICE scene from the same flight for visual assessment. It has a size of 33 × 90 pixel vectors with 10 nm spectral resolution and 1.56 m spatial resolution where five vehicles lined up vertically to park along the tree line in the field where the red (R) pixel

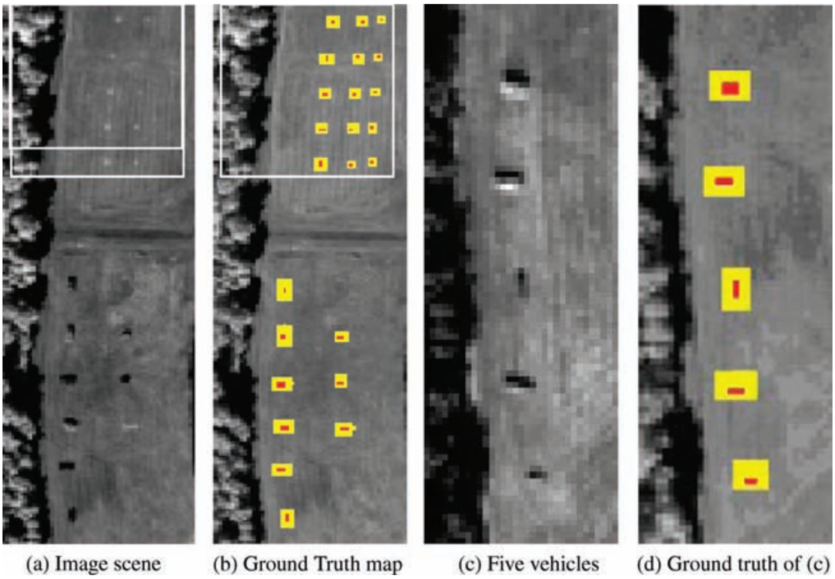


Figure 1.14 HYDICE vehicle scene. (a) Image scene; (b) ground-truth map; (c) five vehicles; and (d) ground truth of (c).

vectors (shown as dark pixels) in Figure 1.14(d) show the center pixel of the vehicles, while the yellow (Y) pixels (shown as bright pixels) are vehicle pixels mixed with background pixels.

A third enlarged HYDICE image scene shown in Figure 1.15(a) is also cropped from the upper part of the image scene in Figure 1.14(a, b) marked by a square.

It has a size of 64 × 64 pixel vectors with 15 panels in the scene. This particular image scene has been well studied in Chang (2003a). Within the scene there is a large grass field background, a forest on the left edge, and a barely visible road running on the right edge of the scene. Low signal/high noise bands: bands 1–3 and bands 202–210; and water vapor absorption bands: bands 101–112 and bands 137–153 were removed. The spatial resolution is 1.56 m, and spectral resolution is 10 nm. There are 15 panels located in the center of the grass field and are arranged in a 5 × 3 matrix as shown in Figure 1.15(b), which provides the ground-truth map of Figure 1.15(a). Each element in this matrix is a square panel and denoted by p_{ij} with row indexed by $i = 1, \ldots, 5$ and column indexed by $j = 1, 2, 3$. For each row $i = 1, \ldots, 5$, the three panels were painted by the same material but have three different sizes. For each column $j = 1, 2, 3$, the five panels have the same size but were painted by five different materials. It should be noted that the panels in rows 2 and 3 are made by the same material with different paints, so did the panels in rows 4 and 5. Nevertheless, they were still considered as different materials. The sizes of the panels in the first, second, and third columns are 3 m × 3 m, 2 m × 2 m, and 1 m × 1 m, respectively. So, the 15 panels have 5 different materials and 3 different sizes. Figure 1.15(b) shows the precise spatial locations of these 15 panels where red pixels (R pixels, i.e., dark pixels) are the panel center pixels and the pixels in yellow (Y pixels, i.e., bright pixels) are panel pixels mixed with background. The 1.56 m spatial resolution of the image scene suggests that the panels in the second and third columns, denoted by p_{12}, p_{13}, p_{22}, p_{23}, p_{32}, p_{33}, p_{42}, p_{43}, p_{52}, p_{53} in Figure 1.15(b) are one pixel in size. Additionally, except the panel in the first row and first column, denoted by p_{11} which also has a size of one pixel, all other panels located in the first column are two-pixel panels, which are the panels in the second row with two pixels lined up vertically, denoted by

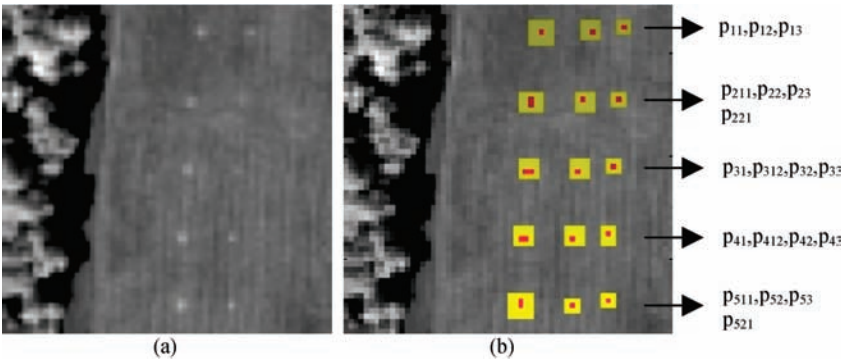

p_{11}, p_{12}, p_{13}
p_{211}, p_{22}, p_{23}
p_{221}
$p_{31}, p_{312}, p_{32}, p_{33}$
$p_{41}, p_{412}, p_{42}, p_{43}$
p_{511}, p_{52}, p_{53}
p_{521}
(a)
(b)

Figure 1.15 (a) A HYDICE panel scene that contains 15 panels; (b) ground-truth map of spatial locations of the 15 panels.

p_{211} and p_{221}; the panel in the third row with two pixels lined up horizontally, denoted by p_{311} and p_{312}; the panel in the fourth row with two pixels also lined up horizontally, denoted by p_{411} and p_{412}; and the panel in the fifth row with two pixels lined up vertically, denoted by p_{511} and p_{521}. Since the size of the panels in the third column is 1 m × 1 m, they cannot be seen visually from Figure 1.15(a) due to its size being smaller than the 1.56 m pixel resolution.

Figure 1.16 plots the five panel spectral signatures obtained from Figure 1.15(b), where the *i*th panel signature, denoted by p_i was generated by averaging the red panel center pixels in row *i*. These panel signatures will be used to represent target knowledge of the panels in each row.

According to visual inspection and ground truth in Figure 1.15(a, b) there are also four background signatures shown in Figure 1.17, which can be identified and marked by interferer, grass, tree, and road. These four signatures along with five panel signatures in Figure 1.16 can be used to form a 9-signature matrix for a linear mixing model to perform supervised linear spectral mixture analysis.

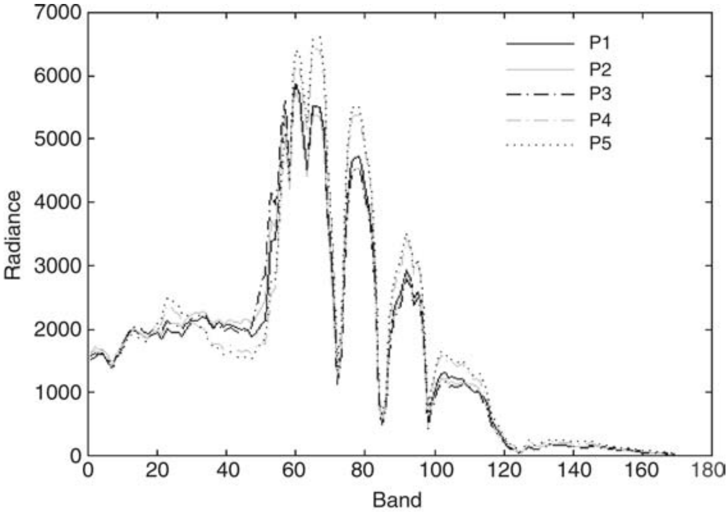


Figure 1.16 Spectra of p_1, p_2, p_3, p_4, and p_5.

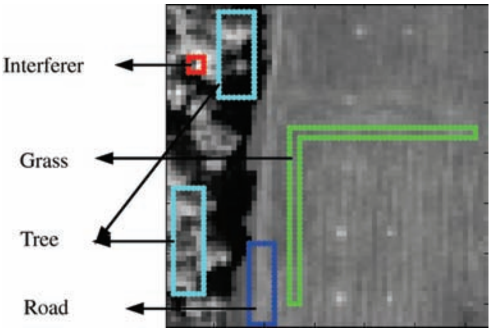


Figure 1.17 Areas identified by ground truth and marked by three background signatures, grass, tree, road plus an interferer.

1.8 Notations and Terminologies to be Used in this Book

Since this book primarily deals with real hyperspectral data, the image pixels are generally mixed and not necessarily pure. The term "endmember" is not used here; instead, a general term "signature" or "signature vector" is used. In addition, because we are only interested in target analysis, the term "targets" instead of "materials" is also used throughout this book. In order to make a distinction between a target pixel and its spectral signature vector, we use notation "**t**" to represent the target pixel vector, "**r**" for an image pixel vector, and "**s**" or "**m**" to indicate its spectral signature vector. We also use bold uppercase for matrices and bold lowercase for vectors. The *italic* upper case "L" will be used for the total number of spectral bands, **K** for the sample spectral covariance matrix, and **R** for the sample spectral correlation matrix. Also, $\delta^*(\mathbf{r})$ is used to represent a detector or classifier that operates on an image pixel vector **r** where the superscript "$*$" in $\delta^*(\mathbf{r})$ specifies what type of a detector or classifier to be used. It should be noted that $\delta^*(\mathbf{r})$ is a real-valued function that takes a form of inner product of a filter vector **w** with **r**, that is, $\delta^*(\mathbf{r}) = (\mathbf{w}^*)^T\mathbf{r}$ with the filter vector $\mathbf{w}^*$ specified by a particular detector or classifier. We also use "$\boldsymbol{\alpha}$"and $\hat{\boldsymbol{\alpha}}$ to represent the abundance vector and its estimate where the notation "hat" over "$\boldsymbol{\alpha}$" indicates "estimate."

$\boldsymbol{\alpha}$: Abundance vector
$\hat{\boldsymbol{\alpha}}$: Estimate of the abundance vector $\boldsymbol{\alpha}$
α_j: jth abundance fraction
$\hat{\alpha}_j$: Estimate of the jth abundance fraction, α_j
A: Weighting or mixing matrix
A_z: Area under an ROC curve
$\mathbf{B}_l$: lth band image
$\mathbf{b}_l$: lth band image represented as a vector
C: Total number of classes
C_j: jth class
d: Desired signature vector
D: Desired signature matrix
$\mathbf{D}_\lambda$: Eigenvalue diagonal matrix
δ: Detector or classifier
Δ: Database
ε: Error threshold
$\mathbf{e}_j$: jth endmember

$\mathbf{I}$: Identity matrix
$I(x)$: Self-information of x
$k(.,.)$: Kernel function
K: Total number of skewers used in PPI
$\mathbf{K}$: Sample covariance matrix
λ: Eigenvalue of sample covariance matrix, $\mathbf{K}$
$\hat{\lambda}$: Eigenvalue of sample correlation matrix, $\mathbf{R}$
l: Index of band number
L: Total number of spectral channels or bands
$\boldsymbol{\Lambda}$: Eigenvector matrix
$\boldsymbol{\mu}$: Global sample mean
$\boldsymbol{\mu}_j$: Global mean of the jth class
$m(.,.)$: Spectral measure
$\mathbf{m}_j$: jth signature vector
$\mathbf{M}$: Signature or endmember matrix
$\mathbf{n}$: Noise vector
N: Total number of image pixel vectors in a band image, i.e., $N = n_r n_c$
n_c: Number of columns in a band image of a hyperspectral image
$n_{\mathbf{D}}$: Number of desired signatures in $\mathbf{D}$
n_r: Number of rows in a band image of a hyperspectral image
$n_{\boldsymbol{\Pi}}$: Number of interferers
n_{T}: Number of training samples
$n_{\mathbf{U}}$: Number of undesired signatures in $\mathbf{U}$
n_{VD}: Value estimated by the VD
p: Number of endmembers
P_{D}: Detection power or probability
P_{F}: False alarm probability
$P_{\mathbf{U}}^{\perp}$: Projector to reject undesired target signatures in $\mathbf{U}$
q: Number of dimensions to be retained after dimensionality reduction
$\hat{q}$: Number of spectral bands required to be selected by band selection
$\mathbf{r}$: Image pixel vector
$\mathbf{R}$: Sample correlation matrix
σ^2: Variance
$\mathbf{S}_{\mathrm{B}}$: Between-class scatter matrix
$\mathbf{S}_{\mathrm{W}}$: Within-class scatter matrix
$\mathbf{t}$: Target signature
τ: Threshold
$\mathbf{w}$: Weight vector
$\mathbf{W}$: Weight matrix
$\mathbf{U}$: Undesired signature matrix
$\mathbf{v}$: Eigenvector
VD_*: The value of the VD obtained by the criterion specified by algorithm "$*$"
ξ: Transform used to perform dimensionality reduction
$\boldsymbol{\Psi}$: Interference matrix
$\mathbf{z}$: Projection vector
$<.,.>$: Inner product

I

Preliminaries

PART I provides readers with preliminary knowledge and basic background to make this book self-contained. It comprises five chapters.

Chapter 1 covers fundamentals of subsample and mixed sample analyses. Chapter 2 addresses one of the challenges in hyperspectral imaging, that is how to deal with subpixels and mixed pixels often encountered in hyperspectral imagery. Since hyperspectral data are not necessarily image data, more generic terms, subsamples and mixed samples instead of subpixels and mixed pixels, are used to indicate that sample data can be either image pixels or spectral signatures.

Chapter 3 develops a new evaluation tool, a 3D ROC analysis, that extends the detection performance-based 2D receiver operating characteristics (ROC) curves commonly used in detection theory to measure estimation performance. As described in Chapter 2, most hyperspectral imaging techniques are developed as estimators rather than detectors to estimate abundance fractions of target substances for various tasks such as discrimination, detection, classification, etc. As a result, target detection that makes hard decisions is actually performed by target estimation that makes soft decisions in which case the 2D ROC analysis is not applicable unless these real-valued abundance fractions are thresholded to make hard decisions. The 3D ROC analysis is developed to meet this need by including a third dimension to convert a soft-decision-based estimator to a hard-decision-based detector.

Chapter 4 describes a set of synthetic image experiments that simulate two types of target insertion into image backgrounds, target implantation and target embededness, each of which has three different scenarios. As a result, a total of six scenarios can be used for quantitative study and analysis. The need of synthetic images arises from algorithm design where an objective quantitative analysis is required to substantiate an algorithm as well as to evaluate an algorithm compared to other algorithms for performance assessment.

Chapter 5 revisits the concept of virtual dimensionality (VD) that was previously coined in Chang (2003a) and defined as the number of spectrally distinct signatures present in the hyperspectral imagery. Since it was introduced, VD has become very useful and found in a wide range of applications (Chang and Du, 2004; Chang, 2006a, 2006b), such as dimensionality reduction (Wang and Chang, 2006a), band selection (Chang and Wang, 2006), endmember extraction (Chang and Plaza, 2006; Chang et al., 2006; Chaudhry et al. 2006; Wang and Chang, 2006b; Plaza and Chang, 2005, 2006), unsupervised target detection (Wang and Chang, 2006c), and unsupervised image classification (Chang et al., 2010, 2011). With its potential, VD can be interpreted in many

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

ways. For example, an endmember can be also considered as a spectrally distinct signature or a target signature of interest. It is generally used to specify a spectral class, specifically a signature to be used by linear spectral mixture analysis (LSMA) for spectral unmixing. However, on many occasions an endmember may also appear as an anomaly. On the other hand an anomaly may be a mixed pixel and may not be necessarily an endmember and vice versa. Therefore, using the same value of VD to estimate the number of signatures of different types does not seem appropriate even if the estimate provides a good and close estimate. In order to address this issue, VD must vary with its applications from an exploitation point of view; this has been further explored in Chang et al. (2010, 2011). Chapter 5 generalizes the concept of VD to explore its utility in data exploitation where a comprehensive set of techniques are developed and studied.

Chapter 6 develops techniques to address another challenging task, that is how to cope with enormous data volumes resulting from hundreds of spectral bands. A general approach is to perform dimensionality reduction (DR) as a preprocessing step prior to data processing so that the original data can be reduced to a manageable low-dimensional data space. There are two approaches to DR, DR by transform (DRT) and DR by band selection (DRBS), each of which is discussed in detail in this chapter.

2

Fundamentals of Subsample and Mixed Sample Analyses

The issues of subpixels and mixed pixels, which have been briefly discussed in Chapter 1, are crucial in hyperspectral data exploitation. Dealing with these issues is considered to be very challenging to hyperspectral image analysts, primarily due to the fact that techniques available in the traditional pure pixel-based image processing are generally not directly applicable or ineffective if they are blindly applied to hyperspectral signal and image processing. The main reason is that the pure pixel-based image analysis is usually carried out by hard (discrete) decisions in the sense that only a finite number of values are available for decision, while the subpixel and mixed pixel analyses are generally performed by soft decisions in the sense that a decision is specified by abundance fraction that is usually a real value. In other words, the relationship between a soft decision and a hard decision is similar to the relationship between an analog signal and a digital signal. With this interpretation, the process of converting a soft decision into a hard decision can be considered as an analog-to-digital (A/D) converter commonly used in communications and signal processing. This chapter reviews fundamentals of subpixel and mixed pixel analyses in hyperspectral data applications to detection and classification from a perspective of hard and soft decision-making processes. To provide a general context, the terms subsample and mixed sample will be used in this chapter instead of commonly used terms subpixel and mixed pixel, to reflect the nature of sample vectors that can be either hyperspectral image pixels or hyperspectral signals such as signatures from spectral library or database that do not appear as a form of pixels.

2.1 Introduction

A subsample target can appear in two different forms. One is a target that is smaller than the sample spatial resolution, in which case the target is embedded in a sample. An example is a vehicle with size of 8 m × 4 m that can be completely embedded in a single pixel with resolution of 20 m × 20 m. Another is a target that partially occupies a sample with certain amount of abundance fraction. That is, a target may or may not have its size greater than the sample resolution but it occupies more than one sample with partial abundance fractions. In either case, such a target is considered as a subsample target since it does not fully occupy an entire sample. It should be noted that as shown in Sections 30.2–30.3 in Chapter 30 the abundance fraction of a target contained in a sample can be used to calculate the partial size of the target occupying the sample.

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

A mixed sample is considered as a sample mixed with a number of target substances with appropriate abundance portions that account for the entire sample. A distinction between a subsample target and a mixed sample is that the latter must have the knowledge of all the target substances present in the sample compared to the former that only needs to know that the target of interest resides in the sample while discarding the information of other target substances in the sample, which is considered to be the background information to the target.

The concept of subsample and mixed sample analyses can be illustrated by following a simple example. Assume that there are five different fruits: apple, banana, lemon, orange, and strawberry. Each of these five fruits is sliced to a small piece and mixed in a blender. After mixing these small pieces of the five fruits, the resulting mixed juice can be viewed as a full sample and each of the five fruits is considered as a target that is present in the mixed juice. Now, someone is asked to taste a small amount of this mixed juice and answer the following questions:

1. Mixed sample classification

   The answers to a series of five questions of "*Is* there apple in the mixed juice?," "*Is* there orange in the mixed juice?," "*Is* there lemon in the mixed juice?," "*Is* there strawberry in the mixed juice?," and "*Is* there banana in the mixed juice?" that exhaust all the five fruits by knowing these five fruits as prior knowledge are "mixed sample classification."
2. Mixed sample identification

   The answer to a question of "*Which* fruit is in the mixed juice?" is "mixed sample identification." In this case, we know all the five fruits in the mixed juice, but we do not know "*which one*." In general, identification is to identify an unknown target substance without prior knowledge. It does not need a database to perform its task. When the identification is performed via a database, it actually performs verification that verifies an unknown target substance from a known database or spectral library. Unfortunately, the term "identification" used in signal processing is somewhat abused when the verification is used. For example, we know all the five fruits in the mixed juice, but we do not know "*which one*." So, in hyperspectral data processing the "identification is actually meant for "verification."
3. Mixed sample quantification

   The answers to a series of five questions of "*How much* concentration of the apple is in the mixed juice?," "*How much* concentration of the orange is in the mixed juice?," "*How much* concentration of the lemon is in the mixed juice?," "*How much* concentration of the strawberry is in the mixed juice?," and "*How much* concentration of the banana is in the mixed juice?" that exhaust all the five fruits by knowing these five fruits as prior knowledge are "mixed sample quantification." This task can be considered as either a subsequent process after mixed sample classification or a stand-alone process.

On the other hand, consider a glass of plain water that mixes with only one of the five fruit juices, say apple juice. In this case, the apple juice and water are considered as a subsample target and the background, respectively. Now if one drinks a glass of water mixed with an unknown fruit juice or a mixed juice without knowing other juice components, the following questions being asked are subsample analysis.

1. "*Is* there apple in the water?". This is "subsample target detection" where the apple is a subsample target embedded in the background water.
2. "*What* fruit juice is in the water?" without assuming any prior knowledge at all. This is "subsample identification." Unlike subsample target detection where a specific target substance is of interest, subsample identification does not know what substances in the plain water,

specifically the fruit juice to be identified may not be one of the five fruits (apple, banana, lemon, orange, and strawberry).

3. "*What* fruit juice is in the water?" via a database or library. This is "subsample verification." As noted above in the mixed sample analysis the key difference between subsample identification and subsample verification is that the former must identify a target substance without any prior knowledge, whereas the latter needs a database or library to identify the target substances via the provided database/library in which case the used database/library is the required prior knowledge. For example, if the database/library is made up of the five fruits, apple, banana, lemon, orange, and strawberry, the target substances to be verified must be one of these five fruits. So, generally speaking, subsample identification is much more challenging than subsample verification. Unfortunately, most problems claimed to be identification problems in reported literature are actually verification problems.
4. "*How much* concentration of the apple juice is in the water?" This is "subsample quantification" in which case, the apple juice is the only target knowledge is known *a priori*. The main and only interest is the amount of the apple juice contained in the plain water, which is the concentration of the apple juice. This task can be considered either as a subsequent process after subsample detection if the target substance is not known or a stand-alone process if the target substance of interest is known in advance.

As an alternative, we can also use the above mixed juice as another example where subsample analysis is performed by assuming only one of the five fruits (apple, banana, lemon, orange, and strawberry) to be known, while the other four fruits are assumed to be unknown and considered to be background to the known subsample target. In order to perform quantification and classification, the complete prior knowledge of target substances must be known *a priori* as opposed to discrimination and identification, both of which do not need any target knowledge at all. The detection is right in between and only needs to know the target substances of interest while discarding all other information. Additionally, despite that subsample target classification is also considered in the literature, it will be considered part of mixed sample classification here because a subsample target substance in this case is considered as a target partially occupying a sample with a certain proportion of its presence and mixed with other target substances present in the same sample. Finally, it should be noted that the above juice-mixing examples are only used for an illustrative purpose where the mixing is not linear but rather nonlinear as was demonstrated in Guilfoyle (2003) and Guilfoyle et al. (2001, 2002). Details of nonlinear mixing can be found in Guilfoyle (2003).

2.2 Subsample Analysis

The most fundamental task in subsample analysis is subsample detection where two types of detectors will be discussed in this section, detectors with hard decisions and detectors with soft decisions, which correspond to pure-sample target detector and subsample target detector, respectively.

2.2.1 Pure-Sample Target Detection

Despite that a subsample target may be present in a sample, the pure-sample target detection is performed by forcing a detector to make a binary decision, whether the sample is to be detected as target sample or not. In other words, the pure-sample target detection can only say "yes" if target is detected and "no" if target is absent. So, even though a subsample target does not fully occupy the entire sample, it must be claimed to be a pure target sample if a detector says "yes" as target is

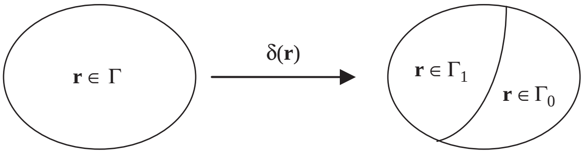


Figure 2.1 A decision rule specified by a partition of Γ, $\Gamma = \Gamma_0 \cup \Gamma_1$.

detected. To emphasize such a nature, the commonly used binary hypotheses-based detectors described in the following are called pure-sample target detectors.

A classical approach to pure-sample target detection is to formulate a signal detection problem as the following binary hypothesis-testing problem.

$$\begin{aligned} &H_0 : \mathbf{r} = \text{target absent} \approx P_0(\mathbf{r}) \\ &\text{versus} \\ &H_1 : \mathbf{r} = \text{target present} \approx P_1(\mathbf{r}) \end{aligned} \tag{2.1}$$

where $\mathbf{r}$ is an observable random variable; the null hypothesis H_0 and alternative hypothesis H_1 represent the case of target absence and the case of target presence with their probability distributions specified by $P_0(\mathbf{r})$ and $P_1(\mathbf{r})$, respectively. A decision rule $\delta(\mathbf{r})$ for H_0 versus H_1 specified by (2.1) is a partition of the observation space Γ into two regions: Γ_1 referred to as rejection region and Γ_0 referred to as acceptance region. By virtue of the defined Γ_0 and Γ_1 with $\Gamma = \Gamma_0 \cup \Gamma_1$, the decision rule $\delta(\mathbf{r})$ is generally described by

$$\delta(\mathbf{r}) = \begin{cases} 1; & \mathbf{r} \in \Gamma_1 \\ 0; & \mathbf{r} \in \Gamma_0 \end{cases} \tag{2.2}$$

and is shown in Figure 2.1.

Now, a solution to (2.2) is to find a best partition in some optimal sense. Specifically, we introduce a cost function specified by a cost matrix $\begin{bmatrix} c_{00} & c_{01} \\ c_{10} & c_{11} \end{bmatrix}$ where c_{ij} is the cost of the decision saying H_i when H_j is actually true and a risk function of the decision rule $\delta(\mathbf{r})$ under hypothesis H_j, $R_j(\delta)$ given by

$$R_j(\delta) = c_{0j}P_j(\Gamma_0) + c_{1j}P_j(\Gamma_1) \tag{2.3}$$

Suppose that the prior probabilities of H_0 and H_1 for (2.1) are specified by π_0 and π_1, respectively. The averaged risk $r(\delta)$ is then defined by

$$r(\delta) = \pi_0 R_0(\delta) + \pi_1 R_1(\delta) \tag{2.4}$$

Minimizing (2.4) over all possible decision rules yields the Bayes detector $\delta^{\text{Bayes}}(\mathbf{r})$ given by

$$\delta^{\text{Bayes}}(\mathbf{r}) = \arg\{\min_\delta r(\delta)\}. \tag{2.5}$$

The detector of this type can be considered as pure sample-based signal detection. If we further assume that the probability distributions $P_0(\mathbf{r})$ and $P_1(\mathbf{r})$ have their own probability density

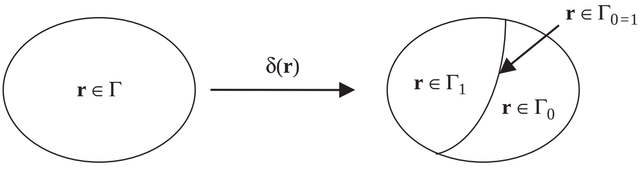

$\mathbf{r} \in \Gamma$
$\delta(\mathbf{r})$
$\mathbf{r} \in \Gamma_1$
$\mathbf{r} \in \Gamma_0$
$\mathbf{r} \in \Gamma_{0=1}$

Figure 2.2 A randomized decision rule specified by a partition of Γ, $\Gamma = \Gamma_0 \cup \Gamma_{0=1} \cup \Gamma_1$.

functions given by $p_0(\mathbf{r})$ and $p_1(\mathbf{r})$, then $P_j(\Gamma_i) = \int_{\Gamma_i} p_j(\mathbf{r})d\mathbf{r}$. It can be shown in Poor (1994) that the solution to (2.5) can be

$$\delta^{\text{Bayes}}(\mathbf{r}) = \begin{cases} 1; \ \Lambda(\mathbf{r}) \geq \tau \\ 0; \ \Lambda(\mathbf{r}) < \tau \end{cases} \tag{2.6}$$

where $\Lambda(\mathbf{r})$ is the likelihood ratio test (LRT) given by $p_1(\mathbf{r})/p_0(\mathbf{r})$ and the threshold τ is given by $\tau = \frac{\pi_0(c_{10}-c_{00})}{\pi_1(c_{01}-c_{11})}$. It should be noted that the Bayes detector in (2.6) declares H_1 when the LRT $\Lambda(\mathbf{r})$ equals the threshold τ. While this is generally true, it is not necessarily correct, particularly when the random variable $\mathbf{r}$ is not continuous, but discrete. In order to address this issue, let $\Gamma_{0=1} = \{\mathbf{r} \in \Gamma | \Lambda(\mathbf{r}) = \tau\}$ and Figure 2.1 becomes Figure 2.2, where $\Gamma = \Gamma_0 \cup \Gamma_{0=1} \cup \Gamma_1$ and κ is the probability that H_1 is true when $\Lambda(\mathbf{r})$ is equal to the threshold τ.

The risk function (2.3) can be further modified as

$$\begin{aligned} R_1(\delta) &= c_{01}[P_1(\Gamma_0) + (1-\kappa)P_1(\Gamma_{0=1})] + c_{11}[P_1(\Gamma_1) + \kappa P_1(\Gamma_{0=1})] \\ R_0(\delta) &= c_{00}[P_0(\Gamma_0) + (1-\kappa)P_0(\Gamma_{0=1})] + c_{10}[P_0(\Gamma_1) + \kappa P_0(\Gamma_{0=1})] \end{aligned} \tag{2.7}$$

As a result of (2.7) the Bayes rule in (2.6) becomes a randomized detector $\delta^{\text{Bayes}}(\mathbf{r})$ specified by the following form:

$$\delta^{\text{Bayes}}(\mathbf{r}) = \begin{cases} 1; & \quad > \tau \\ \kappa; & \Lambda(\mathbf{r}) = \tau \\ 0; & \quad < \tau \end{cases} \tag{2.8}$$

It is worth noting that it is the threshold τ that determines which type of a detector will be. When τ is completely specified by a cost function and prior probabilities of H_0 and H_1 as the case of (2.3) and (2.4), the detector is a Bayes detector. When the τ is only specified by a cost function with no knowledge of prior probabilities of H_0 and H_1, the detector is a minimax detector. A most practical case is that there is no knowledge about the cost function or prior probabilities of H_0 and H_1. Under such circumstances, the τ is determined by a prescribed false alarm probability, P_{F} and the resultant detector becomes a well-known detector, called Neyman–Pearson (NP) detector. More specifically, for a detector, δ let $P_{\text{F}}(\delta)$ be the false alarm probability given by

$$P_{\text{F}}(\delta) = P_0(\Gamma_1) = \int_{\Lambda(\mathbf{r})\geq\tau} p_0(\mathbf{r})d\mathbf{r} \tag{2.9}$$

and $P_{\mathrm{D}}(\delta)$ be the detection probability or detection power given by

$$P_{\mathrm{D}}(\delta) = P_1(\Gamma_1) = \int_{\Lambda(\mathbf{r})\geq\tau} p_1(\mathbf{r})d\mathbf{r} \tag{2.10}$$

The NP detector is one, denoted by $\delta^{\mathrm{NP}}(\mathbf{r})$ to solve

$$\max_{\delta}\{P_{\mathrm{D}}(\delta)\} \text{ subject to } P_{\mathrm{F}}(\delta) \leq \beta \text{ for any given } 0 \leq \beta \leq 1 \tag{2.11}$$

where β is known as the significant level of test. In order to evaluate the detection performance of $\delta^{\mathrm{NP}}(\mathbf{r})$, a receiver operating characteristic (ROC) curve is plotted as a function of P_{D} versus P_{F} for analysis (more details can be found in Chapter 3). Interestingly, no matter which detector is derived as above, the structure of the detector always turns out to be the LRT. In other words, all the Bayes, minimax, or Neyman–Pearson detectors end up with the same form of LRT. Details of signal detection theory can be found in Poor (1994).

As mentioned earlier, in many cases where the probability distributions $p_0(\mathbf{r})$ and $p_1(\mathbf{r})$ are continuous such as Gaussian distributions, the detector $\delta^{\mathrm{Bayes}}(\mathbf{r})$ in (2.8) can be always made a deterministic detector, $\delta(\mathbf{r})$ by setting $\kappa = 1$ with no effect on detection performance. In this case, (2.8) can be simplified and reduced to (2.6).

The hypothesis-testing problem described by (2.1) is a general setting for a detection problem where no signal model is assumed. However, if (2.1) is considered for signal detection in noise, the hypothesis-testing problem (2.1) can be specifically represented by

$$\begin{aligned} &H_0: \quad \mathbf{r} = \mathbf{n} \\ &\text{versus} \\ &H_1: \quad \mathbf{r} = \mathbf{s} + \mathbf{n} \end{aligned} \tag{2.12}$$

where $\mathbf{s}$ is the signal of interest and $\mathbf{n}$ represents an additive noise. Of particular interest is the case that the noise in (2.12) is Gaussian in which case the Bayes decision rule (2.6) becomes a well-known matched filter with the matching signal specified by the signal $\mathbf{s}$.

2.2.2 Subsample Target Detection

In order to apply the signal detection model (2.12) to subsample target detection, we assume that a subsample target signal specified by signature $\mathbf{t}$ is embedded in the background $\mathbf{b}$ with their proportions specified by α and $1-\alpha$, respectively, where the proportion α will be referred to as abundance fraction of $\mathbf{t}$. As a result, the signal $\mathbf{s}$ in (2.12) will be replaced by a subsample target signal $\mathbf{t}$ with its proportion occupied in $\mathbf{r}$ specified by an abundance fraction, α mixed with its background signature $\mathbf{b}$ with the abundance fraction, $1-\alpha$, that is, $\alpha\mathbf{t} + (1-\alpha)\mathbf{b}$. The signal detection model in (2.12) becomes

$$\begin{aligned} &H_0: \quad \mathbf{r} = \mathbf{b} + \mathbf{n} \approx \text{background with no subpixel target} \\ &\text{versus} \\ &H_1: \quad \mathbf{r} = [\mathbf{t}\,\mathbf{b}]\begin{bmatrix}\alpha \\ 1-\alpha\end{bmatrix} + \mathbf{n} = \alpha\mathbf{t} + (1-\alpha)\mathbf{b} + \mathbf{n} \approx \text{subpixel target presence} \end{aligned} \tag{2.13}$$

Using (2.13) subsample target detection is performed by the LRT given by

$$\Lambda(\mathbf{r}) = \frac{p_1(\mathbf{r})}{p_0(\mathbf{r})} = \frac{p(\mathbf{r}|\text{subsample target})}{p(\mathbf{r}|\text{no subsample target})} \tag{2.14}$$

Unlike pure sample-based signal detection specified by (2.12) using a threshold τ, the LRT, $\Lambda(\mathbf{r})$, in (2.14) detects the subsample target $\mathbf{t}$ by estimating the abundance fraction of $\mathbf{t}$, α present in $\mathbf{r}$. Since the amount detected by a detector specified by LRT is proportional to the abundance fraction of α contained in the sample $\mathbf{r}$, LRT essentially serves as an estimator of α, $\hat{\alpha}(\mathbf{r})$ where the $\mathbf{r}$ is included in $\hat{\alpha}(\mathbf{r})$ to indicate the dependency of the abundance estimate on the $\mathbf{r}$. By virtue of (2.14), a subsample target detector can be interpreted as a detector that makes a soft decision based on its estimated abundance $\hat{\alpha}(\mathbf{r})$ instead of the one in (2.6) or (2.8) that makes a hard decision based on a threshold τ. The detector of this type can be considered as subsample signal detection with *soft decisions* via the estimation of abundance fraction α, $\hat{\alpha}(\mathbf{r})$ through $\mathbf{r}$ as opposed to pure sample-based signal detection with *hard decisions* determined by the threshold τ. A similar concept will also be explored in Sections 2.3.1 and 2.3.2.

Since the major focus of subsample analysis is on the subsample target of interest $\mathbf{t}$, the background $\mathbf{b}$ is generally not known and is something we would like to remove or suppress in order to improve detectability of the $\mathbf{t}$. In doing so, two general approaches have been proposed in the past. One is to obtain the background knowledge from a secondary data set originally proposed by Kelly (1986), and the other is to extend a model in (2.13) to a signal-background-noise (SBN) model proposed in Thai and Healey (2002) and signal-decomposed and interference/noise (SDIN) model suggested in Du and Chang (2004).

2.2.2.1 Adaptive Matched Detector (AMD)

As a special case of (2.12) where both probability density functions $p_0(\mathbf{r})$ and $p_1(\mathbf{r})$ are Gaussian distributions specified by $p_0(\mathbf{r}) = N(\bar{\mathbf{b}}, \mathbf{K})$ and $p_1(\mathbf{r}) = N(\bar{\mathbf{t}}, \mathbf{K})$ with the same covariance matrix $\mathbf{K}$ and the background mean, $\bar{\mathbf{b}}$ and target mean, $\bar{\mathbf{t}}$, respectively, (2.14) becomes

$$\begin{aligned} \Lambda(\mathbf{r}) = \frac{p_1(\mathbf{r})}{p_0(\mathbf{r})} &= \exp\left[(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T \mathbf{K}^{-1}(\mathbf{r} - \boldsymbol{\mu})\right] = \exp\left\{\left[\left(\mathbf{K}^{-1}\right)(\bar{\mathbf{t}} - \bar{\mathbf{b}})\right]^T (\mathbf{r} - \boldsymbol{\mu})\right\} \\ \Rightarrow \log\Lambda(\mathbf{r}) &= \log\left[\frac{p_1(\mathbf{r})}{p_0(\mathbf{r})}\right] = \left[\left(\mathbf{K}^{-1}\right)(\bar{\mathbf{t}} - \bar{\mathbf{b}})\right]^T (\mathbf{r} - \boldsymbol{\mu}) \\ &= \left[\left(\mathbf{K}^{-1}\right)(\bar{\mathbf{t}} - \bar{\mathbf{b}})\right]^T \mathbf{r} - (\bar{\mathbf{t}} - \bar{\mathbf{b}})^T \mathbf{K}^{-1}\boldsymbol{\mu} \\ &= \left[\left(\mathbf{K}^{-1}\right)(\bar{\mathbf{t}} - \bar{\mathbf{b}})\right]^T \mathbf{r} - (1/2)(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T \mathbf{K}^{-1}(\bar{\mathbf{t}} + \bar{\mathbf{b}}) \end{aligned} \tag{2.15}$$

Because $(1/2)(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T \mathbf{K}^{-1}(\bar{\mathbf{t}} + \bar{\mathbf{b}})$ does not depend upon the observation $\mathbf{r}$ and can be absorbed in the threshold τ in (2.8) to produce a new threshold defined by

$$\tau' = \log\tau + (1/2)(\bar{\mathbf{t}} - \bar{\mathbf{b}})\mathbf{K}^{-1}(\bar{\mathbf{t}} + \bar{\mathbf{b}}) \tag{2.16}$$

we can define a detector $\delta^{\text{AMD}}(\mathbf{r})$, called adaptive matched detector (AMD) via (2.15) to estimate the abundance α by

$$\delta^{\text{AMD}}(\mathbf{r}) = \hat{\alpha}(\mathbf{r}) = \left[\left(\mathbf{K}^{-1}\right)(\bar{\mathbf{t}} - \bar{\mathbf{b}})\right]^T \mathbf{r} = (\bar{\mathbf{t}} - \bar{\mathbf{b}})^T \mathbf{K}^{-1}\mathbf{r} \tag{2.17}$$

which is a matched filter with the matching signal specified by $(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T\mathbf{K}^{-1}$. However, $(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T\mathbf{K}^{-1} = \bar{\mathbf{t}}^T\mathbf{K}^{-1} - \bar{\mathbf{b}}^T\mathbf{K}^{-1}$ is actually the difference between two whitened means by the covariance matrix $\mathbf{K}$; the matching signal is simply the difference of two means after the data is whitened. If we calculate the variances of the detector of (2.17) under each of two hypotheses, H_0 and H_1 in (2.13), by

$$\sigma^2_{\mathrm{AMD}}\big(\delta(\mathbf{r})|H_j\big) = (\bar{\mathbf{t}} - \bar{\mathbf{b}})\mathbf{K}^{-1}(\bar{\mathbf{t}} - \bar{\mathbf{b}}) \quad \text{for} \quad j = 0, 1 \tag{2.18}$$

their variances turn out to be identical and are independent of hypotheses. Most interestingly, the variance specified by (2.18) is actually the Mahalanobis distance between background mean and target mean.

If we further use (2.18) to normalize AMD in (2.17), the resulting detector is referred to as normalized AMD (NAMD), $\delta^{\mathrm{NAMD}}(\mathbf{r})$ and given by

$$\delta^{\mathrm{NAMD}}(\mathbf{r}) = \frac{\delta(\mathbf{r})}{\sigma^2_{\mathrm{AMD}}\big(\delta(\mathbf{r})|H_j\big)} = \frac{(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T\mathbf{K}^{-1}\mathbf{r}}{(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T\mathbf{K}^{-1}(\bar{\mathbf{t}} - \bar{\mathbf{b}})} \tag{2.19}$$

which becomes the commonly used adaptive matched filter. It should be noted that using the variance in (2.18) as a scaling constant in (2.19) has significant impact on the estimation of α. It has been shown in Chang (1998) and Chang (2003a) that it was this constant to correct estimation error of α. Unfortunately, this constant has been generally referred to as a normalization constant in the literature, which is somewhat misleading. So, in order to further estimate the abundance of α in (2.13) more accurately, we let $\hat{\alpha}^{\mathrm{AMD}}(\mathbf{r}) = \delta^{\mathrm{NAMD}}(\mathbf{r})$ as an abundance estimator and use it as a detector given by

$$\delta^{\mathrm{NAMD}}(\mathbf{r}) = \begin{cases} 1; & > \tau'' \\ \eta; \quad \text{if} \quad \dfrac{(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T\mathbf{K}^{-1}\mathbf{r}}{(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T\mathbf{K}^{-1}(\bar{\mathbf{t}} - \bar{\mathbf{b}})} & = \tau'' \\ 0; & < \tau'' \end{cases} \tag{2.20}$$

where $\tau'' = \frac{\tau'}{(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T\mathbf{K}^{-1}(\bar{\mathbf{t}} - \bar{\mathbf{b}})} = \left((\bar{\mathbf{t}} - \bar{\mathbf{b}})^T\mathbf{K}^{-1}(\bar{\mathbf{t}} - \bar{\mathbf{b}})\right)^{-1}\left[\tau + (1/2)(\bar{\mathbf{t}} - \bar{\mathbf{b}})\mathbf{K}^{-1}(\bar{\mathbf{t}} + \bar{\mathbf{b}})\right]$ is a new threshold obtained by absorbing the constant $\left((\bar{\mathbf{t}} - \bar{\mathbf{b}})^T\mathbf{K}^{-1}(\bar{\mathbf{t}} - \bar{\mathbf{b}})\right)^{-1}$ into the threshold $\tau' = \log\tau + (1/2)(\bar{\mathbf{t}} - \bar{\mathbf{b}})\mathbf{K}^{-1}(\bar{\mathbf{t}} + \bar{\mathbf{b}})$ defined in (2.16) with τ defined in (2.8).

An alternative approach to arriving at the same detector in (2.20) is to perform a whitening process on the original data by implementing a linear transformation specified by

$$\hat{\mathbf{r}} = \xi_{\mathbf{A}}(\mathbf{r}) = \mathbf{A}^T(\mathbf{r} - \bar{\mathbf{b}}) = \frac{\mathbf{K}^{-1/2}(\mathbf{r} - \bar{\mathbf{b}})}{\sqrt{(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T\mathbf{K}^{-1}(\bar{\mathbf{t}} - \bar{\mathbf{b}})}} \tag{2.21}$$

where $\mathbf{A}$ is referred to as a whitening matrix defined by

$$\mathbf{A} = \frac{\mathbf{K}^{-1/2}(\bar{\mathbf{t}} - \bar{\mathbf{b}})}{\sqrt{(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T\mathbf{K}^{-1}(\bar{\mathbf{t}} - \bar{\mathbf{b}})}} \tag{2.22}$$

As a result of (2.22), the transformed data have zero mean and variance normalized to unity where the standard deviation of (2.18) used as the denominator in (2.21) to normalize the original data **r** has the same effect as does the variance in (2.18) used to normalize the detector in (2.19).

In many real applications the statistics of the noise **n**, **K** and the knowledge of the background mean $\bar{\mathbf{b}}$ in (2.20) are generally not known in advance. This implies that both the detector and the threshold τ'' in (2.20) cannot be specifically characterized. As a result, the hypothesis-testing problem specified by (2.13) is no longer a simple binary hypothesis-testing problem, but rather a binary composite hypothesis-testing problem where a uniformly most powerful (UMP) detector is sought to optimize detection performance. Unfortunately, such a UMP detector generally does not exist. A general approach is to extend LRT, $\Lambda(\mathbf{r})$, in (2.20) to so-called generalized LRT, which leads to a maximum likelihood detector. In order to solve GLRT, a common assumption made on the noise is Gaussian so that the maximum likelihood detector can be derived. On the other hand, since the cost function and priorities of each hypothesis are also not unknown, the threshold τ used in (2.20) cannot be determined. To resolve this issue, an NP detector is implemented for this purpose. However, as noted, the performance of an NP detector is determined by a compromise between P_{D} and P_{F} via the ROC analysis. If an NP detector is designed to perform a UMP detector while its false alarm probability is retained at a constant level, such a detector is called constant false alarm rate (CFAR) detector, which has been widely used in radar and sonar signal processing. On the other hand, in order to obtain the background knowledge $p_0(\mathbf{r}) = N(\bar{\mathbf{b}}, \mathbf{K})$, a secondary data set is also needed to produce required information. Many efforts along with this approach have been reported in the literature, Reed et al. (1974), Kelly (1986), Reed and Yu (1990), Manolakis and Shaw (2002), and so on.

2.2.2.2 Adaptive Subspace Detector (ASD)

The AMD-based subsample target detection discussed in Section 2.2.2.1 follows the standard Neyman–Pearson detection theory by finding GLRT or CFAR detector where the probability distribution under each hypothesis and background knowledge such as noise must be known *a priori*, preferably Gaussian distributions from which a GLRT can be derived and an ROC analysis can be further used for detection performance. As a matter of fact, in reality, such assumptions are generally not true for hyperspectral imagery, despite the fact that CFAR- or GLRT-based approaches seem to perform successfully in subsample analysis. So, it is interesting to find out how can an approach perform well, while its assumptions violate practical constraints? This question will be answered by the following approach.

As noted, AMD assumes that noise or background statistics are given *a priori*. In this section, we consider an alternative approach modified from the subspace detector proposed by Kraut et al. (2001), which can be also considered as a CFAR detector. It only assumes subsample target knowledge without knowing noise/background statistics. It is also referred to as adaptive subspace detector (ASD) due to the fact that it is derived from the concept of subspace projection. However, it is worth noting that the ASD derived here is a little bit different from ASD in Kraut et al. (2001) in the sense that no signal model such as (2.13) in Kraut et al. (2001) is assumed and involved in our derivation. The only assumption made is the prior signature knowledge of the subsample target, **t**. ASD derived in (2.28) is nearly identical to CEM derived in (2.33) except the sample covariance matrix used in ASD is replaced by the sample correlation matrix in CEM.

According to Kraut et al. (2001), it first uses the sample covariance matrix, **K**, to whiten the data and is then followed by a subspace projection approach that projects the entire whitened data space into two separate mutually orthogonal linear subspaces, called signal subspace, denoted by $\langle \hat{\mathbf{t}} \rangle$ and its orthogonal subspace, referred to as clutter space, denoted by $\langle \hat{\mathbf{t}} \rangle^{\perp}$ via two orthogonal subspace projection operators specified by $\hat{\mathbf{t}} = \mathbf{K}^{-1/2}\mathbf{t}$ and defined as follows.

$$P_{\hat{\mathbf{t}}} = \hat{\mathbf{t}}\left(\hat{\mathbf{t}}^T\hat{\mathbf{t}}\right)^{-1}\hat{\mathbf{t}}^T \quad \text{and} \quad \mathbf{I} - P_{\hat{\mathbf{t}}} \tag{2.23}$$

Now, let $\hat{\mathbf{w}} = \mathbf{K}^{-1/2}\mathbf{w}$ be the whitened vector of any weighting vector, $\mathbf{w}$ and the signal-to-clutter ratio (SCR) be defined by

$$\begin{aligned}
\text{SCR}(\hat{\mathbf{w}}) &= \frac{\hat{\mathbf{w}}^T P_{\hat{\mathbf{t}}}\hat{\mathbf{w}}}{\left[\hat{\mathbf{w}}^T\left(\mathbf{I} - P_{\hat{\mathbf{t}}}\right)\hat{\mathbf{w}}\right]} = \frac{\hat{\mathbf{w}}^T P_{\hat{\mathbf{t}}}\hat{\mathbf{w}}}{\hat{\mathbf{w}}^T\hat{\mathbf{w}} - \hat{\mathbf{w}}^T P_{\hat{\mathbf{t}}}\hat{\mathbf{w}}} \\
&= \frac{\hat{\mathbf{w}}^T\mathbf{K}^{-1/2}\mathbf{t}\left(\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t}\right)^{-1}\mathbf{t}^T\mathbf{K}^{-1/2}\hat{\mathbf{w}}}{\hat{\mathbf{w}}^T\hat{\mathbf{w}} - \hat{\mathbf{w}}^T\mathbf{K}^{-1/2}\mathbf{t}\mathbf{t}^T\mathbf{K}^{-1/2}\hat{\mathbf{w}}\left(\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t}\right)^{-1}} \\
&= \frac{\left(\mathbf{t}^T\mathbf{K}^{-1/2}\hat{\mathbf{w}}\right)^2\left(\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t}\right)^{-1}}{\hat{\mathbf{w}}^T\hat{\mathbf{w}} - \left(\mathbf{t}^T\mathbf{K}^{-1/2}\hat{\mathbf{w}}\right)^2\left(\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t}\right)^{-1}}
\end{aligned} \tag{2.24}$$

Maximizing (2.25) over $\mathbf{w}$ is equivalent to finding a vector $\mathbf{w}$ minimizing

$$\begin{aligned}
\frac{1}{\text{SCR}(\hat{\mathbf{w}})} &= \frac{\hat{\mathbf{w}}^T\hat{\mathbf{w}} - \left(\mathbf{t}^T\mathbf{K}^{-1/2}\mathbf{w}\right)^2\left(\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t}\right)^{-1}}{\left(\mathbf{t}^T\mathbf{K}^{-1/2}\hat{\mathbf{w}}\right)^2\left(\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t}\right)^{-1}} \\
&= \frac{\hat{\mathbf{w}}^T\hat{\mathbf{w}}}{\left(\mathbf{t}^T\mathbf{K}^{-1/2}\hat{\mathbf{w}}\right)^2\left(\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t}\right)^{-1}} - 1
\end{aligned} \tag{2.25}$$

which is in turn equivalent to finding an optimal vector $\mathbf{w}$ that maximizes

$$\frac{\left(\mathbf{t}^T\mathbf{K}^{-1/2}\hat{\mathbf{w}}\right)^2}{\hat{\mathbf{w}}^T\hat{\mathbf{w}}} = \frac{\left(\hat{\mathbf{w}}^T\mathbf{K}^{-1/2}\mathbf{t}\right)^2}{\hat{\mathbf{w}}^T\hat{\mathbf{w}}} \tag{2.26}$$

Using Schwarz's inequality and following the same argument as (2.6) in Chang (2003a, p. 42), the solution to maximization of (2.26) or (2.24) denoted by $\hat{\mathbf{w}}^*$ can be shown to be

$$\hat{\mathbf{w}}^* = \kappa\mathbf{K}^{-1/2}\mathbf{t} = \arg\left\{\max_{\hat{\mathbf{w}}}\left[\frac{\left(\hat{\mathbf{w}}^T\mathbf{K}^{-1/2}\mathbf{t}\right)^2}{\hat{\mathbf{w}}^T\hat{\mathbf{w}}}\right]\right\} = \arg\{\max_{\hat{\mathbf{w}}}\text{SCR}(\hat{\mathbf{w}})\}$$

with

$$\max_{\hat{\mathbf{w}}}\text{SCR}(\hat{\mathbf{w}}) = \frac{\left[\left(\mathbf{K}^{-1/2}\mathbf{t}\right)^T\left(\mathbf{K}^{-1/2}\mathbf{t}\right)\right]^2}{\left(\mathbf{K}^{-1/2}\mathbf{t}\right)^T\left(\mathbf{K}^{-1/2}\mathbf{t}\right)} = \mathbf{t}^T\mathbf{K}^{-1}\mathbf{t} \tag{2.27}$$

where κ is any constant. Using the weight $\hat{\mathbf{w}}^*$ obtained in (2.27) by letting $\kappa = \left(\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t}\right)^{-1}$, we can define an adaptive subspace detector, $\delta^{\text{ASD}}(\mathbf{r})$ on the original data space by

$$\delta^{\text{ASD}}(\mathbf{r}) = \left(\hat{\mathbf{w}}^*\right)^T\hat{\mathbf{r}} = \frac{\left(\mathbf{K}^{-1/2}\mathbf{t}\right)^T\left(\mathbf{K}^{-1/2}\mathbf{r}\right)}{\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t}} = \frac{\mathbf{t}^T\mathbf{K}^{-1}\mathbf{r}}{\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t}} \tag{2.28}$$

where $\kappa = \left(\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t}\right)^{-1}$ is a scaling constant and is a very important factor in correcting estimation error of the abundance fraction of the subsample target **t** (Chang, 1998).

It is worth mentioning that the SCR defined in (2.24) is slightly different from signal-to-noise ratio (SNR) generally used in signal detection in noise in the sense that the clutter considered in (2.24) may include unwanted signals such as background or interferers that can be treated as structure noise compared to the noise considered in SNR that is assumed to be a random noise and can be viewed as an unstructured noise. Since a subsample target is embedded in a sample specified by its abundance fraction α to reflect its spatial presence, the spatial proportion accounted for background is $(1-\alpha)$. In this case, using SCR is more appropriate than using SNR for subsample target detection.

Comparing (2.28) against (2.19), the **t** and the constant κ in (2.28) play the same role as $(\bar{\mathbf{t}}-\bar{\mathbf{b}})$ and $\left((\bar{\mathbf{t}}-\bar{\mathbf{b}})^T\mathbf{K}^{-1}(\bar{\mathbf{t}}-\bar{\mathbf{b}})\right)^{-1}$ do in (2.19). However, as noted in AMD, a secondary data set is needed to estimate the background so that the background mean $\bar{\mathbf{b}}$ can be removed. In addition, since there is no Gaussian assumption made on the data, target signal $\bar{\mathbf{t}}$ in $p_1(\mathbf{r}) = N(\bar{\mathbf{t}}, \mathbf{K})$ assumed in (2.12) can be replaced by the target signal of interest **t**. With $\bar{\mathbf{b}}$ set to zero and $\bar{\mathbf{t}}$ set to **t**, the $\delta^{\text{AMD}}_{\text{normal}}(\mathbf{r})$ specified by (2.19) is reduced to $\delta^{\text{ASD}}(\mathbf{r})$ in (2.28).

Extensions of ASD from the standard signal detection model in (2.13) to a more general model have been investigated in recent years, such as the ones including background (Thai and Healey, 2002), interference (Du and Chang, 2004), and clutter (Funk et al., 2001). Nevertheless, they can all be considered as variants of the well-known AMD developed by Scharf and Friedlander (1994) and ASD developed by Kraut et al. (2001). Details of such extensions will be discussed in Chapter 12.

Finally, as will be shown in Sections 2.3.2.1 and 12.2.1, ASD is very closely related to the orthogonal subspace projection (OSP).

2.2.3 Subsample Target Detection: Constrained Energy Minimization (CEM)

The ASD presented in Section 2.2.2.2 is derived from subspace projection using maximization of SCR as a criterion for optimality. This section develops a rather different approach, called the constrained energy minimization (CEM) developed in Harsanyi's dissertation (1993) and discussed in great detail in Chang (2003a). It does not assume a signal model or noise/background knowledge. The only knowledge that CEM requires is the subsample target information specified by **t** as the same level of knowledge required by ASD presented in Section 2.2.2.2. Because there is no signal model involved, CEM does not format a detection problem as a binary composite hypothesis testing. So, CEM is not an LRT-based approach such as AMD and ASD. Moreover, CEM does not need to know background information. As a result, no secondary data are required to produce the background data. Therefore, from a practical point of view, CEM is more realistic and general due to the fact that it requires the least amount of information about subsample target of interest without making assumptions on signal model and noise/background statistics.

The CEM owes its idea to the linearly constrained minimum variance (LCMV) originally proposed by Frost (1972) arising in adaptive beamforming. Suppose that a hyperspectral image is represented by a collection of image pixel vectors, denoted by $\{\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_N\}$ where $\mathbf{r}_i = (r_{i1}, r_{i2}, \cdots, r_{iL})^T$ for $1 \le i \le N$ is an L-dimensional pixel vector, N is the total number of pixels in the image, and L is the total number of spectral channels. Further assume that $\mathbf{t} = (t_1, t_2, \cdots, t_L)^T$ is specified by the target signal of interest to be used for detection. The goal is to find a target detector detecting data samples that contain the desired target signal specified by signature **t**. Instead of directly appealing for an LRT-based detector, AMD, or a subspace projection-based detector, ASD, an LCMV-based adaptive beamforming approach is used for this purpose. It assumes that the signals arriving at an array from the desired direction will be passed through an adaptive beamformer, while the energies of signals coming from other directions will be minimized at the output of the beamformer. Now, if we interpret the desired direction as the desired signature that specifies targets to be detected and the beamformer's output as a soft decision-maker for target detection, a soft target detector can actually be deigned by a finite impulse response (FIR) linear filter with L filter coefficients $\{w_1, w_2, \cdots, w_L\}$, denoted by an L-dimensional vector $\mathbf{w} = (w_1, w_2, \cdots, w_L)^T$ that minimizes the filter output energy subject to the constraint $\mathbf{t}^T\mathbf{w} = \mathbf{w}^T\mathbf{t} = 1$. More specifically, let y^i denote the output of the designed FIR filter resulting from the input $\mathbf{r}^i$. Then y^i can be expressed by

$$y_i = \sum\nolimits_{l=1}^{L} w_l r_{il} = (\mathbf{w})^T \mathbf{r}_i = \mathbf{r}_i^T \mathbf{w} \tag{2.29}$$

and the average energy of the filter output is given by

$$(1/N)\sum\nolimits_{i=1}^{N} y_i^2 = (1/N)\sum\nolimits_{i=1}^{N} \left(\mathbf{r}_i^T\mathbf{w}\right)^2 = \mathbf{w}^T\left[(1/N)\sum\nolimits_{i=1}^{N} \mathbf{r}_i\mathbf{r}_i^T\right]\mathbf{w} = \mathbf{w}^T\mathbf{R}\mathbf{w} \tag{2.30}$$

where $\mathbf{R} = \frac{1}{N}\left[\sum_{i=1}^{N} \mathbf{r}^i\mathbf{r}_i^T\right]$ is the sample auto-correlation matrix of the image. The CEM is developed to solve the following linearly constrained optimization problem

$$\min_{\mathbf{w}}\left\{\mathbf{w}^T\mathbf{R}\mathbf{w}\right\} \text{ subject to } \mathbf{t}^T\mathbf{w} = \mathbf{w}^T\mathbf{t} = 1 \tag{2.31}$$

The optimal solution to (2.31) can be derived in Harsanyi (1993) and Chang (2002) by

$$\mathbf{w}^{\text{CEM}} = \frac{\mathbf{R}^{-1}\mathbf{t}}{\mathbf{t}^T\mathbf{R}^{-1}\mathbf{t}} \tag{2.32}$$

With the optimal weight $\mathbf{w}^{\text{CEM}}$ specified by (2.32), a filter called CEM, denoted by $\delta^{\text{CEM}}(\mathbf{r})$, was derived in Harsanyi (1993) and given by

$$\delta^{\text{CEM}}(\mathbf{r}) = \left(\mathbf{w}^{\text{CEM}}\right)^T\mathbf{r} = \left(\frac{\mathbf{R}^{-1}\mathbf{t}}{\mathbf{t}^T\mathbf{R}^{-1}\mathbf{t}}\right)^T\mathbf{r} = \frac{\mathbf{t}^T\mathbf{R}^{-1}\mathbf{r}}{\mathbf{t}^T\mathbf{R}^{-1}\mathbf{t}} \tag{2.33}$$

which is also a matched filter and turns out to be ASD in (2.28) with the covariance matrix $\mathbf{K}$ replaced by the correlation matrix $\mathbf{R}$. Therefore, except for the sample covariance matrix used in ASD and the sample correlation matrix implemented in CEM, both ASD and CEM are essentially identical to each other in terms of detector structure regardless of the fact that the design rationales for ASD and CEM are quite different.

By further comparing (2.33) to (2.19), the only difference between (2.33) and (2.19) is that the covariance matrix $\mathbf{K}$ and $(\bar{\mathbf{t}} - \bar{\mathbf{b}})$ used in (2.19) can be simply replaced with $\mathbf{R}$ and $\mathbf{t}$ in (2.33), respectively. This implies that two different approaches, LRT and LCMV, arrive at the same form of a detector with the same matching signal specified by $\mathbf{t}$. As a consequence, they both give rise to similar performance. This is the major reason why AMD can perform well even if it is derived from Gaussian noise statistics.

2.3 Mixed Sample Analysis

In analogy with subsample analysis presented in Section 2.2, mixed sample analysis can be performed similarly with one key difference. In subsample analysis the background knowledge about the **b** considered in (2.13) is assumed to be unknown or can be obtained *a posteriori* directly from the data or a secondary data set, whereas in mixed sample analysis the target knowledge present in the **r** must be completely specified *a priori*. One fundamental task of mixed sample analysis is to perform spectral unmixing via linear spectral mixture analysis (LSMA). More specifically, LSMA assumes that a data sample vector **r** can be specified by a set of p known signals, $\{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_p\}$ as a linear mixture in terms of their respective mixing abundance fractions specified by $\alpha_1, \alpha_2, \cdots, \alpha_p$. When LSMA is used for classification, it classifies the **r** into one of these p signals, $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_p$. Using the five-fruit mixed juice described in the introduction as an example, the classification of mixed sample analysis is performed by assuming that the five fruits (apple, banana, lemon, orange, and strawberry) represent five known signals $\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_5$ and the mixed juice **r** is an admixture of these five fruits with different concentrations $\alpha_1, \alpha_2, \cdots, \alpha_5$. The classification of the mixed juice **r** is then performed in two ways. One is to determine which fruit is more likely to represent the mixed juice **r**, in which case the classification is performed by hard decisions. The other is to determine how much concentration of each fruit is contained in the mixed juice **r**, in which case the classification is performed by soft decisions via the concentration of each fruit estimated from the **r** as its abundance fraction. In analogy with pure sample-based signal detection making hard decisions and subsample-based signal detection making soft decisions that have been explored in Section 2.2 for detection, two similar types of classification, classification with hard decisions and classification with soft decisions, can be also derived in the same manner. The former can be considered as pure sample-based classification that labels a data sample vector **r** by a specific class, whereas the latter is mixed sample-based classification that estimates abundance fraction of each class present in a sample vector **r** as a likelihood of a particular class to be assigned to the **r**. More specifically, if there are p classes of interest to be assigned to a data sample vector **r**, the classification of **r** is performed by representing **r** as a p-dimensional vector, $\hat{\boldsymbol{\alpha}}(\mathbf{r}) = (\hat{\alpha}_1(\mathbf{r}), \hat{\alpha}_2(\mathbf{r}), \cdots, \hat{\alpha}_p(\mathbf{r}))^T$. When it is a pure-sample classification, only one $\hat{\alpha}_j(\mathbf{r}) = 1$ for some j with $1 \leq j \leq p$ and the rest are zeros in which case only one class can be assigned to **r**. When it is a mixed-sample classification, $\hat{\alpha}_j(\mathbf{r})$ is a real value, which indicates the likelihood of the jth signal $\mathbf{s}_j$ to be assigned to the **r** via its abundance fraction estimate. Therefore, the pure sample-based classification is basically a class-label membership assignment that makes a hard decision on which class is more likely to represent the **r**. Two such well-known classification techniques will be reviewed: Fisher's linear discriminant analysis (FLDA) in Section 2.3.1.1 and support vector machine (SVM) in Section 2.3.1.2. On the other hand, the mixed sample classification essentially performs abundance fraction estimation that assigns a likelihood of a particular class to the sample **r** and the classification is performed by a set of likelihood values specified by abundance fractions. Two such mixed sample classification techniques, orthogonal subspace projection developed by Harsanyi and Chang (1994) and target-constrained interference-minimized filter (TCIMF) developed by Ren and Chang (2000), will be reviewed in Sections 2.3.2.1 and 2.3.2.2, both of which can be considered as extensions of CEM discussed in Section 2.2.3.

2.3.1 Classification with Hard Decisions

In this section, we first describe two supervised pure sample-based classification techniques, FLDA and SVM, which require training data to perform hard-decision classification. From a view point of how to use training data, FLDA and SVM are completely different approaches. While FLDA can be considered as a statistics-based technique using training data on-average case, SVM can be regarded as a geometry-based technique using training data to find a hyperplane maximizing the distance between training samples on worst-case in the sense that training samples are near the separating hyperpalnes. In other words, an FLDA-generated classifier is derived by means and variances of training classes, whereas an SVM-generated classifier is derived by worst training data called support vectors. This is similar to what is used to evaluate computational complexity of an algorithm based on average case and worst case. As a result, in order for an FLDA-based classifier to perform effectively, the pool of training sample must be sufficiently large to produce reliable statistics of the data. By contrast, in order for an SVM-based classifier to perform well, it does not need a large set of training data. Instead, it requires the worst training data samples, that is, support vectors that are close to hyperplanes used to separate data samples. Therefore, both approaches have their own strengths and weaknesses depending upon provided training data.

2.3.1.1 Fisher's Linear Discriminant Analysis (FLDA)

In binary classification, we assume that there are n_j training sample vectors given by $\{\mathbf{r}_i\}_{i=1}^{n_t}$ for C_j, $j = 1, 2$, with n_j being the number of training sample vectors in the jth class C_j. Let $\boldsymbol{\mu}$ be the global mean of the entire training sample vectors, denoted by $\boldsymbol{\mu} = (1/n_1)\sum_{\mathbf{r}_i \in C_1} \mathbf{r}_i + (1/n_2)\sum_{\mathbf{r}_i \in C_2} \mathbf{r}_i$, and $\boldsymbol{\mu}_j$ be the mean of the training sample vectors in the jth class C_j, denoted by $\boldsymbol{\mu}_j = (1/n_j)\sum_{\mathbf{r}_i \in C_j} \mathbf{r}_i$. The main idea of the FLDA is to find a vector that projects all data sample vectors into a new data space called feature space so that the projected data sample vectors in this feature space can have maximum separation of two classes, C_1 and C_2. Let $\mathbf{w}$ denote the projection vector with the projection carried out by $y = \mathbf{w}^T\mathbf{x}$. The training sample vectors $\{\mathbf{r}_i\}_{i=1}^{n_t}$ projected in this new feature space are then given by $\{\hat{r}_i\}_{i=1}^{n_t}$ with $\hat{r}_i = \mathbf{w}^T\mathbf{r}_i$, and the mean and variance of the projected training vectors in C_j can be calculated as $\hat{\mu}_j = (1/n_j)\sum_{\hat{r}_i \in C_j} \hat{r}_i$ and $\hat{\sigma}_j^2 = (1/n_j)\sum_{\hat{r}_i \in C_j} (\hat{r}_i - \hat{\mu}_j)^2$, respectively. The optimal projection vector $\mathbf{w}^*$ that achieves the maximum separability of binary classification should be the one that separates the two projected means, $\hat{\mu}_1$ and $\hat{\mu}_2$, as farther as possible while making two variances $\hat{\sigma}_1^2$ and $\hat{\sigma}_2^2$ as small as possible. In order to accomplish these two goals simultaneously, the $\mathbf{w}^*$ should be the one that maximizes the following criterion:

$$J(\mathbf{w}) = \frac{(\hat{\mu}_2 - \hat{\mu}_1)^2}{\hat{\sigma}_1^2 + \hat{\sigma}_2^2} = \frac{(\mathbf{w}^T\boldsymbol{\mu}_1 - \mathbf{w}^T\boldsymbol{\mu}_2)^2}{\sum_{\mathbf{r}_i \in C_1} (\mathbf{w}^T\mathbf{r}_i - \mathbf{w}^T\boldsymbol{\mu}_1)^2 + \sum_{\mathbf{r}_i \in C_2} (\mathbf{w}^T\mathbf{r}_i - \mathbf{w}^T\boldsymbol{\mu}_2)^2} \tag{2.34}$$

Or alternatively, (2.34) can be reexpressed in terms of matrix forms called Fisher's ratio or Rayleigh's quotient, which is

$$J(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_\mathrm{B}\mathbf{w}}{\mathbf{w}^T\mathbf{S}_W\mathbf{w}} \tag{2.35}$$

where $\mathbf{S}_\mathrm{B}$ and $\mathbf{S}_\mathrm{W}$ are referred as between-class and within-class scatter matrices, respectively, and given by

$$\mathbf{S}_\mathrm{B} = (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \tag{2.36}$$

and

$$\mathbf{S}_\mathrm{W} = \sum_{\mathbf{r}_i \in C_1} (\mathbf{r}_i - \boldsymbol{\mu}_1)(\mathbf{r}_i - \boldsymbol{\mu}_1)^T + \sum_{\mathbf{r}_i \in C_2} (\mathbf{r}_i - \boldsymbol{\mu}_2)(\mathbf{r}_i - \boldsymbol{\mu}_2)^T \tag{2.37}$$

The solution for $\mathbf{w}$ to maximize (2.35) is given in Bischop (1995) by

$$\mathbf{w}^\mathrm{FLDA} \propto \mathbf{S}_\mathrm{W}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) \tag{2.38}$$

which specifies a projection vector $\mathbf{w}^*$ subject to a scale constant. Interestingly, letting two classes be specified by the target class and the background class with their respective means, denoted by target mean $\bar{\mathbf{t}}$ and background mean $\bar{\mathbf{b}}$, (2.38) becomes

$$\mathbf{w}^{\mathrm{FLDA}} \propto \mathbf{S}_{\mathrm{W}}^{-1}(\bar{\mathbf{t}} - \bar{\mathbf{b}}) \tag{2.39}$$

Comparing (2.39) to (2.17) both have the same form except that $\mathbf{K}^{-1}$ in (2.17) is replaced with $\mathbf{S}_{\mathrm{W}}^{-1}$ in (2.32). More details about this relationship can be found in Chang and Ji (2006) and Chapters 13–14 of this book.

A remark on FLDA for binary classification is noteworthy. It has been shown in Bischop (1995) that FLDA-based binary classification is identical to least squares-based approach. In addition, FLDA-based binary classification is also shown by Chang et al. (2006) to be identical to Otsu's image thresholding method. However, it is interesting to note that none of these two can be applied to FLDA-based multiclassification.

In order to extend FLDA for two-class classification to multiple-class classification, assume that there are p classes of interest, $C_1, C_2, \cdots, C_p$ and n_t training sample vectors given by $\{\mathbf{r}_i\}_{i=1}^{n_t}$ for p-class classification, with n_j being the number of training sample vectors in the jth class C_j. Let $\boldsymbol{\mu}$ be the global mean of the entire training sample vectors, denoted by $\boldsymbol{\mu} = (1/n_t)\sum_{i=1}^{n_t}\mathbf{r}_i$, and $\boldsymbol{\mu}_j$ be the mean of the training sample vectors in the jth class C_j, denoted by $\boldsymbol{\mu}_j = \left(1/n_j\right)\sum_{\mathbf{r}_i \in C_j}\mathbf{r}_i$. Now, we can extend (2.36) and (2.37) to define the within-class scatter matrix, $\mathbf{S}_{\mathrm{W}}$ and between-class scatter matrix $\mathbf{S}_{\mathrm{B}}$ for p classes as follows.

$$\mathbf{S}_{\mathrm{W}} = \sum_{j=1}^{p}\mathbf{S}_j \quad \text{where} \quad \mathbf{S}_j = \sum_{\mathbf{r}\in C_j}\left(\mathbf{r} - \boldsymbol{\mu}_j\right)\left(\mathbf{r} - \boldsymbol{\mu}_j\right)^T \tag{2.40}$$

$$\mathbf{S}_B = \sum_{j=1}^{p} n_j\left(\boldsymbol{\mu}_j - \boldsymbol{\mu}\right)\left(\boldsymbol{\mu}_j - \boldsymbol{\mu}\right)^T \tag{2.41}$$

Using (2.40) and (2.41), Fisher's ratio in (2.35) is then generalized and given in Bischop (1995) by

$$\mathrm{Trace}\left\{\left(\mathbf{W}^T\mathbf{S}_W\mathbf{W}\right)^{-1}\mathbf{W}^T\mathbf{S}_{\mathrm{B}}\mathbf{W}\right\} \text{ for any matrix } \mathbf{W} \text{ of size } L \times (p-1)\mathbf{W} \tag{2.42}$$

p-Class FLDA classification is to find a set of feature vectors specified by the matrix $\mathbf{W}^{\mathrm{FLDA}}$ that maximize Fisher's ratio specified by (2.42) where the number of feature vectors found by $\mathbf{W}^{\mathrm{FLDA}}$ is $p-1$, which is determined by the number of classes, p. These FLDA-obtained $p-1$ feature vectors specify boundaries among p classes. More details can be found in Duda and Hart (1973), Bischop (1995, 2006), and Section 9.2.3 in Chang (2003a).

Since the feature matrix $\mathbf{W}^{\mathrm{FLDA}}$ obtained from (2.42) is a matrix that specifies $p-1$ feature vectors, directly finding $\mathbf{W}^{\mathrm{FLDA}}$ by maximizing (2.42) may encounter singularity problems. Instead of solving $p-1$ feature vectors in $\mathbf{W}^{\mathrm{FLDA}}$ all together from (2.42), an alternative solution is to find these $p-1$ feature vectors one at a time by solving (2.38). In other words, solving a multiple-class FLDA problem can be accomplished by solving a sequence of two-class FLDA problems one after another successively. A specific algorithm in doing so is described as follows.

Algorithm for finding successive feature vectors for FLDA

1. Assume that there are p classes of interest and $\mathrm{TS}_1, \mathrm{TS}_2, \cdots, \mathrm{TS}_p$ are p training sample sets where TS_j is the jth training sample set for the jth class C_j and contains n_t training sample vectors.

2. For $j = 1$, calculate the mean of class TS_1 by $\boldsymbol{\mu}_1 = (1/n_1)\sum_{\mathbf{r}_i \in \mathrm{TS}_1} \mathbf{r}_i$ and $\boldsymbol{\mu}_2 = (1/\bar{n}_1)\sum_{\mathbf{r}_i \in \cup_{j=2}^{p} \mathrm{TS}_j} \mathbf{r}_i$ where $\bar{n}_1 = \sum_{i=2}^{p} n_i$.
3. Find the first feature vector $\mathbf{w}_1^*$ by solving (2.38) with $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ obtained in Step 2. This is the same as finding an eigenvector corresponding to the largest eigenvalue of the matrix $\mathbf{S}_{\mathrm{W}}^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$.
4. Using the found $\mathbf{w}_1^*$, produce the OSP specified by $P_{\mathbf{w}_1^*}^{\perp} = \mathbf{I} - \mathbf{w}_1^*\left(\left(\mathbf{w}_1^*\right)^T \mathbf{w}_1^*\right)^{-1}\left(\mathbf{w}_1^*\right)^T$ developed in Harsanyi and Chang (1994) to $\mathrm{TS}_2, \mathrm{TS}_3, \cdots, \mathrm{TS}_p$ to produce $\mathrm{TS}_j^1 = P_{\mathbf{w}_1^*}^{\perp}\mathrm{TS}_j$ for $j = 2, 3, \cdots, p$.
5. Calculate $\boldsymbol{\mu}_2^1 = (1/n_2)\sum_{\mathbf{r}_i \in \mathrm{TS}_2^1} \mathbf{r}_i$, $\boldsymbol{\mu}_3^1 = \left(1/\bar{n}_1^1\right)\sum_{\mathbf{r}_i \in \cup_{j=3}^{p} \mathrm{TS}_j^1} \mathbf{r}_i$, and $\bar{n}_1^1 = \sum_{i=3}^{p} n_i$.
6. Find the second feature vector $\mathbf{w}_2^*$ by solving (2.38) with $\boldsymbol{\mu}_1$ replaced by $\boldsymbol{\mu}_2^1$ and $\boldsymbol{\mu}_2$ replaced by $\boldsymbol{\mu}_3^1$, that is, an eigenvector corresponding to the largest eigenvalue of the matrix $\mathbf{S}_{\mathrm{W}}^{-1}\left(\boldsymbol{\mu}_3^1 - \boldsymbol{\mu}_2^1\right)$.
7. Apply $P_{\mathbf{w}_2^*}^{\perp} = \mathbf{I} - \mathbf{w}_2^*\left(\left(\mathbf{w}_2^*\right)^T \mathbf{w}_2^*\right)^{-1}\left(\mathbf{w}_2^*\right)^T$ to $\mathrm{TS}_3, \mathrm{TS}_4, \cdots, \mathrm{TS}_p$ to produce $\mathrm{TS}_j^2 = P_{\mathbf{w}_2^*}^{\perp}\mathrm{TS}_j^1$ or $\mathrm{TS}_j^2 = P_{\mathbf{W}^{*2}}^{\perp}\mathrm{TS}_j$ with $\mathbf{W}^{*2} = \left[\mathbf{w}_1^*\mathbf{w}_2^*\right]$ for $j = 3, \cdots, p$.
8. Repeat the procedure of steps 2–7 over and over again using $P_{\mathbf{w}_j^*}^{\perp} = \mathbf{I} - \mathbf{w}_j^*\left(\left(\mathbf{w}_j^*\right)^T \mathbf{w}_j^*\right)^{-1}\left(\mathbf{w}_j^*\right)^T$ or $P_{\mathbf{W}^{*j}}^{\perp} = \mathbf{I} - \mathbf{W}^{*j}\left(\left(\mathbf{W}^{*j}\right)^T \mathbf{W}^{*j}\right)^{-1}\left(\mathbf{W}^{*j}\right)^T$ for $j = 3, \cdots, p$ until $p - 1$ feature vectors are generated, $\mathbf{w}_3^*, \cdots, \mathbf{w}_{p-1}^*$.

It is worth noting that the key for the above algorithm to work is to assume that all the $p - 1$ feature factors are orthogonal to each other. This is true because solving the feature vector via (2.38) is equivalent to finding an eigenvector associated with the largest eigenvalue of $\mathbf{S}_{\mathrm{w}}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$. As a result, two different feature vectors result in two distinct eigenvectors that are always orthogonal. The OSP is used to guarantee that its feature spaces from which feature vectors are generated are indeed orthogonal to each other.

2.3.1.2 Support Vector Machines (SVM)

In this section, we describe another interesting pure-sample classifier, support vector machine, which uses a different classification criterion. Instead of maximizing the distance between two class means and minimizing sum of the distances within each of two classes as measured by Fisher's ratio in (2.35), SVM uses a set of training sample vectors, referred to as support vectors for each of two classes and maximizes the distance between the support vectors in these two classes. Following the same approach presented in Chapter 6 of Haykin (1999), its idea can be described mathematically as follows.

Assume that a linear discriminant function, $y = g(\mathbf{r})$ is specified by

$$y = g(\mathbf{r}) = \mathbf{w}^T\mathbf{r} + b \tag{2.43}$$

where $\mathbf{w}$ is a weighting vector that determines the performance of $y = g(\mathbf{r})$ based on the sample vector $\mathbf{r}$ and b is a bias.

Consider a two-class classification problem with a given set of training data $\{(\mathbf{r}_i, y_i)\}_{i=1}^{n}$ where $\{\mathbf{r}_i\}_{i=1}^{n}$ are n training samples with their associated binary decisions $\{d_i\}_{i=1}^{n}$ specified by either 1 or 0. For each pair of $\{(\mathbf{r}_i, d_i)\}_{i=1}^{n}$ it satisfies

$$d_i = \begin{cases} 1; \text{ if } y_i = g(\mathbf{r}_i) = \mathbf{w}^T\mathbf{r}_i + b \geq 0 \\ 0; \text{ if } y_i = g(\mathbf{r}_i) = \mathbf{w}^T\mathbf{r}_i + b < 0 \end{cases} \tag{2.44}$$

where

$$y = g(\mathbf{r}) = \mathbf{w}^T\mathbf{r} + b = 0 \tag{2.45}$$

specifies a hyperplane that is a decision boundary that separates two classes. If $\mathbf{n}$ be the normal vector of the hyperplane specified by (2.45), then (2.45) can be reexpressed as

$$\mathbf{n}^T\mathbf{r} + b = 0 \tag{2.46}$$

which indicates that the $\mathbf{n}$ is orthogonal to the hyperplane. If we further let $\mathbf{u}$ be the normalized unit vector of $\mathbf{n}$, that is, $\mathbf{u} = \mathbf{n}/||\mathbf{n}||$, then any data sample vector $\mathbf{r}$ in the data can be represented by

$$\mathbf{r} = \mathbf{r}_p + r\mathbf{u} \tag{2.47}$$

where $\mathbf{r}_p$ is the projected point of $\mathbf{r}$ onto the hyperplane with $g(\mathbf{r}_p) = \mathbf{w}^T\mathbf{r}_p + b = 0$ and the scalar ρ is the distance of the data sample vector $\mathbf{x}$ from its projected vector $\mathbf{x}_p$ which can be calculated as follows.

$$\begin{aligned} g(\mathbf{r}) &= g(\mathbf{r}_p + \rho\mathbf{u}) = \mathbf{w}^T\left(\mathbf{r}_p + \rho\mathbf{u}\right) + b \\ &= \mathbf{w}^T\mathbf{r}_p + b + \rho\mathbf{w}^T\mathbf{u} = g(\mathbf{r}_p) + \rho\mathbf{w}^T\mathbf{u} = \rho\mathbf{w}^T\mathbf{u} \\ \Rightarrow \rho &= g(\mathbf{r})/\mathbf{w}^T\mathbf{u} \end{aligned} \tag{2.48}$$

If $\mathbf{w} = \mathbf{n}$, then $\rho = g(\mathbf{r})/\mathbf{n}^T\mathbf{u} = g(\mathbf{r})/||\mathbf{n}||$, which implies that $g(\mathbf{r}) = \rho||\mathbf{n}||$. In particular, if $\mathbf{r} = \mathbf{0}$ and $\mathbf{w} = \mathbf{n}$, $\rho = b/||\mathbf{n}||$. Figure 2.3 depicts their graphical relationship.

An SVM is a linear discriminant function that can be used as a linear binary classifier. It was originally developed in statistical machine learning theory by Vapnik (1998). For a given set of training data, $\{(\mathbf{r}_i, d_i)\}_{i=1}^n$, an SVM finds a weighting vector $\mathbf{w}$ and bias b that satisfy

$$d_i = \begin{cases} 1; \text{ if } y_i = g(\mathbf{r}_i) = \mathbf{w}^T\mathbf{r}_i + b \geq 0 \\ -1; \text{ if } y_i = g(\mathbf{r}_i) = \mathbf{w}^T\mathbf{r}_i + b < 0 \end{cases} \tag{2.49}$$

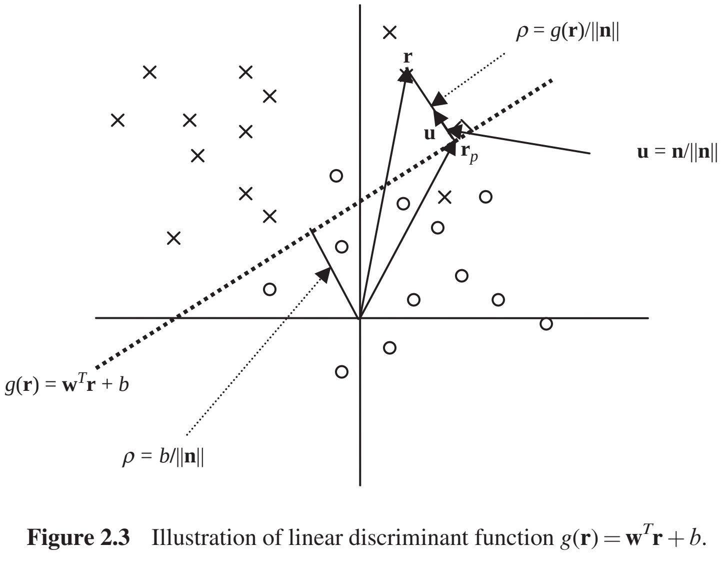

$\rho = g(\mathbf{r})/||\mathbf{n}||$
$\mathbf{r}$
$\mathbf{u}$
$\mathbf{r}_p$
$\mathbf{u} = \mathbf{n}/||\mathbf{n}||$
$g(\mathbf{r}) = \mathbf{w}^T\mathbf{r} + b$
$\rho = b/||\mathbf{n}||$

Figure 2.3 Illustration of linear discriminant function $g(\mathbf{r}) = \mathbf{w}^T\mathbf{r} + b$.

and maximize the margin of separation defined by distance between a hyperplane and closest data samples. In particular, (2.44) can be rederived by incorporating its binary decision into the discriminant function (2.49) as follows

$$d_i\left(\mathbf{w}^T\mathbf{r}_i + b\right) \geq 1 \quad \text{for} \quad 1 \leq i \leq n \tag{2.50}$$

It should be noted that (2.49) is modified from (2.44) by replacing decision 0 with decision 1 when $y_i = g(\mathbf{r}_i) = \mathbf{w}^T\mathbf{r}_i + b < 0$ so that two decisions in (2.49) can be combined in to one equation (2.50).

According to (2.48) the distance ρ between a sample vector $\mathbf{r}$ and its projected vector on the hyperplane $g(\mathbf{r}) = \mathbf{w}^T\mathbf{r} + b = 0$ is specified by $\rho = g(\mathbf{r})/||\mathbf{w}||$, with $\mathbf{w}$ being the normal vector of the hyperplane. Since $g(\mathbf{x})$ takes only $+1$ or -1, the distance ρ is then defined by

$$\rho = \begin{cases} 1/||\mathbf{w}|| \text{ if } y_i = +1 \\ -1/||\mathbf{w}|| \text{ if } y_i = -1 \end{cases} \tag{2.51}$$

Using (2.51) we define the margin of separation between two classes as

$$\rho = 2/||\mathbf{w}|| \tag{2.52}$$

By virtue of (2.50)–(2.52), the linear separability problem for the SVM is to find an optimal weight vector $\mathbf{w}^*$ that maximizes the distance specified by (2.52), which is equivalent to minimizing

$$\Phi(\mathbf{w}) = (1/2)\mathbf{w}^T\mathbf{w} = (1/2)||\mathbf{w}||^2 \tag{2.53}$$

subject to constraints specified by (2.50). Figure 2.4 illustrates the concept of the SVM where two classes of data sample vectors are denoted by “open circles” and “crosses” and the vectors satisfying the equality of (2.50) are called support vectors.

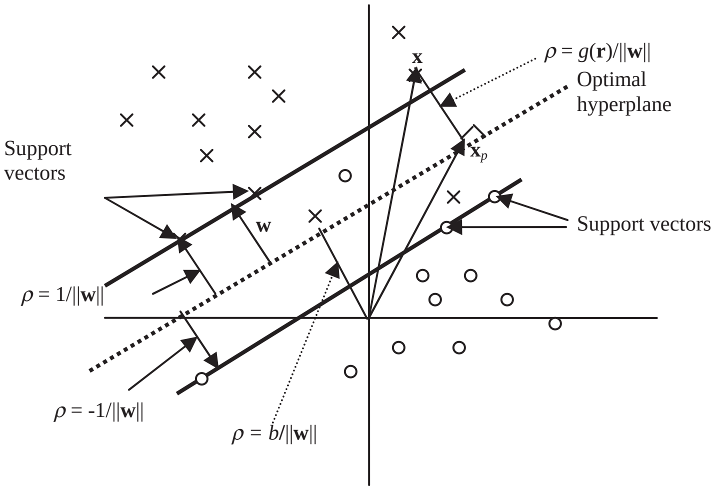

$\rho = g(\mathbf{r})/||\mathbf{w}||$
Optimal hyperplane
x
$\mathbf{x}_p$
Support vectors
Support vectors
w
$\rho = 1/||\mathbf{w}||$
$\rho = -1/||\mathbf{w}||$
$\rho = b/||\mathbf{w}||$

Figure 2.4 Illustration of SVM.

In order to solve the constrained optimization problem specified by (2.53), we introduce a Lagrangian defined by

$$J(\mathbf{w}, b, \mathbf{a}) = (1/2)\mathbf{w}^T\mathbf{w} - \sum\nolimits_{i=1}^{n} a_i\left[d_i\left(\mathbf{w}^T\mathbf{r}_i + b\right) - 1\right] \tag{2.54}$$

which takes care of the cost function given by (2.54) along with constraints specified by (2.46). Differentiating $J(\mathbf{w}, b, \mathbf{a})$ in (2.54) with respect to $\mathbf{w}$ and b yields

$$\frac{\partial J(\mathbf{w}, b, \mathbf{a})}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w}^* = \sum\nolimits_{i=1}^{n} a_i d_i \mathbf{r}_i \tag{2.55}$$

$$\frac{\partial J(\mathbf{w}, b, \mathbf{a})}{\partial b} = 0 \Rightarrow \sum\nolimits_{i=1}^{n} a_i d_i = 0 \tag{2.56}$$

along with the constraint (2.50)

$$a_i\left[d_i\left(\mathbf{w}^T\mathbf{r}_i + b\right) - 1\right] = 0 \text{ for } 1 \leq i \leq n \tag{2.57}$$

Expanding $J(\mathbf{w}, b, \mathbf{a})$ and $\mathbf{w}^T\mathbf{w}$ we obtain

$$J(\mathbf{w}, b, \mathbf{a}) = (1/2)\mathbf{w}^T\mathbf{w} - \sum\nolimits_{i=1}^{n} a_i d_i \mathbf{w}^T\mathbf{r}_i - b\sum\nolimits_{i=1}^{n} a_i d_i + \sum\nolimits_{i=1}^{n} a_i \tag{2.58}$$

$$\mathbf{w}^T\mathbf{w} = \sum\nolimits_{i=1}^{n} a_i d_i \mathbf{w}^T\mathbf{r}_i = \sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{n} a_i a_j d_i d_j \mathbf{r}_i^T\mathbf{r}_j \tag{2.59}$$

Using (2.55) and (2.56) we can reformulate $J(\mathbf{w}, b, \mathbf{a})$ as

$$Q(\mathbf{a}) = \sum\nolimits_{i=1}^{n} a_i - (1/2)\sum\nolimits_{i=1}^{n}\sum\nolimits_{j=1}^{n} a_i a_j d_i d_j \mathbf{r}_i^T\mathbf{r}_j \tag{2.60}$$

In light of (2.60) an alternative form of the linear separability problem for the SVM can be formulated as follows.

Alternative Form of Linear Separability Problem for SVM

Given a set of training pool, $\{(\mathbf{r}_i, d_i)\}_{i=1}^{n}$, find a set of Lagrange multiplier vector $\mathbf{a} = (a_1, a_2, \cdots, a_n)^T$ that maximizes $Q(\mathbf{a})$ is given by (2.60) subject to the following constraints

1. $\sum_{i=1}^{n} a_i d_i = 0$
2. $a_i \geq 0$ for $1 \leq i \leq n$

Solution to the above optimization problem is given by

$$\mathbf{w}^{\mathrm{SVM}} = \sum\nolimits_{i=1}^{n} a_i^{\mathrm{SVM}} d_i \mathbf{r}_i \tag{2.61}$$

$$1 = d_s = \left(\mathbf{w}^{\mathrm{SVM}}\right)^T\mathbf{r}^s + b \Rightarrow b = 1 - \left(\mathbf{w}^{\mathrm{SVM}}\right)^T\mathbf{r}^s \tag{2.62}$$

with $\mathbf{r}^s$ is a support vector on the hyperplane with its decision $d_s = +1$.

By far the SVM discussed above was developed to separate two classes that are linearly separable. That is, the data sample vectors in two classes can be separated by a distance greater than ρ from the hyperplane shown in Figure 2.4. However, in many applications, such desired situations may not occur. In other words, some data sample vectors fall in the region within the distance less than ρ from the hyperplane or even on the wrong side of the hyperplane. These data sample vectors can be considered to be either bad or confusing data sample vectors and they cannot be linearly separated. In this case, the SVM developed for linear separable problems outlined by (2.58)–(2.60) must be rederived to take care of such confusing data sample vectors. In order for linear SVMs to solve linear nonseparable problems, two approaches are of particular interest. One is a kernel-based approach that will be discussed in detail in Section 33.5. The other is to introduce a set of slack variables into SVMs to resolve the dilemma caused by confusing data sample vectors. Specifically, a new set of positive parameters, denoted by $\{\xi_i\}_{i=1}^n$ and referred to as slack variables, must be introduced to measure the deviation of a data sample vector from the ideal condition of linear separability, in which case $\xi_i < 0$. However, if $0 \leq \xi_i \leq 1$, the ith data sample vector $\mathbf{x}_i$ falls within the region with distance less than the margin of separation but on correct side of the decision surface specified by the hyperplane. On the other hand, if $\xi_i > 1$, the ith data sample vector $\mathbf{x}_i$ falls on the wrong side of its decision surface. In light of the mathematical interpretation, these issues can be addressed by the following inequalities:

$$d_i\left(\mathbf{w}^T\mathbf{r}_i + b\right) \geq 1 - \xi_i \quad \text{for} \quad 1 \leq i \leq n \tag{2.63}$$

$$\xi_i \geq 0 \quad \text{for} \quad 1 \leq i \leq n \tag{2.64}$$

By incorporating (2.63) and (2.64) in (2.53) the object function $\Phi(\mathbf{w})$ can be modified as

$$\Phi(\mathbf{w}) = (1/2)\mathbf{w}^T\mathbf{w} + C\sum\nolimits_{i=1}^n \xi_i \quad \text{with} \quad C > 0 \tag{2.65}$$

By virtue of (2.63)–(2.65) a linear nonseparable problem solved by the SVM can be described as follows.

Given a set of training pools, $\{(\mathbf{r}_i, d_i)\}_{i=1}^n$, find an optimal weight vector $\mathbf{w}^{\mathrm{SVM}}$ and bias b^{SVM} such that they satisfy the constraints specified by (2.63) and (2.64) and such that the pair of ($\mathbf{w}^{\mathrm{SVM}}$, b^{SVM}) and $\{\xi_i\}_{i=1}^n$ minimizes (2.65). Or an alternative form can be obtained as follows.

Given a set of training pool, $\{(\mathbf{x}_i, d_i)\}_{i=1}^n$, find a set of Lagrange multiplier vectors $\mathbf{a} = (a_1, a_2, \cdots, a_n)^T$ that maximize $Q(\mathbf{a})$ given by (2.60) subject to the following constraints:

1. $\sum_{i=1}^n a_i d_i = 0$
2. $C \geq a_i \geq 0$ for $1 \leq i \leq n$, with C being a predetermined positive parameter.

The optimal solution $\mathbf{w}^{\mathrm{SVM}}$ to the above optimization problem is given by

$$\mathbf{w}^{\mathrm{SVM}} = \sum\nolimits_{i=1}^{n_s} a_i^{\mathrm{SVM}} d_i^s \mathbf{r}_i^s \tag{2.66}$$

where n_s is the number of support vectors.

It should be noted that the slack variables $\{\xi_i\}_{i=1}^n$ along with their Lagrange multipliers $\{\eta_i\}_{i=1}^n$ that are used to constrain non-negativity of $\{\xi_i\}_{i=1}^n$ do not appear in the alternative form. Instead, the constraint constant C associated with the slack variables is included to bound the Lagrange multiplier $\{a_i\}_{i=1}^n$ from above. Nevertheless, by means of (2.60) we can still derive the following equations that include the slack variables $\{\xi_i\}_{i=1}^n$ and their associated Lagrange multipliers $\{\eta_i\}_{i=1}^n$.

$$a_i\left[d_i\left(\mathbf{w}^T\mathbf{r}_i + b\right) - (1-\xi_i)\right] = 0 \quad \text{for} \quad 1 \le i \le n \tag{2.67}$$

$$\xi_i\eta_i = 0 \quad \text{for} \quad 1 \le i \le n \tag{2.68}$$

Differentiating with the slack variables, $\{\xi_i\}_{i=1}^n$ and setting to zero yields

$$a_i + \eta_i = C \tag{2.69}$$

Combining (2.68) and (2.69) results in

$$\xi_i = 0 \quad \text{if} \quad a_i < C \tag{2.70}$$

In this case, the b^{SVM} can be obtained by taking any data sample vector $\mathbf{x}_i$ with $0 < a_i^{\text{SVM}} < C$ (in which case, $\xi_i = 0$ due to (2.70)) and (2.67).

As originally designed, SVM is a binary classifier. In order to extend SVM to a multiclass classifier, two approaches have been proposed in the past. One is the so-called one-against-all also known as winner-take-all approach, which reduces a multiclass classification problem to a set of two-class binary classification problems, each of which is produced by SVM to separate one particular training data sample vector from the rest of training sample vectors. That is, let Ω denote the set of classes $\Omega = \{\omega_k\}_{k=1}^{n_C}$ and n_C be the number of classes. For each training sample vector $\mathbf{r}$, we consider two classes, $\Omega_k = \{\omega_k\}$ and $\Omega - \Omega_k$. The discriminant function or decision function of a sample vector $\mathbf{x}$ derived by the SVM that separates the class ω_k from $\Omega - \Omega_k$ is given by

$$y_k(\mathbf{r}) = \left(\mathbf{w}_k^{\text{SVM}}\right)^T \boldsymbol{\phi}(\mathbf{r}) + b_k = \sum\nolimits_{i=1}^n a_i d_i K(\mathbf{r}_i, \mathbf{r}) \tag{2.71}$$

The data sample vector $\mathbf{r}$ is then classified into the class denoted by class($\mathbf{r}$) that yields the maximum value of $y_k(\mathbf{r})$ over all $1 \le k \le n_C$, which is

$$\text{class}(\mathbf{r}) = \arg\{\max\nolimits_{1\le k\le n_C} y_k(\mathbf{r})\} \tag{2.72}$$

A second approach is to reduce a multiclass classification problem to a set of binary classification problems by pairing any two classes as a binary classification problem. As a result, there are $n_C(n_C-1)/2$ pairwise binary classification problems. In other words, each binary classification is specified by two separate classes ω_k and ω_l. The SVM separating these two classes is specified by its discriminant function or decision function $y_{kl}(\mathbf{r}_i)$ for a sample vector $\mathbf{r}$. If we define a score function of a sample vector $\mathbf{r}$ associated with class ω_k by

$$S_k(\mathbf{r}) = \sum\nolimits_{l=1, l\ne k}^{n_C} \text{sgn}(y_{kl}(\mathbf{r})) \tag{2.73}$$

The class assigned to $\mathbf{r}$ is one that yields the maximum score given by

$$\text{class}(\mathbf{r}) = \arg\{\max_{1 \le k \le n_C} S_k(\mathbf{r})\} \tag{2.74}$$

The issues arising from using SVM to perform multi-classification are similar to those encountered in FLDA discussed in Section 2.3.1.1 and are addressed in detail in Bischop (2006).

2.3.2 Classification with Soft Decisions

Recall that the model in (2.13) is used for subsample target detection where the desired target $\mathbf{t}$ is assumed to partially occupy the sample vector $\mathbf{r}$ as a subsample target with its abundance fraction specified by α and the background $\mathbf{b}$ is accounted for the remaining abundance $(1-\alpha)$. In this case, the algorithms developed for subsample target detection are designed to enhance the detectability of the target of interest while suppressing the background as much as possible. The CEM developed in Section 2.2.3 was the one that adopts this design rationale and philosophy. However, this is not true for mixed sample analysis where the background knowledge is completely specified by other known target signals which are also of interest. In order for CEM to be able to perform mixed sample analysis, several CEM-based mixed sample classification techniques were investigated in Chang (2002b) and presented in Chang (2003a), among which two techniques are of particular interest, orthogonal subspace projection derived from linear mixture analysis and target-constrained interference-minimized filter, which will be discussed in this and following sections.

2.3.2.1 Orthogonal Subspace Projection (OSP)

The OSP was originally developed by Harsanyi and Chang (1994) for linear spectral mixture analysis. Its idea is similar to that used in the ASD discussed in Section 2.2.2.2. It makes an assumption that a data sample vector $\mathbf{r}$ is completely characterized by a linear mixture of a finite number of so-called endmembers that are assumed to be present in the image data. More specifically, suppose that L is the number of spectral bands and $\mathbf{r}$ is an L-dimensional data sample vector. Let $\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_p$ be p endmembers, which are generally referred to as digital numbers (DNs). The data sample $\mathbf{r}$ can be then linearly represented by $\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_p$ with appropriate abundance fractions specified by α_1, $\alpha_2, \cdots$, α_p. In other words, let $\mathbf{r}$ be an $L \times 1$ column vector, $\mathbf{M}$ be an endmember signature matrix, denoted by $[\mathbf{m}_1\ \mathbf{m}_2\ \cdots\ \mathbf{m}_p]$, and $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_p)^T$ be a $p \times 1$ abundance column vector associated with $\mathbf{r}$ where α_j denotes the abundance fraction of the jth endmember $\mathbf{m}_j$ present in the $\mathbf{r}$. Then an L-dimensional data sample $\mathbf{r}$ can be represented by a linear mixture model as follows.

$$\mathbf{r} = [\mathbf{m}_1 \mathbf{m}_2 \cdots \mathbf{m}_p] \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_p \end{bmatrix} + \mathbf{n} = \sum\nolimits_{j=1}^{p} \alpha_j \mathbf{m}_j + \mathbf{n} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n} \tag{2.75}$$

where $\mathbf{n}$ is noise or can be interpreted as either a measurement or a model error.

If we assume that the target signal of interest is $\mathbf{m}_p$ without loss of generality and compare (2.75) with (2.13), an immediate finding is that the subsample target of interest $\mathbf{t}$ and the background $\mathbf{b}$ in (2.13) are now represented by $\mathbf{m}_p$, with the $\mathbf{b}$ completely characterized by the remaining $p-1$ target signals, $\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_{p\,-\,1}$ as a background matrix $\mathbf{B} = [\mathbf{m}_1 \mathbf{m}_2 \cdots \mathbf{m}_{p-1}]$. With this interpretation the model (2.75) can be reexpressed as a subsample target detection model similar to (2.13) and given by

$$\begin{aligned}\mathbf{r} &= \left[\mathbf{m}_1\mathbf{m}_2\cdots\mathbf{m}_p\right]\begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_p\end{bmatrix}+n=\alpha_p\mathbf{m}_p+\textstyle\sum_{j=1}^{p-1}\alpha_j\mathbf{m}_j+n\\ &=\alpha_p\mathbf{m}_p+\mathbf{B}\begin{bmatrix}\alpha_1\\ \alpha_2\\ \vdots\\ \alpha_{p-2}\end{bmatrix}+\mathbf{n}=\alpha_p\mathbf{m}_p+\mathbf{B}\gamma+\mathbf{n}\end{aligned} \tag{2.76}$$

where $\mathbf{m}_p = \mathbf{t}$ with $\alpha_p = \alpha$ and $\mathbf{b} = \mathbf{B}$, with $(1-\alpha)$ replaced by the $p-1$ dimensional abundance vector $\boldsymbol{\gamma} = \left[\alpha_1, \alpha_2, \cdots, \alpha_{p-1}\right]^T$.

It has been shown in Harsanyi and Chang (1994), Chang (2003a), and Chang (2005) that the optimal solution of α_p to (2.76), $\hat{\alpha}_p(\mathbf{r})$ is given by an OSP projector $\delta^{\mathrm{OSP}}(\mathbf{r})$

$$\hat{\alpha}_p(\mathbf{r}) = \delta^{\mathrm{OSP}}(\mathbf{r}) = \kappa\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} \text{ subject to a constant } \kappa \tag{2.77}$$

where the desired signal $\mathbf{d}$ is designated by the signal of interest $\mathbf{m}_p$, $\mathbf{U} = \left[\mathbf{m}_1\mathbf{m}_2\cdots\mathbf{m}_{p-1}\right]$ is undesired signal matrix, and $P_{\mathbf{U}}^{\perp}$ is an undesired signal subspace projector given by

$$P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\mathbf{U}^{\#} \tag{2.78}$$

with $\mathbf{U}^{\#} = \left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T$ being the pseudoinverse of $\mathbf{U}$. The notation $\frac{\perp}{\mathbf{U}}$ in $P_{\mathbf{U}}^{\perp}$ indicates that the projector $P_{\mathbf{U}}^{\perp}$ maps the data sample $\mathbf{r}$ into the orthogonal complement of $\langle\mathbf{U}\rangle$, denoted by $\langle\mathbf{U}\rangle^{\perp}$. If we compare (2.77) to (2.27), we immediately find that both ASD and OSP are derived from maximizing SCR or SNR and also yield the same detector structure except that the subtarget signal $\mathbf{t}$ and the inverse of the sample covariance matrix, $\mathbf{K}^{-1}$ in (2.27) and (2.28) are replaced by the desired signal $\mathbf{d}$ to be classified and the undesired signal subspace projector $P_{\mathbf{U}}^{\perp}$ in (2.85), respectively. This is because OSP has prior knowledge about the undesired signals, $\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_{p-1}$ and takes advantage of $P_{\mathbf{U}}^{\perp}$ to annihilate such unwanted interference prior to the use of matched filter. By contrast, ASD does not have such prior knowledge about $\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_{p-1}$. As a consequence, it makes use of a whiten matrix $\mathbf{K}^{-1}$ resulting from the sample covariance matrix $\mathbf{K}$ to suppress unknown interference. Detailed discussion on this issue can be found in Chapter 12 and Chang (2005).

By virtue of (2.77) $\delta^{\mathrm{OSP}}(\mathbf{r})$ performs classification with soft decisions by estimating the abundance fractions of each of p endmembers $\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_p$ in such a way that each of p endmembers $\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_p$ is designated as a subsample target of interest specified by the desired signal $\mathbf{d}$, while other signals are considered as undesired signals to form an undesired signal matrix $\mathbf{U}$.

It was shown in Chang (1998) that $\delta^{\mathrm{OSP}}(\mathbf{r})$ in (2.77) performed as a detector rather than an estimator since it did not consider the estimation accuracy of abundance α_p. In order to make $\delta^{\mathrm{OSP}}(\mathbf{r})$ perform an estimator, a scaling constant $\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$ is included in $\delta^{\mathrm{OSP}}(\mathbf{r})$ to correct the estimation error resulting from $\delta^{\mathrm{OSP}}(\mathbf{r})$. The resultant estimator turns out to be a least squares-based estimator given by

$$\delta^{\mathrm{LSOSP}}(\mathbf{r}) = \frac{\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r}}{\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}} \tag{2.79}$$

which has a form similar to that of CEM in (2.33) except that the subtarget signal **t** and the inverse of the sample covariance matrix, $\mathbf{R}^{-1}$ in (2.33) are replaced by the desired signal **d** to be classified and the undesired signal subspace projector, $P_{\mathbf{U}}^{\perp}$ in (2.77). Furthermore, comparing (2.79) to (2.28), both ASD in (2.28) and LSOSP in (2.77) judiciously select an appropriate constant κ, $\kappa = \left(\mathbf{t}^T\mathbf{K}^{-1}\mathbf{t}\right)^{-1}$in (2.27) and $\kappa = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$ in (2.77) to account for the error caused by estimating the abundance of target signal **t** in (2.13) or desired signal **d** in (2.75).

2.3.2.2 Target-Constrained Interference-Minimized Filter (TCIMF)

As shown in Section 2.2.2 CEM was designed to detect a single subsample target. If there are multiple subsample targets in the sample vector **r**, one subsample target must be carried out by CEM at a time. On the other hand, the OSP derived in (2.77) and (2.79) is developed for mixed sample classification for each of endmembers, $\mathbf{m}_1, \mathbf{m}_2, \cdots, \mathbf{m}_p$. However, in reality, background may not be completely characterized by known signals. In most cases, interferers that cannot be identified *a priori* may be also present in the sample vector **r**. The OSP is not designed to account for such unknown interferers. In order to extend the capability of CEM to perform detection of multiple targets as well as take advantage of the capability of OSP in suppressing unknown interference a TCIMF was recently developed by Ren and Chang (2000), which can be viewed as a generalization of the OSP and CEM.

Let $\mathbf{D} = [\mathbf{d}_1\mathbf{d}_2\cdots\mathbf{d}_{n_{\mathbf{D}}}]$ denote the desired target signature matrix and $\mathbf{U} = [\mathbf{u}_1\mathbf{u}_2\cdots\mathbf{u}_{n_{\mathbf{U}}}]$ be the undesired target signature matrix where $n_{\mathbf{D}}$ and $n_{\mathbf{U}}$ are the number of the desired target signals and the number of the undesired target signals, respectively. Now, we can develop an FIR filter that passes the desired target signals in **D** using an $n_{\mathbf{D}} \times 1$ unity constraint vector $\mathbf{1}_{n_{\mathbf{D}}\times 1} = \underbrace{(1, 1, \cdots, 1)^T}_{n_{\mathbf{D}}}$ while annihilating the undesired target signals in **U** using an $n_{\mathbf{U}} \times 1$ zero constraint vector $\mathbf{0}_{n_{\mathbf{U}}\times 1} = \underbrace{(0, 0, \cdots, 0)^T}_{n_{\mathbf{U}}}$. In doing so, the constraint in (2.31) is replaced by

$$[\mathbf{D}\,\mathbf{U}]^T\mathbf{w} = \begin{bmatrix} \mathbf{1}_{n_{\mathbf{D}}\times 1} \\ \mathbf{0}_{n_{\mathbf{U}}\times 1} \end{bmatrix} \tag{2.80}$$

and the optimization problem specified by (2.31) becomes the following linearly constrained optimization problem

$$\min_{\mathbf{w}}\left\{\mathbf{w}^T\mathbf{R}\mathbf{w}\right\} \text{ subject to } [\mathbf{D}\,\mathbf{U}]^T\mathbf{w} = \begin{bmatrix} \mathbf{1}_{n_{\mathbf{D}}\times 1} \\ \mathbf{0}_{n_{\mathbf{U}}\times 1} \end{bmatrix} \tag{2.81}$$

The filter solving (2.81) is called target-constrained interference-minimized filter (TCIMF) (Ren and Chang, 2000) and given by

$$\delta^{\mathrm{TCIMF}}(\mathbf{r}) = \left(\mathbf{w}^{\mathrm{TCIMF}}\right)^T\mathbf{r} \tag{2.82}$$

with the optimal weight vector, $\mathbf{w}^{\text{TCIMF}}$ given by

$$\mathbf{w}^{\text{TCIMF}} = \mathbf{R}^{-1}[\mathbf{D}\,\mathbf{U}]\left([\mathbf{D}\,\mathbf{U}]^{T}\mathbf{R}^{-1}[\mathbf{D}\,\mathbf{U}]\right)^{-1}\begin{bmatrix}\mathbf{1}_{n_{\mathbf{D}}\times 1}\\ \mathbf{0}_{n_{\mathbf{U}}\times 1}\end{bmatrix} \tag{2.83}$$

A discussion on the relationship between OSP and CEM via TCIMF can be found in Chang (2002, 2003a, 2005). For example, if $\mathbf{D} = \mathbf{m}_p = \mathbf{d}$ with $n_{\mathbf{D}} = 1$ and $\mathbf{U} = \left[\mathbf{m}_1\mathbf{m}_2\cdots\mathbf{m}_{p-1}\right]$ with $n_{\mathbf{U}} = p - 1$, $\delta^{\text{TCIMF}}(\mathbf{r})$ performs like $\delta^{\text{OSP}}(\mathbf{r})$. On the other hand, if $\mathbf{D} = \mathbf{m}_p = \mathbf{d}$ with $n_{\mathbf{D}} = 1$ and $\mathbf{U} = \varnothing$ with $n_{\mathbf{U}} = 0$, $\delta^{\text{TCIMF}}(\mathbf{r})$ performs like $\delta^{\text{CEM}}(\mathbf{r})$. More details on OSP in comparison with CEM and TCIMF can be found in Chapter 12.

2.4 Kernel-Based Classification

Kernel-based approaches have recently received considerable interest in solving linear nonseparable problems in classification. Their main idea is to introduce a nonlinear kernel that maps the original data into a high-dimensional feature space from which the extracted features can be used to resolve the issue of linear nonseparability encountered in the original data space. The property of the Mercer's theorem in Scholkopf and Smola (2002) then plays a trick called kernel trick to implicitly calculate the dot products in the feature space **F** without actually mapping all data sample vectors into the feature space **F** in which case the kernel function was not even identified.

2.4.1 Kernel Trick Used in Kernel-Based Methods

Prior to introducing kernel-based classifiers a kernelization procedure and kernel trick need to be discussed. The idea of kernel-based techniques is to obtain a nonlinear version of a linear algorithm by implicitly mapping the original data to a higher-dimensional feature space. Suppose that $\mathbf{X} = \{\mathbf{r}_i\}_{i=1}^{N}$ forms a data space of $\{\mathbf{r}_i\}_{i=1}^{N}$ and ϕ is a nonlinear function that maps the data space **X** into a feature space **F** with dimensionality yet to be determined by a kernel, that is,

$$\phi : \mathbf{X} \to \mathbf{F} \quad \text{by} \quad \mathbf{x} \mapsto \phi(\mathbf{x}) \tag{2.84}$$

It is shown in Scholkopf and Smola (2002) that an effective kernel-based learning algorithm known as "kernel trick" can be implemented. This technique implicitly computes dot products in **F** without mapping the original data sets into a feature space, which provides feasibility to implement classifiers in a higher-dimensional space. The kernel representation of dot products in **F** is expressed as

$$K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle = \phi(\mathbf{x}) \bullet \phi(\mathbf{y}) \tag{2.85}$$

where K is a kernel function of the original data. There are three common types of kernel that are generally used:

1. Polynomial kernel, $(1 + \mathbf{x}^T\mathbf{y})^p$,
2. Radial basis function kernel, $\exp\left(-\frac{1}{2\sigma^2}||\mathbf{x} - \mathbf{y}||^2\right)$, and
3. Sigmoid function kernel, $\tanh(\beta_0 + \beta_1\mathbf{x}^T\mathbf{y})$.

2.4.2 Kernel-Based Fisher's Linear Discriminant Analysis (KFLDA)

The idea of kernel-based FLDA (KFLDA) is to first map the data matrix $\mathbf{X}$ via (2.84) using a nonlinear kernel ϕ into a higher-dimensional feature space $\mathbf{F}$ in which FLDA is performed where the between-scatter and within-class scatter matrices in (2.40) and (2.41) can be reexpressed by

$$\mathbf{S}_{\mathrm{B}}^{\phi} = \sum\nolimits_{j=1}^{p} (n_j)\left(\boldsymbol{\mu}_j^{\phi} - \boldsymbol{\mu}^{\phi}\right)\left(\boldsymbol{\mu}_j^{\phi} - \boldsymbol{\mu}^{\phi}\right)^T \tag{2.86}$$

$$\mathbf{S}_{\mathrm{w}}^{\phi} = \sum\nolimits_{j=1}^{p} \sum\nolimits_{i}^{n_j} \left(\phi(\mathbf{r}_i) - \boldsymbol{\mu}_j^{\phi}\right)\left(\Phi(\mathbf{r}_i) - \boldsymbol{\mu}_j^{\phi}\right)^T \tag{2.87}$$

where $\boldsymbol{\mu}^{\phi} = (1/N)\sum_{i=1}^{N}\phi(\mathbf{r}_i)$ and $\boldsymbol{\mu}_j^{\phi} = \sum_{\mathbf{r}_i \in C_j}\phi(\mathbf{r}_i) = \sum_{i=1}^{n_j}\phi(\mathbf{r}_i)$. Using (2.86) and (2.87) the Fisher's ratio in the feature space $\mathbf{F}$ can be derived as follows.

$$J^{\phi}(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_{\mathrm{B}}^{\phi}\mathbf{w}}{\mathbf{w}^T\mathbf{S}_{\mathrm{w}}^{\phi}\mathbf{w}} \tag{2.88}$$

where $\mathbf{w}$ is a nonzero vector in $\mathbf{F}$. According to the theory of reproducing kernel, any data vector in the feature space $\mathbf{F}$ can be also represented as a linear combination of $\{\phi(\mathbf{r}_1), \phi(\mathbf{r}_2), \cdots, \phi(\mathbf{r}_N)\}$ by

$$\mathbf{w} = \sum\nolimits_{i=1}^{N} w_i\phi(\mathbf{r}_i) = \boldsymbol{\phi}^T\boldsymbol{\omega} \text{ for some combinatorial vector } \boldsymbol{\omega} \tag{2.89}$$

where $\boldsymbol{\phi} = (\phi(\mathbf{r}_1), \phi(\mathbf{r}_2), \cdots, \phi(\mathbf{r}_N))^T$ and $\boldsymbol{\omega} = (\omega_1, \omega_2, \cdots, \omega_N)^T$. Substituting (2.89) in (2.86) and (2.87) yields

$$\mathbf{w}^T\mathbf{S}_{\mathrm{B}}^{\phi}\mathbf{w} = \boldsymbol{\omega}^T\mathbf{K}_{\mathrm{B}}^{\phi}\boldsymbol{\omega} \tag{2.90}$$

$$\mathbf{w}^T\mathbf{S}_{\mathrm{W}}^{\phi}\mathbf{w} = \boldsymbol{\omega}^T\mathbf{K}_{\mathrm{W}}^{\phi}\boldsymbol{\omega} \tag{2.91}$$

where $\mathbf{K}_{\mathrm{B}}^{\phi}$ and $\mathbf{K}_{\mathrm{W}}^{\phi}$ are obtained by the kernel trick. Using (2.90) and (2.91), (2.88) can be reexpressed as

$$J^{\phi}(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{S}_{\mathrm{B}}^{\phi}\mathbf{w}}{\mathbf{w}^T\mathbf{S}_{\mathrm{W}}^{\phi}\mathbf{w}} = \frac{\boldsymbol{\omega}^T\mathbf{K}_{\mathrm{B}}^{\phi}\boldsymbol{\omega}}{\boldsymbol{\omega}^T\mathbf{K}_{\mathrm{W}}^{\phi}\boldsymbol{\omega}} \tag{2.92}$$

The optimal solution to maximizing (2.92) is given by a set of $p-1$ eigenvectors $\mathbf{w}_1^{\mathrm{FLDA}}, \mathbf{w}_2^{\mathrm{FLDA}}, \cdots, \mathbf{w}_{p-1}^{\mathrm{FLDA}}$ that solve the following generalized eigenvalue problem

$$\mathbf{K}_{\mathrm{B}}^{\phi}\mathbf{w}_j^{\mathrm{FLDA}} = \lambda_i\mathbf{K}_{\mathrm{W}}^{\phi}\mathbf{w}_j^{\mathrm{FLDA}} \quad \text{for} \quad 1 \leq j \leq p-1 \tag{2.93}$$

For a new data sample vector $\phi(\mathbf{r}_i)$ in the feature space $\mathbf{F}$ its projection onto the optimal jth discriminant vector $\mathbf{w}_j^{\mathrm{FLDA}}$ is given by

$$\left(\mathbf{w}_j^{\mathrm{FLDA}}\right)^T\phi(\mathbf{r}_i) = \sum\nolimits_{k=1}^{N}\omega_k^{\mathrm{FLDA}}\phi(\mathbf{r}_{jk})\phi(\mathbf{r}_i) = \sum\nolimits_{k=1}^{N}\omega_k^{\mathrm{FLDA}}K(\mathbf{r}_k, \mathbf{r}_i) \tag{2.94}$$

where

$$\mathbf{w}_j^{\text{FLDA}} = \sum\nolimits_{k=1}^{N} \omega_k^{\text{FLDA}} \phi(\mathbf{r}_k) = \boldsymbol{\phi}\boldsymbol{\omega}^{\text{FLDA}} \quad \text{for some combinatorial vector } \boldsymbol{\omega}^{\text{FLDA}} \tag{2.95}$$

Let $\mathbf{W}^{\text{FLDA}} = \left[\mathbf{w}_1^{\text{FLDA}}\mathbf{w}_2^{\text{FLDA}} \cdots \mathbf{w}_{p-1}^{\text{FLDA}}\right]$ be the optimal feature matrix formed by $\mathbf{w}_1^{\text{FLDA}}, \mathbf{w}_2^{\text{FLDA}}, \cdots, \mathbf{w}_{p-1}^{\text{FLDA}}$. The data space $\tilde{\mathbf{X}}$ in the feature space $\mathbf{F}$ can be obtained via the optimal discriminant feature matrix $\mathbf{W}^*$ specified by (2.95) as follows

$$\tilde{\mathbf{X}} = \mathbf{W}^{\text{FLDA}}\phi(\mathbf{X}) = \left[\mathbf{w}_1^{\text{FLDA}}\mathbf{w}_2^{\text{FLDA}} \cdots \mathbf{w}_{p-1}^{\text{FLDA}}\right]^T \phi(\mathbf{X}) \tag{2.96}$$

where $\mathbf{X}$ is the original data space. KFLDA is then used to perform the FLDA on $\tilde{\mathbf{X}}$ and is expected to perform better than FLDA directly operating on the original space $\mathbf{X}$.

2.4.3 Kernel Support Vector Machine (K-SVM)

The SVM discussed in Section 2.3.1.2 is a linear machine that performs a binary decision for linear separable or nonseparable problems. As we have seen, in order for a linear SVM to solve linear nonseparable problems, a set of slack variables are introduced to resolve the dilemma caused by confusing data sample vectors. In this section, we consider a rather different approach that generalizes the linear discriminant function $g(\mathbf{r}) = \mathbf{w}^T\mathbf{r} + b$ to allow a much larger class of possible decision boundaries by transforming an original data sample $\mathbf{x}$ via a set of M predetermined nonlinear functions $\phi_j(\mathbf{r})$, $\{\phi_j(\mathbf{r})\}_{j=1}^M$ referred to as basis functions or kernels. The output of $\mathbf{r}$ resulting from such a nonlinear transform can be represented by a linear combination of these functions $\{\phi_j(\mathbf{r})\}_{j=1}^M$ as follows

$$y_k = g_k(\mathbf{r}) = \sum\nolimits_{j=1}^{M} w_{kj}\phi_j(\mathbf{r}) + b_k \tag{2.97}$$

In order to absorb the bias b_k in the linear sum in (2.97), we can introduce an auxiliary function $\phi_0(\mathbf{r}) = 1$ so that the Equation (2.71) can be reexpressed as a linear form by

$$y_k = g_k(\mathbf{r}) = \sum\nolimits_{j=0}^{M} w_{kj}\phi_j(\mathbf{r}) = \mathbf{w}_k^T\boldsymbol{\phi}(\mathbf{r}) \tag{2.98}$$

where $\boldsymbol{\phi}(\mathbf{r}) = (\phi_0(\mathbf{r}), \phi_1(\mathbf{r}), \cdots, \phi_M(\mathbf{r}))^T$ and the weight vector $\mathbf{w}_k$ is represented by $\mathbf{w} = (w_{k0}, w_{k1}, \cdots, w_{kM})^T$ with $w_{k0} = b_k$. By adapting (2.98) we can also obtain a kernel-based SVM specified by the weight vector $\mathbf{w}^{\text{SVM}}$ given by

$$\mathbf{w}^{\text{SVM}} = \sum\nolimits_{i=1}^{n} \alpha_i^{\text{SVM}} d_i \boldsymbol{\phi}(\mathbf{r}_i) \tag{2.99}$$

$$\sum\nolimits_{i=1}^{n} \alpha_i^{\text{SVM}} y_i \boldsymbol{\phi}(\mathbf{r}_i)\boldsymbol{\phi}(\mathbf{r}) = 0 \tag{2.100}$$

If we further define the kernel $K(\mathbf{r},\mathbf{r}_i)$ as the inner product of $\mathbf{r}$ and $\mathbf{r}_i$ by

$$K(\mathbf{r}, \mathbf{r}_i) = \boldsymbol{\phi}^T(\mathbf{r})\boldsymbol{\phi}(\mathbf{r}_i) = \sum\nolimits_{j=1}^{M} \phi_j(\mathbf{r})\phi_j(\mathbf{r}_i) \tag{2.101}$$

we can form a kernel-based SVM (KSVM) as follows.

Given a set of training pool, $\{(\mathbf{r}_i, d_i)\}_{i=1}^n$, find a set of Lagrange multiplier vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \cdots, \alpha_n)^T$ that maximizes $Q(\boldsymbol{\alpha})$, which is modified from (2.60) and given by

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - (1/2) \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j d_i d_j K(\mathbf{r}_i, \mathbf{r}_j) \tag{2.102}$$

subject to the following constraints:

1. $\sum_{i=1}^{n} \alpha_i d_i = 0$
2. $C \geq \alpha_i \geq 0$ for $1 \leq i \leq n$, with C being a user-specified positive parameter.

An optimal solution to the above kernel-based SVM optimization problem is given by

$$\mathbf{w}^{\text{SVM}} = \sum_{i=1}^{n_s} \alpha_i^{\text{SVM}} d_i^s \mathbf{r}_i^s \tag{2.103}$$

where n_s is the number of support vectors.

The key issue to solve the KSVM is to determine the kernel used in (2.103). Interestingly, according to Mercer's theorem, if a kernel-based inner product $K\,(\mathbf{x},\mathbf{y})$ is continuous and symmetric in a closed interval $a \leq \mathbf{x}, \mathbf{y} \leq b$, $K\,(\mathbf{x},\mathbf{y})$ can be expanded in a series

$$K(\mathbf{x}, \mathbf{y}) = \sum_{j=1}^{\infty} \lambda_j \phi_j(\mathbf{x}) \phi_j(\mathbf{y}) \tag{2.104}$$

where $\{\lambda_j\}$ are eigenvalues and $\{\phi_j\}$ are their associated eigenfunctions.

The KFLDA and KSVM discussed above are derived from two popular hard decision-made classifiers, FLDA and SVM. Correspondingly, there should also be kernel versions of soft decision-made classifiers as discussed in Section 2.3.2.1 such as OSP. However, their derivations are much more involved. Therefore, their kernel versions, referred to as kernel-based linear spectral mixture analysis (KLSMA), will be developed and discussed in a separate and individual chapter, Chapter 15, which includes kernel versions of various LSMA-based classifiers.

2.5 Conclusions

Subsample detection and mixed sample classification play central roles in hyperspectral image analysis. However, it seems that there is a lack of detailed treatment on decision issues of these two topics. Generally speaking, a subsample is a target sample of interest embedded in a sample with an unknown proportion while the remaining proportion of its embedded sample is considered as the background. On the other hand, a mixed sample comprises a set of known target signatures mixed linearly or nonlinearly in a sample. In light of this interpretation a major difference between a subsample and a mixed sample is that the background of a sample in which a subsample is embedded is unknown, while the background of a mixed sample is completely specified by other known target signatures. As a result, there is no background issue in a mixed sample and a mixed sample is generally performed by classification. By contrast, the background of a subsample is generally unknown and unspecified. Therefore, a subsample is usually performed by detection rather than by classification and the effectiveness of the subtarget detection is heavily determined by background suppression. Because of that this chapter is devoted to detection and classification via two types of decision-making processes, hard decisions and soft decisions, to address the issues of subsamples and mixed samples. As for detection, two approaches are reviewed. One is the likelihood ratio test using a threshold in

Table 2.1 Summary of pure/subsample detectors and pure/mixed-sample classifiers

	Signal model	Linear model	$\mathbf{n}$	$\mathbf{B}$	Formula	Equation numbers
LRT $\delta(\mathbf{r})$	(2.12)	No	Yes	Yes	$\Lambda(\mathbf{r}) = \frac{p_1(\mathbf{r})}{p_0(\mathbf{r})}$	(2.8)
AMD $\delta^{\text{AMD}}(\mathbf{r})$	(2.13)	No	Yes	Yes	$(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T \mathbf{K}^{-1} \mathbf{r}$	(2.17)
NAMD $\delta^{\text{NAMD}}(\mathbf{r})$	(2.13)	No	Yes	Yes	$\frac{(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T \mathbf{K}^{-1} \mathbf{r}}{(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T \mathbf{K}^{-1} (\bar{\mathbf{t}} - \bar{\mathbf{b}})}$	(2.19)
ASD $\delta^{\text{ASD}}(\mathbf{r})$	No	No	$\mathbf{K}$	No	$\frac{\mathbf{t}^T \mathbf{K}^{-1} \mathbf{r}}{\mathbf{t}^T \mathbf{K}^{-1} \mathbf{t}}$	(2.28)
CEM $\delta^{\text{CEM}}(\mathbf{r})$	No	No	$\mathbf{R}$	No	$\frac{\mathbf{t}^T \mathbf{R}^{-1} \mathbf{r}}{\mathbf{t}^T \mathbf{R}^{-1} \mathbf{t}}$	(2.33)
FLDA $\delta^{\text{FLDA}}(\mathbf{r})$	No	No	No	No	$\kappa(\bar{\mathbf{t}} - \bar{\mathbf{b}})^T \mathbf{S}_W^{-1} \mathbf{r}$	(2.39)
OSP $\delta^{\text{OSP}}(\mathbf{r})$	No	(2.76)	No	$\mathbf{U}$	$\kappa \mathbf{d}^T \mathbf{P}_{\mathbf{U}}^{\perp} \mathbf{r}$	(2.77)
LSOSP $\delta^{\text{LSOSP}}(\mathbf{r})$	No	(2.76)	No	$\mathbf{U}$	$\frac{\mathbf{d}^T \mathbf{P}_{\mathbf{U}}^{\perp} \mathbf{r}}{\mathbf{d}^T \mathbf{P}_{\mathbf{U}}^{\perp} \mathbf{d}}$	(2.79)
TCIMF $\delta^{\text{TCIMF}}(\mathbf{r})$	No	(2.76)	$\mathbf{R}$	$\mathbf{DU}$	$\left(\mathbf{R}^{-1}(\mathbf{DU})[(\mathbf{DU})^T \mathbf{R}^{-1} (\mathbf{DU})]^{-1} \begin{bmatrix} 1_{n_D \times 1} \\ 0_{n_U \times 1} \end{bmatrix}\right)^T \mathbf{r}$	(2.82)

terms of a cost function or the false alarm probability, which can be derived from a standard binary composite hypothesis testing approach that can be implemented as either pure sample detection or subsample detection. The other includes generalized LRT (GLRT)-based CFAR and the constrained energy minimization, both of which are developed for subsample detection with soft decisions. In analogy with detection, classification with hard and soft decisions is also considered. The classification with hard decisions is addressed by two pure sample-based classification techniques, FLDA and SVM, while classification with soft decisions is dealt by two mixed sample-based classification techniques, orthogonal subspace projection and target-constrained interference-minimization filter. Table 2.1 summarizes pure/subsample detectors and pure/mixed-sample classifiers presented in this chapter where κ, $\mathbf{n}$, $\mathbf{B}$, $\mathbf{D}$, and $\mathbf{U}$ represent a scaling constant, noise, background, and desired and undesired signals, respectively and a “yes” or “no” indicates that the knowledge is required or not required. In addition, a “**K**” or an “**R**” indicates sample covariance or correlation matrix is required without knowing noise statistics. It should be noted that SVM is not included in Table 2.1 due to the fact that the training samples used in SVM are support vectors, which can be considered as worst samples for training. As a result, the mean of support vectors of each class cannot represent the mean of the class to which they belong. In this case, the target and/or background knowledge generated by these support vectors is not reliable and not representative.

3

Three-Dimensional Receiver Operating Characteristics (3D ROC) Analysis

Receiver operating characteristics (ROC) analysis is a widely used performance evaluation tool in signal processing, communications, and medical diagnosis. It utilizes two-dimensional (2D) curves plotted between detection rate (P_{D}) and false alarm rate (P_{F}) to assess effectiveness of a detector or sensor/device for detection. However, P_{D} and P_{F} are actually dependent parameters resulting from a more crucial but implicit parameter hidden in the ROC curves, that is, threshold τ, which is determined by the cost of implementing a detector or sensor/device except the case when the Bayes theory is used for detection, where τ is completely determined by the Bayes cost. This chapter extends the traditional ROC analysis for single-signal detection to multiple-signal detection and classification. It also explores the relationship among the three parameters, P_{D}, P_{F}, and τ, and further develops a new concept of three-dimensional ROC (3D ROC) analysis that uses 3D ROC curves plotted with P_{D}, P_{F}, and τ to evaluate detection effectiveness rather than only P_{D} and P_{F} used by 2D ROC analysis. As a result of using the 3D ROC curves, three 2D ROC curves can also be derived, the conventional 2D ROC curve plotted by P_{D} versus P_{F} and two new 2D ROC curves plotted by P_{D} versus τ and P_{F} versus τ. To demonstrate the utility of 3D ROC analysis, four applications are considered: hyperspectral target detection, medical diagnosis, chemical/biological agent detection, and biometric recognition.

3.1 Introduction

The receiver operating characteristics (ROC) analysis has been widely used in signal processing and communications to assess effectiveness of a sensor or detector/device for detection (Poor, 1994). In recent years, it has also become a common evaluation tool for effectiveness of a medical modality in medical diagnosis, specifically for computer-assisted diagnostic systems (Metz, 1978; Swets and Pickett, 1982), automatic target recognition (ATR) (Parker et al., 2005a, 2005b; Blasch and R. Broussard, 2000; Bauman et al., 2005), and fusion analysis (Blasch et al., 2001; Blasch and Plano, 2003; Blasch, 2008). The idea is simple. For a given detector or detection technique how can we objectively evaluate whether or not it is effective and in what sense? There are many criteria

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

or cost functions available for such assessment, such as least-squares error, signal-to-noise ratio, and misclassification error. Unfortunately, none of these criteria can be considered as a general criterion to fit all detection problems. For example, least-squares error or signal-to-noise ratio may be a good criterion for detection problems in signal processing and communications but may not be suitable for measuring image quality or classification accuracy in image processing. So, in order to avoid using a specific criterion for performance evaluation, the ROC analysis is introduced for this purpose. It does not necessarily specify a particular criterion or cost function. Instead, it focuses on the effect of a decision made for a detection problem regardless of what specific criterion or cost function should be used. More specifically, it casts a detection problem as a binary decision problem, that is, binary hypothesis testing problem which results in four decisions that need to be considered:

1. When the ground truth of the problem is true, we made a "not true" decision. In this case, we commit an error, referred to as "miss" or "false negative" decision.
2. When the ground truth of the problem is true, we made a "true" decision. In this case, we made a correct decision, referred to as "detection" or "true positive" decision.
3. When the ground truth of the problem is not true, we made a "true" decision. In this case, we commit another type of error, referred to as "false alarm" or "false positive" decision.
4. When the ground truth of the problem is not true, we made a "not true" decision. In this case, we made another type of correct decision, referred to as "true negative" decision.

Using the above four decisions we can evaluate a given detector or a detection technique according to its effectiveness without actually appealing for a specific performance criterion or cost function. Since two error decisions contradict each other, a general practice is to choose the "false alarm," that is, "false positive" as a base to produce the best decision. In other words, by constraining the false alarm rate to a certain level for which the considered problem can tolerate, what can a detector or detection technique achieve as the best detection power in terms of probability? The ROC analysis is developed to address this issue. For any given detector or detection technique the ROC analysis plots a curve, referred to as ROC curve, based on the probability of detection power, that is, detection rate versus false alarm rate where an ROC curve is a function of two parameters, the detection rate P_{D} and the false alarm rate P_{F}, and is used to evaluate how effective or good a given detector or detection technique is. For example, suppose that two detectors or detection techniques are considered for performance evaluation. To see which one performs better, we first generate and compare their ROC curves. If for any given false alarm rate the detection rate of one technique is higher than the other, we can conclude that this technique is more effective or better than the other. Since the ROC curve is always convex, we can calculate the areas under their ROC curves, called area under curve (AUC), A_z instead of examining their ROC curves without actually calculating their individual pairs (false alarm rate, detection rate). As a result, the higher the value of A_z is, the better the detection performance.

Despite that the ROC analysis does not use any cost function it does use P_{F} as a cost measure to evaluate detection performance where the P_{F} indeed involves with an implicit parameter hidden in the P_{F}, called threshold τ, which actually determines the P_{F} (see the Neyman–Pearson detector specified in Section 3.5). In other words, it is τ, which is a real cost parameter, that implements a detector with both P_{D} and P_{F} calculated as functions of τ. Unfortunately, this τ is not shown in the ROC analysis since the P_{F} of an ROC curve is considered as an independent variable ranging from 0 to 1 rather than a variable dependent on τ. To deal with such an issue, a concept of developing a three-dimensional (3D) ROC was first envisioned by Alsing et al. (1999) who introduced 3D ROC trajectory by including a third parameter, such as probability of rejection, to assess the detector's degree of difficulty to recognize unknown targets due to the lack of confidence. Nearly at the same time, Chang et al. (2001b) also proposed a rather different concept that directly involves τ to determine P_{F} and P_{D}. It argued that threshold τ ultimately determines the detection performance. When a detector is not ready to make its decision due to lack of confidence in provided evidence, two approaches can resolve this dilemma. One is to reject the decision as proposed by Alsing et al. (1999) with applications to fusion analysis for user-synthetic aperture radar (SAR)-ATR systems (Plano and Blasch, 2003) rather than making a hard decision. The other is to make a soft decision based on likely cost determined by the threshold τ (i.e., in this case, the likelihood ratio test is equal to the threshold τ in (2.6) or (2.8)). Under this circumstance, the detector is forced to make a random decision according to the likelihood of each decision in terms of probability calculated by the threshold τ. The resulting detector is called a randomized detector. The work using a threshold representing the likelihood of the signal presence in Parker et al. (2005a) and the work using the confidence error as a criterion in Parker et al. (2005b) are good examples in this aspect. As a matter of fact, a random decision is better than a rejection because a randomized detector offers likelihood of each decision to be made in terms of its probability compared to the latter which simply does not make a decision by rejection. In a broader sense, a rejection can be interpreted in the context of a random decision where the probability of rejection actually describes the likelihood of a decision to be rejected. Since a great deal of research effort devoted to the 3D ROC analysis using the probability of rejection as a third parameter has been made in Alsing et al. (1999) and Plano and Blasch (2003), this chapter will focus only on the development of a 3D ROC analysis using threshold τ as the third parameter which has been investigated in hyperspectral imaging (Chang et al., 1998; Chang, 2002, 2003a), magnetic resonance imaging (Wang et al., 2003, 2005; Chen et al., 2005), chemical/biological agent detection (Chang, 2006; Liu et al., 2005), and biometrics identification (Du and Chang, 2007, 2008) where a 3D ROC curve can be plotted according to the three parameters, P_{D}, P_{F}, and threshold τ. This is because a detector implemented in the above-mentioned applications is actually an estimator whose estimated values represent the strength of signal detectability used to perform signal detection via a threshold τ that helps a signal estimator make a binary decision. By virtue of these three parameters, P_{D}, P_{F}, and τ, we can derive a 3D ROC analysis to generate a 3D ROC curve as a function of P_{D}, P_{F}, and τ. As a result of a 3D ROC curve, three 2D ROC curves can also be derived and plotted. One is an ROC curve of $(P_{\mathrm{D}},P_{\mathrm{F}})$ which turns out to be the ROC curve by the traditional ROC analysis. The other two are new 2D ROC curves, which are the ROC curves of (P_{D},τ) and (P_{F},τ).

Since the Neyman–Pearson detection theory is mainly focused on detection of signal in noise the decision of detecting noise, that is, the fourth decision described earlier, does not make any sense. However, in other applications such as medical diagnosis the probability of making the fourth decision of true negative represents specificity of a medical modality. Besides, when signals to be detected are multiple signals, it may require multiple hypotheses to perform multisignal detection. This problem can be actually addressed by signal classification problems where different threshold values of τ are required to classify different signal classes. Unfortunately, the 2D ROC curve of $(P_{\mathrm{D}},P_{\mathrm{F}})$ does not provide such information of τ. This chapter makes an attempt to address this need and explores 3D ROC analysis and its utility in four different applications.

3.2 Neyman–Pearson Detection Problem Formulation

In Section 2.2.1 a binary hypothesis testing problem (2.1) is used to formulate the pure-sample target detection as two hypotheses, H_0 and H_1, which represent the absence and presence of a signal source in an observed sample $\mathbf{r}$, respectively. This section places its main focus on a particular type of detection problem when there is no prior knowledge of the two hypotheses and cost functions. It is generally called the Neyman–Pearson detection problem cast by (2.9)–(2.11).

More specifically, assume that the observation process is described by a random process Y_t. When this process is observed at a particular time instant $t = t_0$, it is referred to as an observation y which can be described by a random variable Y_{t_0}. If the probability distribution of Y_{t_0} is further assumed to be $P(y)$ with its probability density function given by $p(y)$, the binary hypothesis testing problem (2.1) can be described by

$$\begin{aligned} &H_0: \quad Y_{t_0} \approx p_0(y) \\ &\text{versus} \\ &H_1: \quad Y_{t_0} \approx p_1(y) \end{aligned} \tag{3.1}$$

where the hypotheses H_0 and H_1 can be observed from the variable Y_{t_0} whose probability distributions are derived from $p(y)$ under each hypothesis, denoted by $p_0(y)$ under H_0 and $p_1(y)$ under H_1. The hypotheses "H_0" and "H_1" are generally called "*null hypothesis*" and "*alternative hypothesis*," respectively. In applications of signal processing and communications, "H_1," such as in Section 2.2.1, represents the case when the signal is present along with noise, while the hypothesis "H_0" indicates that no signal is present. So, when hypothesis "H_1" is true, it implies that there is a signal present in the observation y.

Assume that the observation y is the data sample vector denoted by $\mathbf{r}$ and the observation space Γ is denoted by data sample vector space. Using (2.2) and Figure 2.1 and renumbering (2.9)–(2.11), $P_{\mathrm{D}}(\delta)$ is defined as the detection probability/rate or detection power specified by (2.10),

$$P_{\mathrm{D}}(\delta) = P_1(\Gamma_1) = \int_{\Gamma_1} p_1(\mathbf{r})d\mathbf{r} \tag{3.2}$$

and $P_{\mathrm{F}}(\delta)$ as the false alarm probability/rate specified by (2.9),

$$P_{\mathrm{F}}(\delta) = P_0(\Gamma_1) = \int_{\Gamma_1} p_0(\mathbf{r})d\mathbf{r}. \tag{3.3}$$

An NP detector, denoted by $\delta^{\mathrm{NP}}(\mathbf{r})$, is obtained by solving the following constrained optimization problem specified by (2.11) and recapped as follows:

$$\max_\delta\{\mathrm{P_D}(\delta)\} \text{ subject to } P_{\mathrm{F}}(\delta) \leq \beta \text{ for any given } 0 \leq \beta \leq 1 \tag{3.4}$$

where β is prescribed and known as the significant level of test and the maximum is sought over all possible decision rules, $\delta(\mathbf{r})$. The well-known Neyman–Pearson lemma shows that the optimum solution to (3.4) turns out to be a likelihood ratio test, $\Lambda(\mathbf{r})$, similar to (2.8) and given by

$$\delta^{\mathrm{NP}}(\mathbf{r}) = \begin{cases} 1 & \text{if } \Lambda(\mathbf{r}) \geq \tau \\ 1 & \text{with probability } \kappa \text{ if } \Lambda(\mathbf{r}) = \tau \\ 0 & \text{if } \Lambda(\mathbf{r}) < \tau \end{cases} \tag{3.5}$$

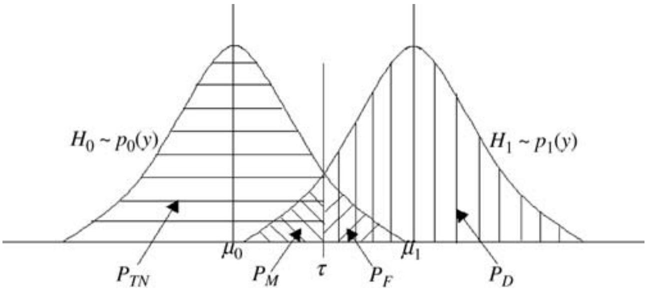


Figure 3.1 An illustration of probabilities P_{D}, P_{F}, P_{M}, and P_{TN}.

where $\Lambda(\mathbf{r}) = p_0(\mathbf{r})/p_1(\mathbf{r})$, the threshold τ is determined by the constraint β, and γ is the probability of saying H_1 when $\Lambda(\mathbf{r}) = \tau$. By virtue of $\delta^{\mathrm{NP}}(\mathbf{r})$, P_{D} and P_{F} in (3.2)–(3.3) can be obtained and expressed as follows:

$$P_{\mathrm{D}} = \int_{\Lambda(\mathbf{r})>\tau} p_1(\mathbf{r})\mathrm{d}\mathbf{r} + \kappa P(\{\mathbf{r}|\Lambda(\mathbf{r}) = \tau\}) \tag{3.6}$$

$$P_{\mathrm{F}} = \int_{\Lambda(\mathbf{r})>\tau} p_0(\mathbf{r})\mathrm{d}\mathbf{r} + \kappa P(\{\mathbf{r}|\Lambda(\mathbf{r}) = \tau\}) \tag{3.7}$$

with τ determined by $P_{\mathrm{F}} = \beta$ via (3.3). Details of signal detection theory can be found in Poor (1994). Figure 3.1 shows how a decision can be made by adjusting threshold τ using (3.6) and (3.7) where the regions corresponding to four decisions described in Section 3.1 $P_{\mathrm{M}} = 1 - P_{\mathrm{D}}$ (decision 1), P_{D} (decision 2), P_{F} (decision 3), and $P_{\mathrm{TN}} = 1 - P_{\mathrm{F}}$ (decision 4) are indicated with different shaded areas.

A final remark is noteworthy. According to standard detection theory four types of decisions described in Section 3.1 can be made based on the binary hypothesis testing problem specified by (3.1). However, in some applications there is a fifth decision that is no action to be taken due to insufficient knowledge. For example, in automatic target recognition (Parker et al., 2005a, 2005b; Blasch et al., 1999; Blasch and Broussard, 2000; Bauman et al., 2005), documentation analysis such as character recognition (Suen et al., 1980) and text detection in video images (Du et al., 2003), and biometric recognition (Du and Chang, 2008), when the level of confidence of making a decision is low due to the lack of knowledge, an alternative option is rejection. However, this approach can be actually interpreted in the context of what is the so-called randomized decision in statistical detection theory. When the detector statistics, $\Lambda(\mathbf{r})$ specified by (3.5), is equal to the threshold τ, two actions can be taken. One is doing nothing but rejection which is the fifth type of decision as described above in addition to the four described in Section 3.1. The other is to force the detector to make a soft decision on two hypotheses to express the confidence level of each decision in terms of its probability. As a consequence, such a random decision is better off than a rejection since the latter simply does not make any decision by rejecting.

3.3 ROC Analysis

Interestingly, according to (2.8), (2.11), and (3.5), no matter what the type of detector is, its detector statistics always turns out to be the likelihood ratio test (LRT). In other words, all the Bayes, minimax, or Neyman–Pearson detectors can be shown to have the same functional form determined by LRT. Nevertheless, NP detectors are the most practical detectors in real applications since they do not require prior knowledge of the cost function and probabilities of hypotheses that are generally unknown or difficult to obtain in practice. Instead, the performance is evaluated based on the four decisions described in Section 3.1 but with general descriptive terms given in the following:

1. When "H_0" is true, "H_0" is also true.
   In this case, a correct decision is made and the decision is called "*true negative*" (TN). However, it should be noted that there is no detection term corresponding to this decision since nothing is detected other than noise.
2. When "H_0" is true, "H_1" is true.
   In this case, an incorrect decision is made and the decision is called "*false alarm*" (FA) or "*false positive*" (FP) in medical diagnosis.
3. When "H_1" is true, "H_0" is also true.
   In this case, an incorrect decision is made and the decision is called "*miss*" or "*false negative*" (FN).
4. When "H_1" is true, "H_1" is also true.
   In this case, a correct decision is made and the decision is called "*detection probability*, *rate or power*" or "*true positive*" (TP).

As per the above decisions, we have two correct decisions 1 and 4, that is, TN and detection rate or power or true positive (TP), and two incorrect decisions 2 and 3, that is, FA (FP) and miss (FN). Consequently, a good detector must be the one that maximizes the probabilities yielded by the correct decisions, TN and detection rate/TP, and in the mean time it also minimizes the probabilities resulting from the two incorrect decisions, FA/FP and miss/FN. However, it is generally true that the two incorrect decisions contradict each other, so do the correct decisions. In other words, minimizing miss/FN is also to increase FA/FP and vice versa. Since minimizing miss/FN is equivalent to maximizing detection power/TP, a common practice is to maximize the detection rate/TP while imposing a constraint on FA/FP specified by (3.4). Using (3.6) and (3.7) we can further derive

$$\begin{aligned} P_{\mathrm{M}}(\delta^{\mathrm{NP}}) &= \int_{\Lambda(\mathbf{r})<\tau} p_1(\mathbf{r})d\mathbf{r} + (1-\kappa)P(\{\mathbf{r}|\Lambda(\mathbf{r})=\tau\}) \\ &= 1 - P_{\mathrm{D}} \end{aligned} \tag{3.8}$$

$$P_{\mathrm{TN}}(\delta^{\mathrm{NP}}) = \int_{\Lambda(\mathbf{r})<\tau} p_0(\mathbf{r})d\mathbf{r} + \kappa P(\{\mathbf{r}|\Lambda(\mathbf{r})=\tau\}) = 1 - P_{\mathrm{F}}. \tag{3.9}$$

To evaluate the detection performance of $\delta^{\mathrm{NP}}(\mathbf{r})$, the ROC analysis is commonly used as an evaluation tool to assess the effectiveness of a detector based on an ROC curve plotted as a function of P_{D} versus P_{F} as shown in Figure 3.2. As an alternative to the use of ROC curves, the area under curve (AUC), A_z, which has been widely used in medical diagnosis (Metz, 1978; Swets and Pickett, 1982), is also calculated by the area under an ROC curve.

The use of A_z has advantages over the ROC curves. For example, on some occasions two detectors δ^1 and δ^2 may generate two different 2D ROC curves but have the same area, A_z, as shown in Figure 3.3. In this case, both δ^1 and δ^2 yield the same detection performance even when they have different ROC curves.

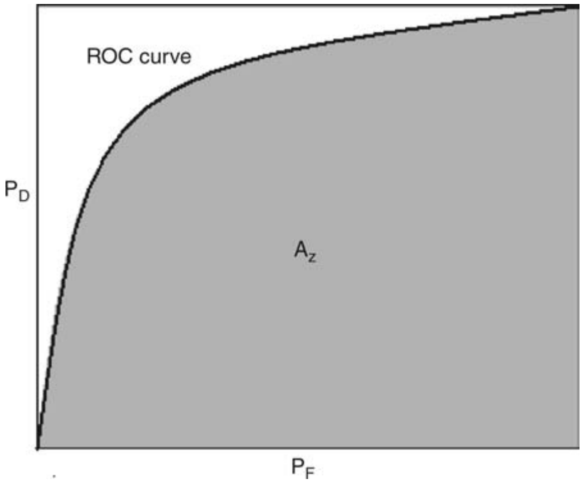


Figure 3.2 An example of AUC, A_z, calculated by an ROC curve.

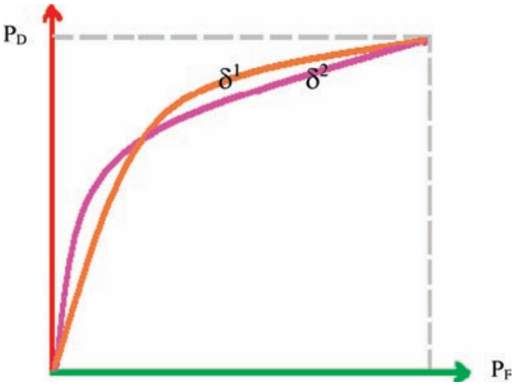
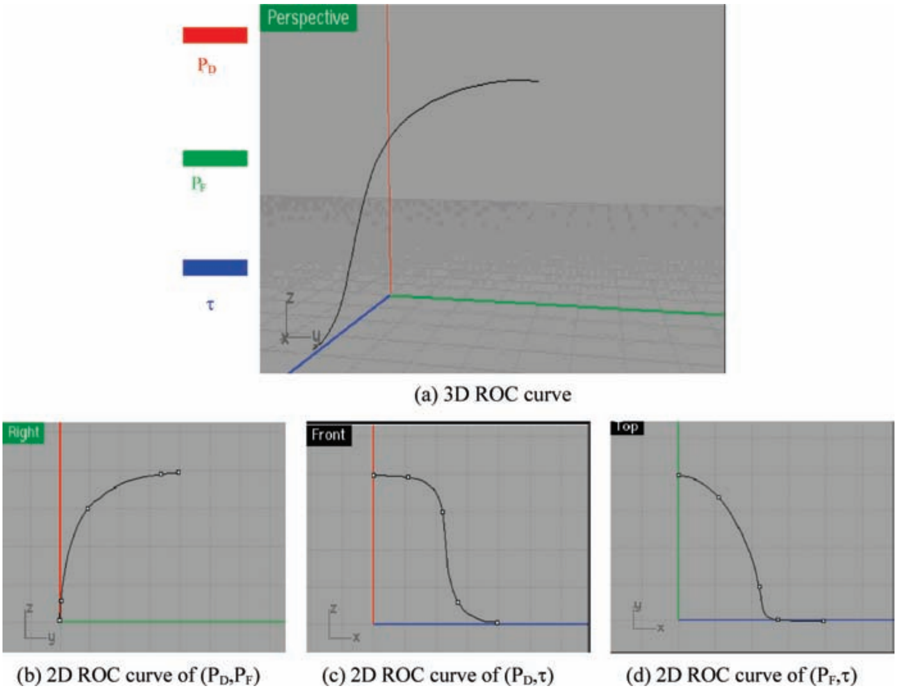


Figure 3.3 An example of ROC curves generated by two detectors with the same A_z.

3.4 3D ROC Analysis

From the hypothesis testing problem specified by (3.1), a detector makes a binary hard decision by thresholding a real-valued LRT, $\Lambda(\mathbf{r})$, via a threshold τ (see (3.5)). Accordingly, the detector performance is determined by two parameters, $\Lambda(\mathbf{r})$ and τ, both of which are real values. As a result, the detection rate, P_{D}, in (3.2) and (3.6) and the false alarm probability/rate P_{F} in (3.3) and (3.7) are indeed functions of $\Lambda(\mathbf{r})$ and threshold τ. However, in the Neyman–Pearson detection theory the cost function $\{c_{ij}\}$ and prior probabilities $\{\pi_i\}$ are assumed to be not known, nor is τ. In this case, the false alarm rate P_{F} is used as a cost function and the threshold τ becomes a dependent function of P_{F} via (3.7) by setting $P_{\mathrm{F}} = \beta$ in (3.4). This is contradictory to the original detection problem where $P_{\mathrm{F}} = \beta$ is actually obtained by a specific value of the threshold τ. Therefore, when an ROC curve is plotted in Figure 3.4 based on P_{D} versus P_{F}, the threshold τ is implicitly absorbed in P_{F} and there is no way to show how the threshold τ specifies P_{F} as the way it should be in Bayes detection theory in (3.1). To resolve this issue, this section develops a new approach to ROC



(a) 3D ROC curve

(b) 2D ROC curve of (P_D,P_F) (c) 2D ROC curve of (P_D,τ) (d) 2D ROC curve of (P_F,τ)

Figure 3.4 An example of a 3D ROC curve along with three 2D ROC curves.

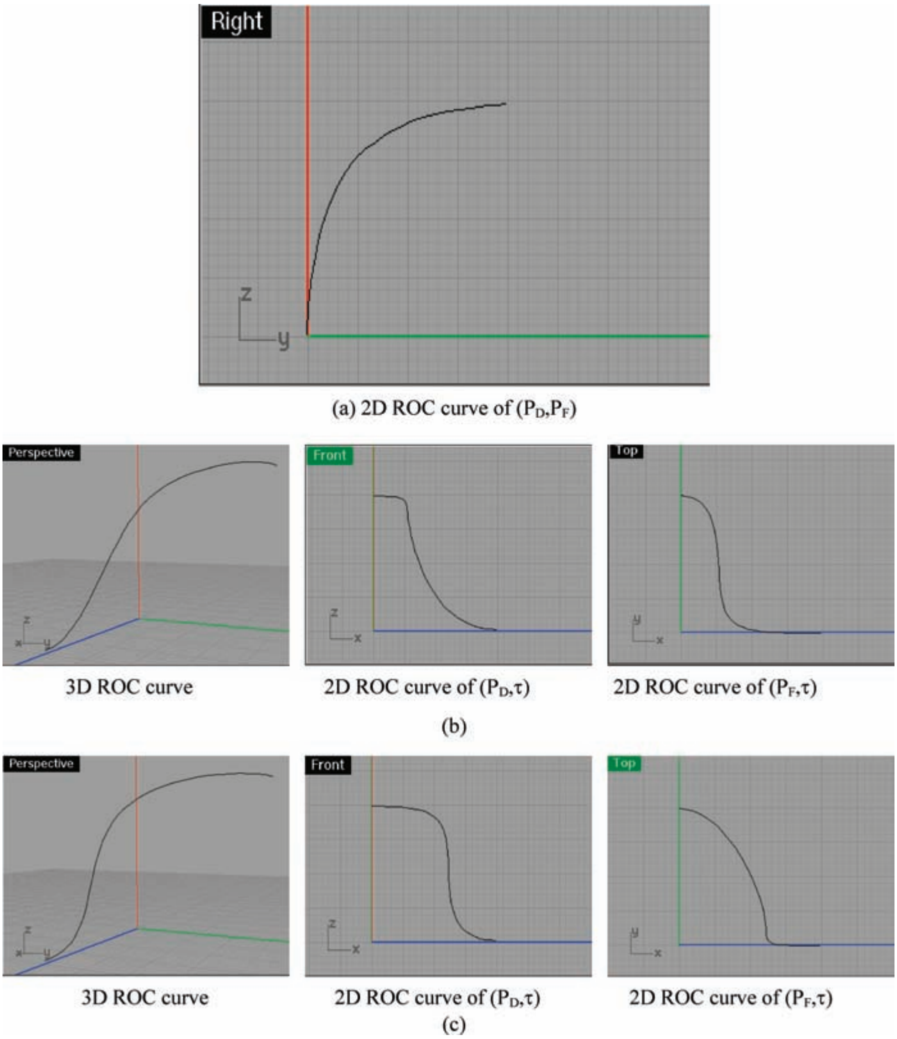
analysis, referred as 3D ROC analysis, which extends the traditional 2D ROC analysis in Section 3.3 by including the threshold τ as a third parameter along with P_D and P_F used in the 2D ROC analysis. In other words, the proposed 3D ROC analysis makes use of a 3D ROC curve plotted based on three parameters, P_D, P_F and τ, referred to as the 3D ROC curve of (P_D,P_F,τ) from which three new 2D ROC curves can be derived, the 2D ROC curve of (P_D,P_F), the 2D ROC curve of (P_D,τ), and the 2D ROC curve of (P_F,τ), where the 2D ROC curve of (P_D,P_F) turns out to be the traditional ROC curve in Figure 3.2. An example of a 3D ROC curve along with its three 2D ROC curves is shown in Figure 3.4, where the x, y, and z axes are specified by P_F (green), τ (blue), and P_D (red). In Figure 3.4(a), a 3D ROC curve is plotted in a perspective view in Figure 3.4(a) where the three 2D ROC curves, that is, 2D ROC curve of (P_D,P_F), 2D ROC curve of (P_D,τ), and 2D ROC curve of (P_F,τ), are plotted in Figures 3.4(b)-3.4(d), respectively, in the right, front and top views.

As shown by the two new 2D ROC curves in Figure 3.4(c) and (d), the lower the threshold τ is, the higher the P_D and P_F. This new 3D ROC analysis addresses several inherent issues arising in the 2D ROC analysis.

1. First, the original rationale of the ROC analysis is based on signal detection in noise where a fixed threshold τ is calculated to determine whether or not a signal is present. However, when signal detection is extended to signal classification where there are multiple signals to be detected or classified, the signal profile such as amplitude and energy is of major interest and a single fixed threshold τ may not be applicable. In this case, the traditional ROC analysis must be modified for multiple decisions for multisignal detection/classification or soft decisions for signal classification. The new 3D ROC analysis is developed to address this issue

by considering the threshold τ as an independent variable instead of a constant parameter treated in signal detection theory. Varying the threshold τ results in different pairs of P_D and P_F which in turn also generate different ROC curves of (P_D, P_F). Such an advantage cannot be obtained using the 2D ROC analysis as will be demonstrated by the applications in the following section.

2. Despite the fact that an ROC curve of (P_D, P_F) is generated by varying P_F, actually both P_D and P_F are determined by the threshold τ and functions of τ where each pair of (P_D, P_F) is obtained by a single value of threshold τ. Unfortunately, a direct relationship between P_D and P_F via τ is hidden in the 2D ROC curve of (P_D, P_F). As a consequence, it may occur that two different detectors δ^1 and δ^2 may generate the same 2D ROC curves of (P_D, P_F) as shown in



(a) 2D ROC curve of (P_D,P_F)

3D ROC curve

2D ROC curve of (P_D,τ)

2D ROC curve of (P_F,τ)

(b)

3D ROC curve

2D ROC curve of (P_D,τ)

2D ROC curve of (P_F,τ)

(c)

Figure 3.5 Two examples of 3D and 2D ROC curves in (b) and (c) with the same 2D ROC curve of (P_D,P_F) as in (a).

Figure 3.5(a) but their 3D ROC curves along with the other two 2D ROC curves of (P_D,τ) and (P_F,τ) are in fact quite different as shown in Figures 3.5(b), and 3.5(c).

The example illustrated in Figure 3.5 demonstrates that the traditional ROC analysis is ineffective in capturing the impact and effect of different threshold values on P_D and P_F individually. As will also be demonstrated in the experiments, the 2D ROC curves of (P_D,τ) and (P_F,τ) provide certain crucial information about how to choose a best possible threshold τ to compromise P_D and P_F, a task that the 2D ROC curve of (P_D, P_F) cannot really deliver.

3. The traditional 2D ROC analysis generally makes an assumption that the noise is Gaussian so that a closed form for the 2D ROC curve of (P_D, P_F) can be generated analytically in such a way that P_D can be expressed as a function of P_F without actually calculating the threshold τ. However, in many real-world applications the Gaussian assumption is usually not valid. In this case, no analytical form for an ROC curve of (P_D, P_F) can be derived. Accordingly, we need to consider the original detector structure in (3.5) where the threshold τ is the key parameter that determines the detector performance. The 3D ROC curve provides an exit tool from this dilemma to represent detection performance in terms of the three parameters P_D, P_F, and τ where both P_D and P_F can be expressed as functions of τ as shown in Figures 3.4(c) and 3.4(d).

3.5 Real Data-Based ROC Analysis

In real applications only a limited number of samples are available for data analysis, referred to as the power of the test. In this case, the data sample pool is generally not sufficiently large to constitute reliable statistics that can be used to characterize the LRT $\Lambda(\mathbf{r})$ implemented by a detector. Under such a circumstance there is no effective means of producing $\Lambda(\mathbf{r})$ and the ROC analysis must be carried out with data samples rather than statistics, $p_0(\mathbf{r})$ and $p_1(\mathbf{r})$.

3.5.1 How to Generate ROC Curves from Real Data

In what follows, we define

N = total number of data samples used for a particular detection method (technique)

N_{signal} = total number of data samples with presence of a signal (according to ground truth)

$N_{\text{no-signal}}$ = total number of data samples with absence of a signal (according to ground truth)

N_D = total number of data samples with presence of a signal which is actually detected by the method

N_F = total number of data samples with absence of a signal, but claimed to have an signal detected by the method

N_M = total number of data samples with presence of a signal which is not detected by the method

N_{TN} = total number of data samples with presence of a signal and also claimed to have no signal detected by the method.

False alarm or false positive rate/probability is defined by

$$P_F = P_{FP} = \frac{N_F}{N_{\text{no-signal}}} = \frac{N_F}{N_{TN} + N_{FP}}. \tag{3.10}$$

False negative or miss rate/probability:

$$P_{\mathrm{M}} = P_{\mathrm{FN}} = \frac{N_{\mathrm{M}}}{N_{\mathrm{signal}}} = \frac{N_{\mathrm{M}}}{N_{\mathrm{TP}} + N_{\mathrm{FN}}}. \tag{3.11}$$

Detection power/true-positive rate/probability:

$$P_{\mathrm{D}} = P_{\mathrm{TP}} = \frac{N_{\mathrm{D}}}{N_{\mathrm{signal}}} = \frac{N_{\mathrm{D}}}{N_{\mathrm{TP}} + N_{\mathrm{FN}}}. \tag{3.12}$$

True-negative rate/probability:

$$P_{\mathrm{TN}} = \frac{N_{\mathrm{TN}}}{N_{\text{no-signal}}} = \frac{N_{\mathrm{TN}}}{N_{\mathrm{TN}} + N_{\mathrm{FP}}}. \tag{3.13}$$

Based on (3.10)–(3.13), the following relationships are true:

$$N = N_{\mathrm{signal}} + N_{\text{no-signal}}$$

$$N_{\mathrm{signal}} = N_{\mathrm{D}} + N_{\mathrm{M}} = N_{\mathrm{TP}} + N_{\mathrm{FN}} \tag{3.14}$$

$$N_{\text{no-signal}} = N_{\mathrm{TN}} + N_{\mathrm{FP}} \tag{3.15}$$

with

$$P_{\mathrm{D}} = 1 - P_{\mathrm{M}} \quad \text{and} \quad P_{\mathrm{F}} = 1 - P_{\mathrm{TN}}. \tag{3.16}$$

3.5.2 How to Generate Gaussian-Fitted ROC Curves

Until now Equations (3.10)–(3.16) are defined based on real samples. So, for a given set of sample pool used for testing any detection technique, only one point (P_{D}, P_{F}) can be generated for the ROC curve of a particular technique. Therefore, in order to produce a complete ROC curve for any specific detection technique (method), an infinite number of samples pool are required, which is practically impossible. One way to mitigate this difficulty is to assume that the noise in the binary hypothesis decision problem described by (3.1) is a zero-mean Gaussian distribution and the given sample pool is sufficiently large to generate reliable statistics. In this case, we can calculate the sample mean and variance for each hypothesis, and then assume these calculated sample means and variances to be the Gaussian means and variances under each hypothesis. With these new Gaussian distributions, finding the ROC curve of a specific detection technique (method) becomes feasible and can be actually derived mathematically from a standard signal detection theory as follows.

Now if we further assume that the probability density functions $p_0(y)$ and $p_1(y)$ in (3.1) that govern H_0 and H_1 are Gaussian distributions with means μ_0 and μ_1 and variances σ_0^2 and σ_1^2 calculated from a large pool of samples, respectively, the $\Lambda(\mathbf{r})$ in (3.5) becomes $\Lambda(y) = y$. As a result, (3.6) and (3.7) can be further simplified to

$$P_{\mathrm{F}}(\delta^{\mathrm{NP}}) = \int_{\frac{\tau - \mu_0}{\sigma_0}}^{\infty} p_0(y)dy \tag{3.17}$$

$$P_{\mathrm{D}}(\delta^{\mathrm{NP}}) = \int_{\frac{\tau - \mu_1}{\sigma_1}}^{\infty} p_1(y)dy \tag{3.18}$$

where $p_0(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(y-\mu_0)^2}{2\sigma_0^2}}$ and $p_1(y) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(y-\mu_1)^2}{2\sigma_1^2}}$ are two Gaussian distributions with means μ_0 and μ_1 and variances σ_0^2 and σ_1^2, respectively. Furthermore, if both of the variances σ_0^2 and σ_1^2 are set to 1, (3.25) and (3.26) simplify to most familiar forms:

$$P_{\mathrm{F}}(\delta^{\mathrm{NP}}) = \int_{\tau}^{\infty} p_0(y)dy \tag{3.19}$$

$$P_{\mathrm{D}}(\delta^{\mathrm{NP}}) = \int_{\tau}^{\infty} p_1(y)dy. \tag{3.20}$$

Using Figure 3.3 as an example, a decision can be made by adjusting threshold τ via (3.19) and (3.20) where the regions corresponding to P_{D}, P_{F}, P_M, and P_{TN} are indicated with different shaded areas. For example, P_{D} is the area to the right of the threshold τ under the Gaussian distribution $p_1(y)$ when H_1 is true and P_{F} is the area to the right of the threshold τ under the Gaussian distribution $p_0(y)$ when H_0 is true. On the contrary, P_M is the area to the left of the threshold τ under the Gaussian distribution $p_1(y)$ when H_1 is true and P_{TN} is the area to the left of the threshold τ under the Gaussian distribution $p_0(y)$ when H_0 is true. It should be noted that the threshold τ is determined by the false alarm rate. If the false alarm rate is upper bounded by β in (3.4) with Gaussian distributions, using (3.17) we can calculate the corresponding τ by the following:

$$\begin{aligned}\beta = P_{\mathrm{F}}(\delta^{\mathrm{NP}}) = \int_{\frac{\tau - \mu_0}{\sigma_0}}^{\infty} p_0(y)dy \Rightarrow \beta = 1 - \Phi\left(\frac{\tau - \mu_0}{\sigma_0}\right) \\ \Rightarrow \tau = \mu_0 + \sigma_0 \Phi^{-1}(1-\beta)\end{aligned} \tag{3.21}$$

where $\Phi(x)$ is a standard Gaussian distribution given by

$$\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy. \tag{3.22}$$

Therefore, the best decision to find an optimal threshold τ for (3.18) is (3.21) which is determined only by β. We now substitute τ given by (3.21) for τ in (3.18) and obtain the best detection power given by

$$P_{\mathrm{D}}(\delta^{\mathrm{NP}}) = \int_{\frac{\mu_0 - \mu_1 + \sigma_0 \Phi^{-1}(1-\beta)}{\sigma_1}}^{\infty} p_1(y)dy. \tag{3.23}$$

Using (3.21) and (3.23) we can plot a Gaussian-fitted ROC curve of P_{D} versus $P_{\mathrm{F}} = \beta$ for real data.

3.5.3 How to Generate 3D ROC Curves

A major disadvantage resulting from the use of the traditional ROC curve is that both the detector statistics, $\Lambda(\mathbf{r})$, implemented by the Neyman–Pearson detector $\delta^{\text{NP}}(\mathbf{r})$ and the threshold τ are independent parameters and are not specified in the ROC curve of (P_{D}, P_{F}) where P_{D} and P_{F} are actually dependent on $\Lambda(\mathbf{r})$ and τ. Since the value of $\Lambda(\mathbf{r})$ obtained from a data sample $\mathbf{r}$ is generally real valued and represents the detected signal strength present in $\mathbf{r}$, that is, concentration level of a signal in $\mathbf{r}$, a soft decision must be made directly on $\Lambda(\mathbf{r})$ by varying the threshold τ instead of using the parameter β imposed on P_{F}. In doing so, we introduce a normalized detected signal strength of $\Lambda(\mathbf{r})$ as

$$\Lambda_{\text{normalized}}(\mathbf{r}) = \frac{\Lambda(\mathbf{r}) - \min_{\mathbf{r}}\Lambda(\mathbf{r})}{\max_{\mathbf{r}}\Lambda(\mathbf{r}) - \min_{\mathbf{r}}\Lambda(\mathbf{r})}. \tag{3.24}$$

Using τ as a detection threshold value between 0 and 1 for (3.24) we can further define a normalized Neyman–Pearson detector, denoted by $\delta_{\tau}^{\text{NP}}(\mathbf{r})$ based on $\Lambda_{\text{normalized}}(\mathbf{r})$ as follows:

$$\delta_{\tau}^{\text{NP}}(\mathbf{r}) = \begin{cases} 1, & \text{if } \Lambda_{\text{normalized}}(\mathbf{r}) \geq \tau \\ 0, & \text{if } \Lambda_{\text{normalized}}(\mathbf{r}) < \tau, \end{cases} \tag{3.25}$$

which uses τ as a threshold value to convert the normalized real value of $\Lambda_{\text{normalized}}(\mathbf{r})$ to a binary value. Accordingly, a "1" produced by (3.25) indicates that a target is detected; otherwise, there is no target present. By varying $\tau \in [0,1]$ in (3.25), a family of detectors $\left\{\delta_{\tau}^{\text{NP}}(\mathbf{r})\right\}_{\tau\in[0,1]}$ are generated for target detection, where for each τ the detector $\delta_{\tau}^{\text{NP}}(\mathbf{r})$ produces its pair of detection rate and a false alarm rate, (P_{D}, P_{F}). Therefore, if a third dimension is created to specify the threshold τ that is used to define a detector $\delta_{\tau}^{\text{NP}}(\mathbf{r})$ via (3.35), a 3D ROC curve can be generated and plotted based on three parameters, P_{D}, P_{F} and τ. With such a 3D ROC curve, three 2D ROC curves can also be generated, the 2D ROC curve of (P_{D}, P_{F}) which is the traditional ROC curve in Figure 3.1, a 2D ROC curve of (P_{D},τ) and a 2D ROC curve of (P_{F},τ). To generate a 3D ROC curve for real data, three steps are performed:

1. The data samples will be first classified into two categories, falsely alarmed sample pool Ω_{FA} and correctly detected sample pool Ω_{D}. The samples in Ω_{FA} are those samples which are detected as signal samples but actually contain no signals according to the ground truth. The detected sample pool Ω_{D} are those samples that are correctly detected by the normalized NP detector $\delta_{\tau}^{\text{NP}}(\mathbf{r})$ according to the ground truth. The sample set Ω_{signal} denotes the set of samples which actually have signal strength/concentration present in the $\mathbf{r}$ according to the ground truth.
2. Let Ω denote the total sample pool used for evaluation and Ω_{S} be the set of samples with signal presence, that is, samples with correctly detected and falsely missed signal samples (see (3.14)), and Ω_{NS} be the set of samples with signal absence, that is, samples with no signal detected and falsely detected signal samples (see (3.15)). In addition, let Ω_{SD} be the set of samples with detected signal strength/concentration greater than zero and Ω_{NSD} be the set of samples with no signal detected, that is, signal with zero strength/concentration. Then $\Omega = \Omega_{\text{S}} \cup \Omega_{\text{NS}} = \Omega_{\text{SD}} \cup \Omega_{\text{NSD}}$ with $\Omega_{\text{S}} \cap \Omega_{\text{NS}} = \varnothing$ and $\Omega_{\text{SD}} \cap \Omega_{\text{NSD}} = \varnothing$. The threshold τ in (3.21) is used to generate probabilities of falsely alarmed sample pool Ω_{FA} and signal detected sample pool Ω_{SD}.

3. For each threshold τ calculated in step 2, a pair of probabilities P_{F} and P_{D} are defined as follows, where *N(A)* denotes the number of samples in a sample pool A:

$$P_{\mathrm{D}} = \frac{N(\Omega_{\mathrm{D}})}{N(\Omega_{\mathrm{S}})} = \frac{N(\Omega_{\mathrm{S}} \cap \Omega_{\mathrm{SD}})}{N(\Omega_{\mathrm{S}})} \tag{3.26}$$

$$P_{\mathrm{M}} = \frac{N(\Omega_{\mathrm{M}})}{N(\Omega_{\mathrm{S}})} = \frac{N(\Omega_{\mathrm{S}} \cap \Omega_{\mathrm{NSD}})}{N(\Omega_{\mathrm{S}})} = 1 - P_{\mathrm{D}} \tag{3.27}$$

$$P_{\mathrm{F}} = \frac{N(\Omega_{\mathrm{FA}})}{N(\Omega_{\mathrm{NS}})} = \frac{N(\Omega_{\mathrm{NS}} \cap \Omega_{\mathrm{SD}})}{N(\Omega_{\mathrm{NS}})} \tag{3.28}$$

where $\Omega_{\mathrm{S}} \cap \Omega_{\mathrm{SD}} = \Omega_{\mathrm{D}}$, $\Omega_S \cap \Omega_{\mathrm{NSD}} = \Omega_M$, $\Omega_{\mathrm{NS}} \cap \Omega_{\mathrm{SD}} = \Omega_{\mathrm{FA}}$, and $\Omega_{\mathrm{NS}} \cap \Omega_{\mathrm{NSD}} = \Omega_{\mathrm{ND}}$ with $|\Omega_{\mathrm{D}}| = \mathrm{N}_{\mathrm{TP}}$, $|\Omega_M| = N_{\mathrm{M}}$, $|\Omega_{\mathrm{FA}}| = N_{\mathrm{FP}}$, and $|\Omega_{\mathrm{ND}}| = N_{\mathrm{TN}}$. (See the definitions given in Section 3.5.1 and |X|= number of samples in X.)

Figure 3.6 shows a diagram of relationships among Ω_{S}, Ω_{SD}, Ω_{M}, Ω_{FA}, Ω_{NS}, and Ω_{NSD}.

In analogy with Section 3.5.2, we can also generate Gaussian fitted 3D ROC curves by the following steps:

1. Calculate sample means and variances for Ω_{S} and Ω_{NS}, denoted by μ_{S}, σ_{S}^2, and μ_{NS}, σ_{NS}^2, respectively.
2. Find the Gaussian probability distributions under hypotheses H_0 and H_1, that is, $p_1(y) = N(\mu_{\mathrm{NS}}, \sigma_{\mathrm{NS}}^2)$ for H_0 and $p_1(y) = N(\mu_{\mathrm{S}}, \sigma_{\mathrm{S}}^2)$ for H_1.
3. Calculate the pair of probabilities P_{F} and P_{D} according to the following formulas:

$$P_{\mathrm{D}} = \int_{\varepsilon}^{+\infty} p_1(y)\,dy = \int_{\varepsilon}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{S}}} \left\{ -\frac{(y-\mu_{\mathrm{S}})^2}{2\sigma_{\mathrm{S}}^2} \right\} dy \tag{3.29}$$

$$P_{\mathrm{F}} = \int_{\varepsilon}^{+\infty} p_0(y)dy = \int_{\varepsilon}^{+\infty} \frac{1}{\sqrt{2\pi}\sigma_{\mathrm{NS}}} \left\{ -\frac{(y-\mu_{\mathrm{NS}})^2}{2\sigma_{\mathrm{NS}}^2} \right\} dy. \tag{3.30}$$

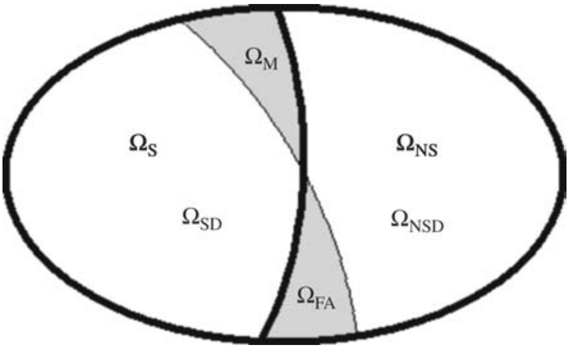


Figure 3.6 A diagram of relationships among Ω_{S}, Ω_{SD}, Ω_{M}, Ω_{F}, Ω_{NS}, and Ω_{NSD}.

It should be noted that the means and variances in (3.29) and (3.30) can be calculated from a given sample pool. For example, using the notations, $N_{\text{no-signal}}$ and N defined in Section 3.5.1, we can calculate the μ_0, μ_1, σ_0^2 and σ_1^2 for (3.29) and (3.30) as follows:

$$\mu_0 = \frac{1}{N_{\text{no}-\text{signal}}} \sum\nolimits_{i \in \text{no-signal}} f(y_i) \tag{3.31}$$

$$\sigma_0^2 = \frac{1}{N_{\text{no-signal}}} \sum\nolimits_{i \in \text{no-signal}} (f(y_i) - \mu_0)^2 \tag{3.32}$$

$$\mu_1 = \frac{1}{N} \sum\nolimits_i f(y_i) \tag{3.33}$$

$$\sigma_1^2 = \frac{1}{N} \sum\nolimits_i (f(y_i) - \mu_1)^2 \tag{3.34}$$

where $f(y_i)$ is the value of the ith sample y_i.

3.5.4 How to Generate 3D ROC Curves for Multiple Signal Detection and Classification

The hypothesis testing problem (3.1) considered so far assumes the standard signal detection in noise (SN) model where hypotheses H_0 and H_1 represent noise and signal + noise, respectively. There have been studies on extending (3.1) to two scenarios. One is called the signal/background/noise model proposed in Thai and Healey (2002) which includes background **B** as a third signal source described by

$$\begin{aligned} &H_0: \quad Y \approx \mathbf{B} + \mathbf{N} \\ &\text{versus} \\ &H_1: \quad Y \approx \mathbf{S} + \mathbf{B} + \mathbf{N}. \end{aligned} \tag{3.35}$$

A second scenario is called the signal-decomposed interference/noise (SDIN) model suggested in Du and Chang (2004b) and is given by

$$\begin{aligned} &H_0: \quad Y \approx \mathbf{U} + \mathbf{I} + \mathbf{N} \\ &\text{versus} \\ &H_1: \quad Y \approx \mathbf{D} + \mathbf{U} + \mathbf{I} + \mathbf{N} \end{aligned} \tag{3.36}$$

where the signal source **S** considered in the SBN model (3.35) is further decomposed into the desired signal source **D** and the undesired signal source **U** and the background **B** considered in the SBN model is included in the interference signal source matrix **I**.

Using the SDIN model specified by (3.36), we can interpret various commonly used models as follows. When $\mathbf{U} = \mathbf{I} = \varnothing$ and $\mathbf{D} = \mathbf{S}$, the SDIN model is reduced to the standard SN model. If $\mathbf{D} = \mathbf{S}$ and $\mathbf{I} = \mathbf{B}$ with $\mathbf{U} = \varnothing$, then the SDIN model becomes the SBN model. The SDIN allows us to deal with multisignal detection and classification by interpret $\mathbf{I} = \varnothing$ and signal sources as a signal source matrix $\mathbf{S} = [\mathbf{d}\,\mathbf{U}]$ comprising multiple signal sources with $\mathbf{D} = \mathbf{d}$ representing the target signal source of interest to be detected and **U** being other target signal sources with no interest to **d**.

In order to extend a single target signal detection-based ROC analysis to a multiple-signal detection model specified by (3.36), we assume that there are p signal sources of interest, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$. Then the detection rate $R_{\text{D}}(\mathbf{m}_j)$ and false alarm rate $R_{\text{F}}(\mathbf{m}_j)$ for the jth signal source $\mathbf{m}_j$ defined by

$$R_{\text{D}}(\mathbf{m}_j) = \frac{N_{\text{D}}(\mathbf{m}_j)}{N(\mathbf{m}_j)} \tag{3.37}$$

$$R_{\text{F}}(\mathbf{m}_j) = \frac{N_{\text{F}}(\mathbf{m}_j)}{N - N(\mathbf{m}_j)} \tag{3.38}$$

where $N_D(\mathbf{m}_j)$ is the total number of true pixels which are $\mathbf{m}_j$ and detected as $\mathbf{m}_j$, $N_F(\mathbf{m}_j)$ is the total number of true pixels which are not $\mathbf{m}_j$ but detected as $\mathbf{m}_j$, $N(\mathbf{m}_j)$ is the total number of pixels that are specified by target signature $\mathbf{m}_j$ and N is the total number of pixels in the image. For detection of multiple signal sources, the detection rate/power P_{D} and false alarm rate P_{F} are then replaced by the mean detection rate $\bar{R}_{\text{D}}$ and mean false alarm rate $\bar{R}_{\text{F}}$, respectively, which can be defined by taking the mean of $R_D(\mathbf{m}_j)$ and the mean of $R_F(\mathbf{m}_j)$ over all the p $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ as

$$\bar{R}_{\text{D}} = \sum\nolimits_{j=1}^{p} p(\mathbf{m}_j) R_D(\mathbf{m}_j) \tag{3.39}$$

$$\bar{R}_{\text{F}} = \sum\nolimits_{j=1}^{p} p(\mathbf{m}_j) R_F(\mathbf{m}_j) \tag{3.40}$$

where $p(\mathbf{m}_j) = \frac{N(\mathbf{m}_j)}{N(\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p)}$, $N(\mathbf{m}_j)$ is the total number of pixels, which are, $\mathbf{m}_j$ and $N(\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p)$ is the total number of all target pixels given by $N(\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p) = \sum_{j=1}^{p} N(\mathbf{m}_j)$.

It notes that through $\delta_{\tau}^{\text{NP}}(\mathbf{r})$ specified by (3.25) along with (3.39) and (3.40) each of multiple signals, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ will be detected and classified *jointly* by a *fixed* and *same* threshold τ for all the p signal sources $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ to produce a point in a 2D space given by $(\bar{\text{R}}_{\text{D}}, \bar{\text{R}}_{\text{F}})$. This is different from the single signal detection of $\mathbf{m}_j$ which uses its own and separate individual threshold τ_i in (3.25) to produce its own pair (P_{D}, P_{F}). However, such a subtle and crucial difference cannot be seen from the traditional 2D ROC curve of (P_{D}, P_{F}) since the threshold τ is hidden in P_{D} and P_{F} and the curve cannot show its influence on both P_{D} and P_{F}.

By decreasing τ from 1 to 0 in a third dimension, it results in a 3D mean-ROC curve, which can be used to evaluate the performance of a detector where the (x,y) coordinate is specified by ($\bar{\text{R}}_{\text{D}}$,$\tau$) and the z-axis is specified by $\bar{\text{R}}_{\text{D}}$. Using this 3D mean-ROC curve we can further plot three 2D curves, a curve of $\bar{R}_{\text{D}}$ versus $\bar{\text{R}}_{\text{F}}$ which is the traditional ROC curve, a curve of $\bar{R}_{\text{D}}$ versus τ and a curve of $\bar{\text{R}}_{\text{F}}$ versus τ for detection performance analysis. Once the 2D ROC curve of ($\bar{\text{R}}_{\text{D}}$,$\bar{\text{R}}_{\text{F}}$) is generated, the area under the curve is calculated and defined as detection rate, which can be used to evaluate the effectiveness of a detector. The higher the detection rate the better the detector.

3.6 Examples

One immediate application of 3D ROC analysis is hyperspectral imaging where the strength/concentration of a detected signal is specified by the abundance fraction of a particular target which is actually a real value and plays the central role in data analysis. By varying the value of threshold τ in (3.25) the resulting different abundance fractions can be generated and different performances can be produced.

3.6.1 Hyperspextral Imaging

Two examples, hyperspectral target detection and linear hyperspectral mixture analysis, are presented in this section for illustration.

3.6.1.1 Hyperspectral Target Detection

In this section, the CEM specified by (2.33) in Section 2.2.3 of Chapter 2 will be used for target detection for the HYDICE scene in Figure 1.15 and panel signatures in Figure 1.16 will be used for desired target signature, **t**. Since we are only interested in panel detection, pixels other than panel pixels will be considered as background pixels. In this case, there are five panel classes identified as target classes, of which one class for each of five panel signatures and one background class which accounts for all nontarget pixels. Figure 3.7 shows the detection results for all the five panel signatures with each of the five panel signatures in Figure 1.16 used as the desired target signature **t** for detection. As shown in Figure 3.7, all the panels in five rows are detected very well and effectively.

The results in Figure 3.7 provide only qualitative assessment by visual inspection and not quantitative analysis. Since the detector specified by (2.33) actually estimates the abundance fraction of the desired target signal **t** rather than detects the target signal **t** itself, we can tabulate the detection results according to the ground truth in Figure 1.13(b) and use 3D ROC analysis for performance evaluation. By virtue of (3.24) we can derive a normalized CEM by

$$\delta^{\text{CEM}}_{\text{normalized}}(\mathbf{r}) = \frac{\delta^{\text{CEM}}(\mathbf{r}) - \min_{\mathbf{r}}\delta^{\text{CEM}}(\mathbf{r})}{\max_{\mathbf{r}}\delta^{\text{CEM}}(\mathbf{r}) - \min_{\mathbf{r}}\delta^{\text{CEM}}(\mathbf{r})}. \tag{3.41}$$

Analogous to (3.25) we can also use (3.41) to define a hard-decision making detector as

$$\delta^{\text{CEM}}_{\tau}(\mathbf{r}) = \begin{cases} 1, \text{ if } \delta^{\text{CEM}}_{\text{normalized}}(\mathbf{r}) \geq \tau \\ 0, \text{ if } \delta^{\text{CEM}}_{\text{normalized}}(\mathbf{r}) < \tau \end{cases}. \tag{3.42}$$

By varying the threshold $\tau \in [0, 1]$ we can produce a 3D ROC curve plotted in Figure 3.8(a) along with their corresponding three 2D ROC curves in Figures 3.8(b) and 3.8(d) detecting of each of the five panel signatures.

As shown in Figure 3.8(b), the 2D ROC curve of (P_{D}, P_{F}) is nearly close to the one which indicated that CEM performed extremely well in detecting 15 panel pixels. As also shown in Figure 3.8(c), the 2D ROC curve of (P_{D},τ) implies that the CEM detected abundance fractions to reflect why detection rate is so high. On the other hand, the 2D ROC curve of (P_{F},τ) in Figure 3.8(d) shows that P_{F} is very high when τ is set to very small values less than 0.15, but P_{F}

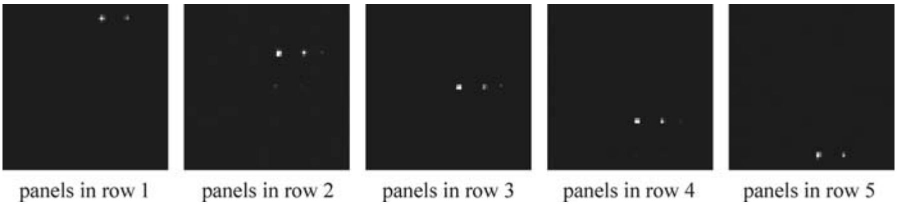
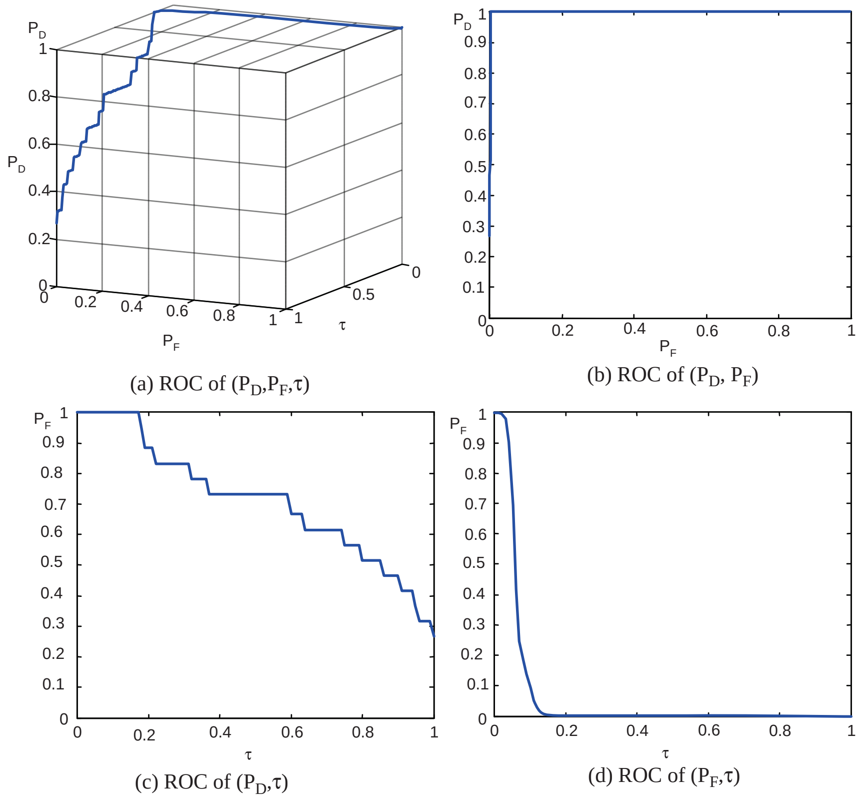

panels in row 1 panels in row 2 panels in row 3 panels in row 4 panels in row 5

Figure 3.7 Detection of 15 panels by CEM.



(a) ROC of (P_D, P_F, τ)

(b) ROC of (P_D, P_F)

(c) ROC of (P_D, τ)

(d) ROC of (P_F, τ)

Figure 3.8 3D ROC curves of five-panel signature detection.

dropped down to nearly zero when τ is greater than 1.3. Based on the 2D ROC curves in Figures 3.8(c) and 3.8(d), the best τ would be around 0.2. Such information is not provided by the 2D ROC curve of (P_D, P_F) in Figure 3.8(b). Table 3.1 also tabulates their AUCs, A_z, for 2D ROC curve of (P_D, P_F).

3.6.1.2 Linear Hyperspectral Mixture Analysis

Linear hyperspectral mixture analysis is one of fundamental data processing techniques widely used in hyperspectral imaging. One of known and popular LSMA techniques is the orthogonal subspace projection (OSP) developed in Section 2.3.2.1. It is originally developed by Harsanyi and Chang (1994) to estimate abundance fractions of each of image endmembers assumed to be present in the data to perform mixed pixel classification. Since OSP is an abundance-unconstrained estimator it may not accurately estimate the true abundance fractions of each of image endmembers used to form the linear mixture model (2.75), two abundance-constrained least-squares-based LSU techniques are further used for comparison. One is a partially abundance-constrained least-squares LSMA, called the non-negativity abundance-constrained least-squares (NCLS) method

Table 3.1 A_z calculated panels detected in five rows in Figure 3.8(b)

	Panels in row 1	Panels in row 2	Panels in row 3	Panels in row 4	Panels in row 5
A_z	0.99919	0.99963	0.99969	0.99969	0.9996

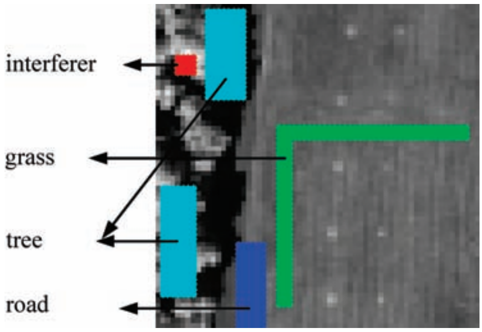


Figure 3.9 Finding background and interferer signatures.

(Chang 2003a, Chapter 3), and fully abundance-constrained least-squares (FCLS) methods (Chang 2003a, Chapter 10) are also used for experiments.

Obviously, from the scene in Figure 1.13(a) there are more than 15 panels. Because there is no ground truth of signatures provided other than 15 panels in the scene, we use the virtual dimensionality (VD) in Chapter 5 originally developed in Chang (2003a, Chapter 17) to estimate a total number of spectrally distinct signatures assumed to be present in this scene, which is 9. Based on this information and visual inspection, we identify five areas marked in Figure 3.9 to produce four other signatures, grass, tree road plus an interferer in addition to five panel signatures in Figure 1.14.

In order to effectively perform the LSMSA, the SDIN model specified by (3.36) was used as the signature matrix **M** with the desired panel signatures denoted by $\mathbf{S} = [\mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3\ \mathbf{p}_4\ \mathbf{p}_5]$ designated as the signal matrix and **I** = [**interferer grass tree road**] as the interference matrix. In this particular example, we can formed a background signature matrix **B** made up of **grass**, **tree**, **road** which were considered as part of the interference matrix **I**. With this interpretation the SBN in (3.35) became a special case of the SDIN. It should be noted that since the **interferer** on the left upper corner shows strong signal presence by its gray intensity, it is chosen as an unwanted signal source.

The linear spectral unmixing was then performed to unmix 15 panel pixels using $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, $\mathbf{p}_5$ for mixed pixel classification where $\mathbf{d}$ = one specific panel signature to be used for classification and **U** made up of all the other four panel signatures. Three mixed pixel classifiers LSOSP (Tu et al., 1997), NCLS, and FCLS that were then used to unmix 15 panel pixels in Figure 1.13 where two orthogonal subspace projectors, $P^{\perp}_{[\mathbf{U}\ \mathbf{B}]}$ and $P^{\perp}_{[\mathbf{U}\ \mathbf{I}]}$, defined by (2.78) were used for the SBN and SDIN models, respectively, to reject all unwanted signatures prior to detection of **d** by a matched filter. It should be noted that LSOSP was used instead of OSP because LSOSP provides better abundance fraction estimation than OSP does as shown in Tu et al. (1997) and Chang et al. (1998). Figures 3.10 and 3.11 show classification of the 15 panels with the 19 panel pixels unmixed by LSOSP, NCLS and FCLS using SBN and SDIN models, respectively, where as expected, the 15-panel (19 panel pixels) classification results using the SDIN model performed better than that produced by using the SBN model.

Since the unmixed results in Figures 3.10 and 3.11 are abundance fraction maps, they cannot be directly used for hard-decision-based classification to perform 3D ROC analysis. In this case, we followed the same treatment as we derived (3.41) and (3.42) for CEM by defining a normalized LSMA as

$$\delta^{\mathrm{LSMA}}_{\mathrm{normalized}}(\mathbf{r}) = \frac{\delta^{\mathrm{LSMA}}(\mathbf{r}) - \min_{\mathbf{r}}\delta^{\mathrm{LSMA}}(\mathbf{r})}{\max_{\mathbf{r}}\delta^{\mathrm{LSMA}}(\mathbf{r}) - \min_{\mathbf{r}}\delta^{\mathrm{LSMA}}(\mathbf{r})}. \tag{3.43}$$

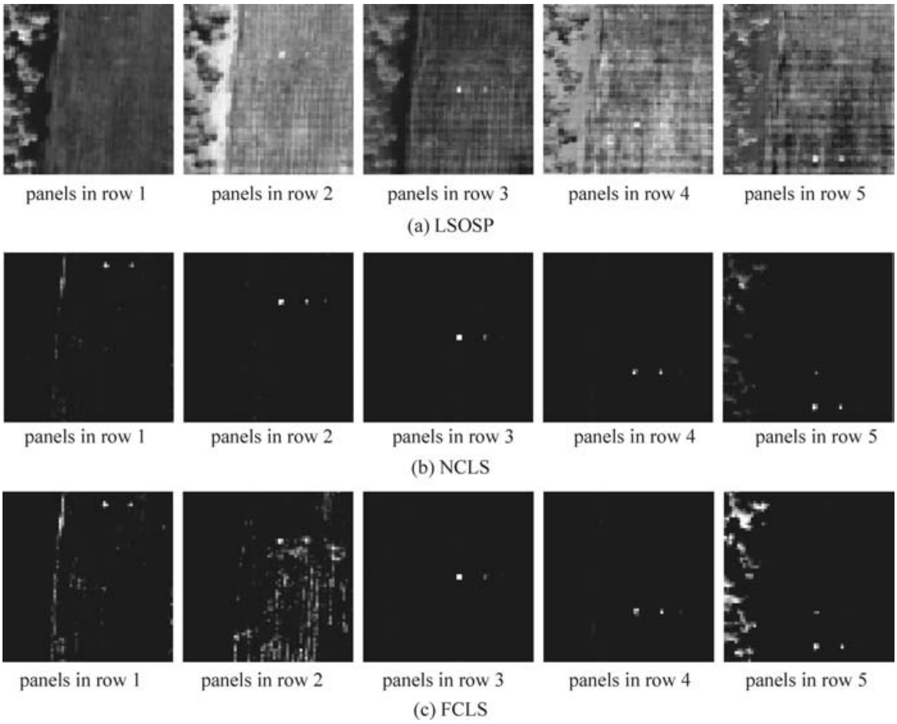


Figure 3.10 Classification of 19 panel pixels produced by LSOSP, NCLS, and FCLS using the SBN model.

and a hard-decision-making detector as

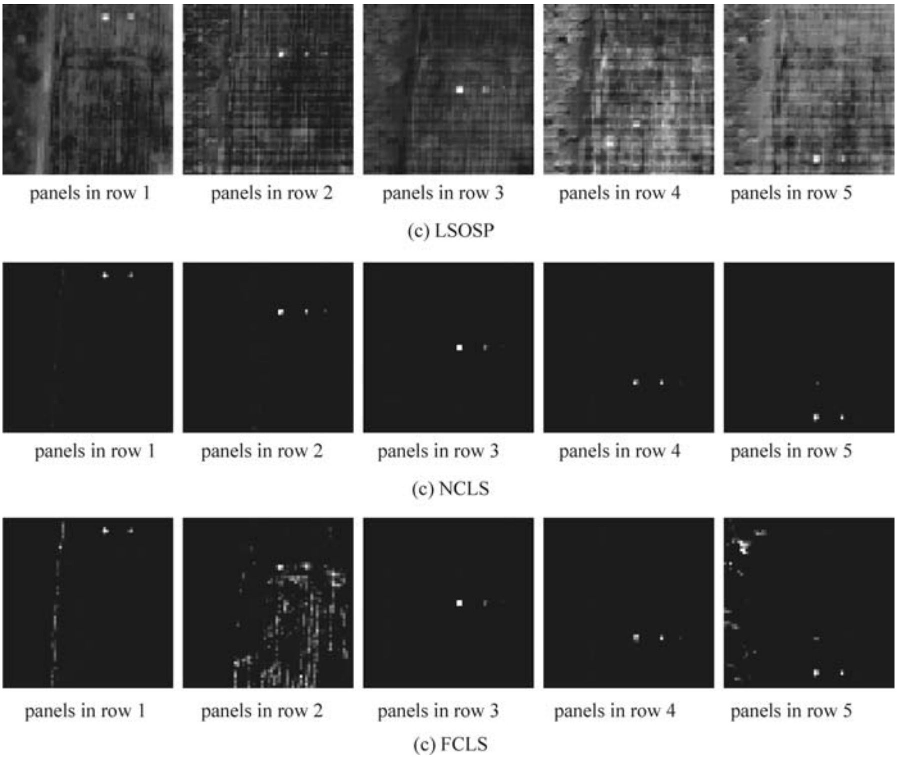
$$\delta_{\tau}^{\text{LSMA}}(\mathbf{r}) = \begin{cases} 1, \text{if} & \delta_{\text{normalized}}^{\text{LSMA}}(\mathbf{r}) \geq \tau \\ 0, \text{if} & \delta_{\text{normalized}}^{\text{LSMA}}(\mathbf{r}) < \tau \end{cases} \tag{3.44}$$

where the superscript "LSMA" in (3.43) and (3.44) is used to indicate any specific LSMA technique that will be converted to hard decisions. In this case, hard-decision-making LSOSP, NCLS, and FCLS detectors can be defined via (3.43) and (3.44).

By varying τ in the range of [0,1] in (3.43) and (3.44) we produced 3D ROC curves of ($\bar{R}_{\text{D}}$,$\bar{R}_{\text{F}}$,τ) for performance evaluation. Figures 3.12(a) and 3.13(a) show 3D ROC curves of 15-panel-pixel detection results in Figures 3.10 and 3.11 using the SBN and SDIN, respectively, where the same threshold τ was used in (3.44). Along with the 3D ROC curves their corresponding 2D ROC curves are also plotted in Figures 3.12(b)–(d) and 3.13(b)–(d) for comparison and their AUCs for the 2D ROC curve of (P_{D}, P_{F}), A_z are also tabulated in Table 3.2.

From the results in Figures 3.13, 3.14, and Table 3.2 NCLS clearly outperformed LSOSP and FCLS for both SBN and SDIN models in terms of 15-panel (19-panel pixels) mean detection.

To further make a comparison between joint detection of multiple signals, $\mathbf{S} = [\mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3\ \mathbf{p}_4\ \mathbf{p}_5]$, using the same threshold $\tau_{\mathbf{S}}$ and single-signal detection of five individual panel signatures, $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, $\mathbf{p}_5$, using five separate and independent thresholds τ_1, τ_2, τ_3, τ_4, τ_5, tabulated in Table 3.3, their AUCs, A_z, were calculated where it can be clearly seen that independent and separate single signal detection of multiple signals always performed better than joint detection of multiple signals. This is because the latter used different independent thresholds, τ_1, τ_2, τ_3, τ_4, τ_5, to detect



(c) LSOSP

(c) NCLS

(c) FCLS

Figure 3.11 Classification of 19 panel pixels produced by LSOSP, NCLS, and FCLS using the SDIN model.

five distinct panel signatures, while the former must use the same threshold τ for all the five panel signatures.

As also shown in Figures 3.12(c)–(d) and 3.13(c)–(d), it suggested that an effective detector such as NCLS should have smaller areas under the 2D ROC curves of ($\bar{R}_D$,τ) and ($\bar{R}_F$,τ). In particular, these figures also provide information about what a good threshold to be compromised between P_{D} and P_{F} is where τ is around 0.18 for the SBN model and 0.1 for the SDIN model. Additionally, Figures 3.12(d) and 3.13(d) also show that how effective NCLS was in terms of P_{F} versus the threshold τ where such information could not be provided by the traditional 2D ROC curve of (P_{D}, P_{F}).

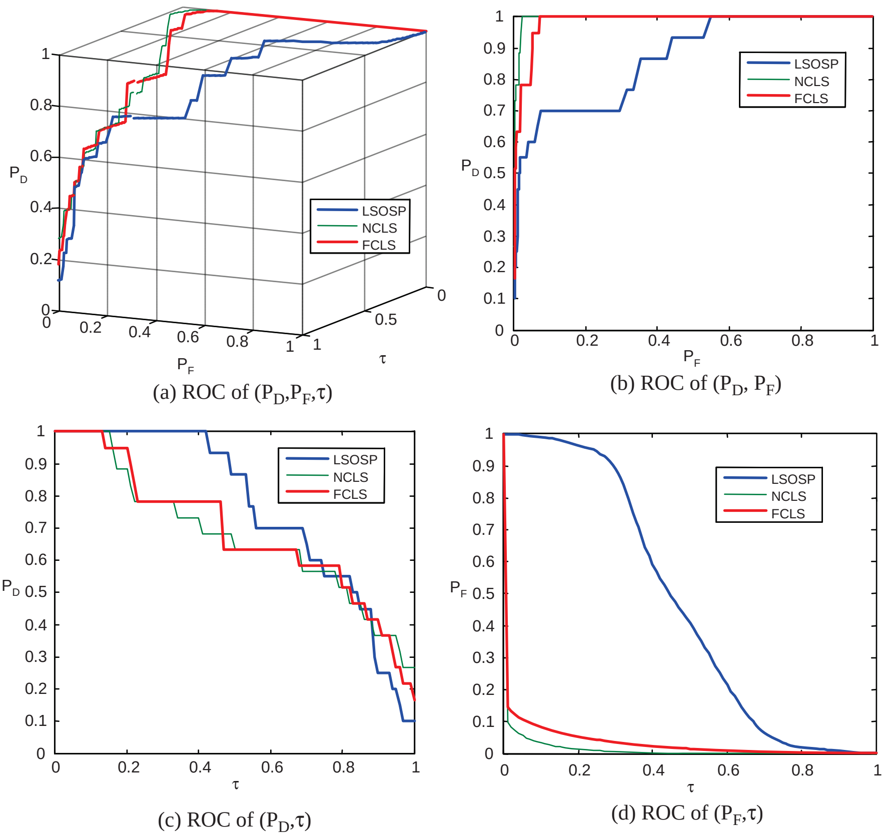
3.6.2 Magnetic Resonance (MR) Breast Imaging

This section presents another application of 3D ROC analysis in real breast MR image experiments. We define two specific terminologies commonly used in medical imaging community, sensitivity

$$\text{sensitivity} = P_{\text{D}} = \frac{N_{\text{TP}}}{N_{\text{TP}} + N_{\text{FN}}} \tag{3.45}$$

which is exactly the same as the detection rate in detection theory, and specificity

$$\text{specificity} = P_{\text{TN}} = \frac{N_{\text{TN}}}{N_{\text{TN}} + N_{\text{FP}}} \tag{3.46}$$



(a) ROC of (P_D,P_F,τ)

(b) ROC of (P_D, P_F)

(c) ROC of (P_D,τ)

(d) ROC of (P_F,τ)

Figure 3.12 3D and 2D ROC curves of LSOSP, NCLS, and FCLS for the SBN model.

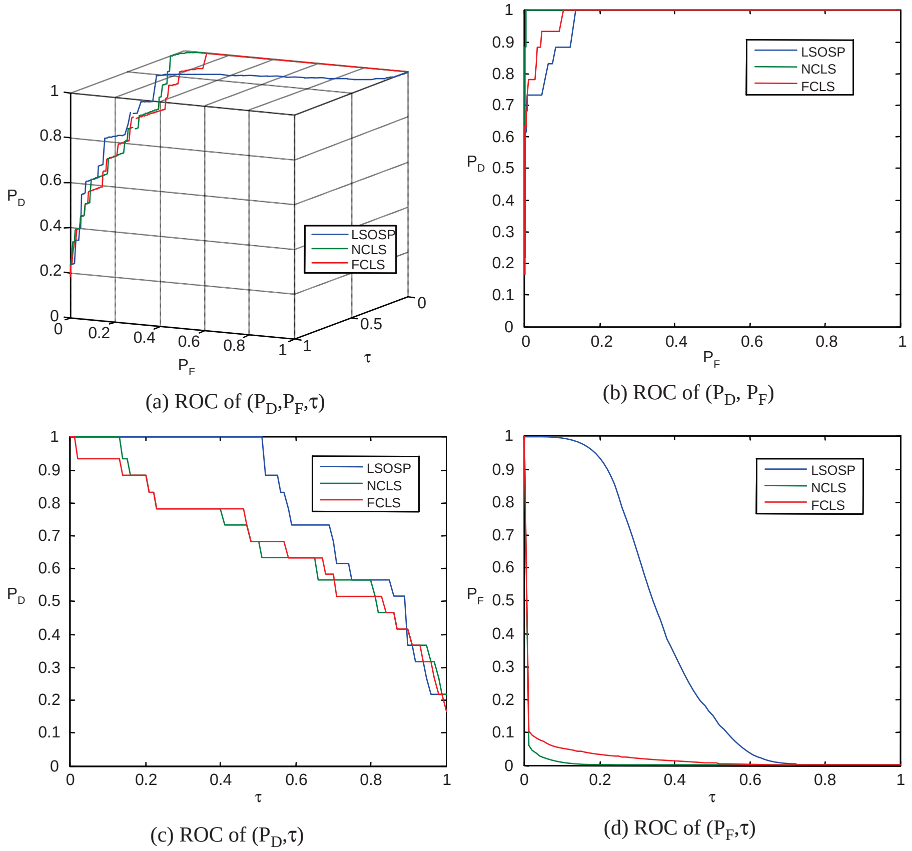
which does not exist in detection theory because it is the case of noise detection where detecting noise does not make sense in detection theory. However, it does make sense in medical imaging because it represents "negative" in diagnosis which implies a normal case.

3.6.2.1 Breast Tumor Detection

Four multispectral MR breast images shown in Figure 3.14 were used for the experiments. Each image has the same size of 427 × 427 pixel vectors with five breast tissues of interest, fatty, glandular, muscle, tumor mass, and vessel, specified in Figure 3.15.

Table 3.2 A_z calculated for LSOSP, NCLS, and FCLS for SBN and SDIN models in Figures 3.12(b) and 3.13(b)

	LSOSP	NCLS	FCLS
SBN	0.865	0.996	0.987
SDIN	0.974	0.999	0.966



(a) ROC of (P_D, P_F, τ)

(b) ROC of (P_D, P_F)

(c) ROC of (P_D, τ)

(d) ROC of (P_F, τ)

Figure 3.13 3D and 2D ROC curves of LSOSP, NCLS, and FCLS for the SDIN model.

Specifically, band 1 is the flash 2D spectral image acquired by TR/TE = 352 ms/4.7 ms; band 2 is a T1-weighted 2D spectral image acquired by TR/TE = 832 ms/20 ms; band 3 is a T2-weighted 2D spectral image acquired by TR/TE = 3000 ms/105 ms; and band 4 is photon-density (PD)-weighted 2D spectral image acquired by TR/TE = 3000 ms/15 ms. The slice thickness is 3 mm and sagittal sections are taken by the Siemens 1.5-T system. By stacking these four MR breast images as an image cube, the CEM specified by (2.33) was used to detect these five breast tissues. Since three signatures from Figure 3.15, fatty, tumor, and muscle, are of our major interest, these tissues

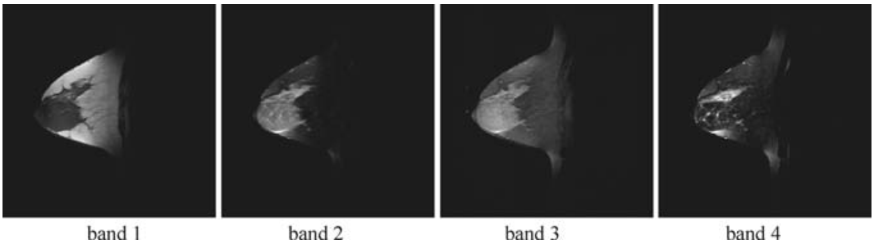


Figure 3.14 Four breast MR images.

Table 3.3 A_z calculated for 19-panel pixels in five rows

	Panels in row 1	Panels in row 2	Panels in row 3	Panels in row 4	Panels in row 5
SBN LSOSP	0.93221	0.92984	0.99426	0.96103	0.95261
NCLS	0.98971	0.99976	0.99951	0.99948	0.98596
FCLS	0.98604	0.99161	0.99976	0.99905	0.97476
SDIN LSOSP	0.97432	0.99149	0.99976	0.99939	0.99289
NCLS	0.99475	0.99884	0.99963	0.96451	0.95401
FCLS	0.99646	0.99976	0.99969	0.9996	0.99945

were then used as the desired tissues by CEM for target detection. The detection results for each of these three breast tissues (fatty, glandular, and tumor mass) are shown in Figure 3.16, where the breast tumor mass is clearly extracted in Figure 3.16(a).

Since the detection images in Figure 3.16 are actually abundance fraction maps, the detection rate for each of the three tissues must be determined by the threshold τ specified in (3.42). As a result, a 3D ROC curve along with three 2D ROC curves were generated for fatty, tumor, and glandular, respectively. Because the tumor mass is of primary interest, only its 3D ROC curves along with three 2D ROC curves are plotted in Figure 3.17 with $A_z = 0.8764$ where the P_D and P_F are TP and FP, respectively.

To further illustrate the need for 3D ROC analysis, Figure 3.18 shows various results of detecting each of the three tissues in Figure 3.16 using four threshold values, $\tau = 0.2, 0.4, 0.6, 0.8$.

As shown in Figure 3.18, the detection performance varied and was largely determined by the threshold τ, which in turn determined the pair of 2D ROC curves of (TP,FP) in Figure 3.17(b)–(d).

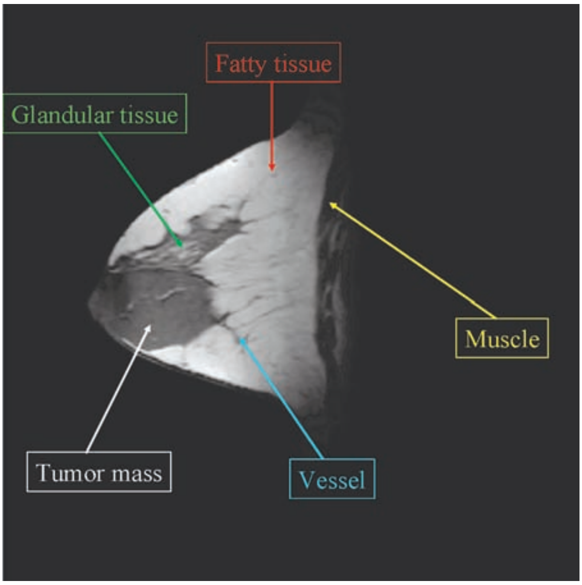


Figure 3.15 Five breast tissues of interest.

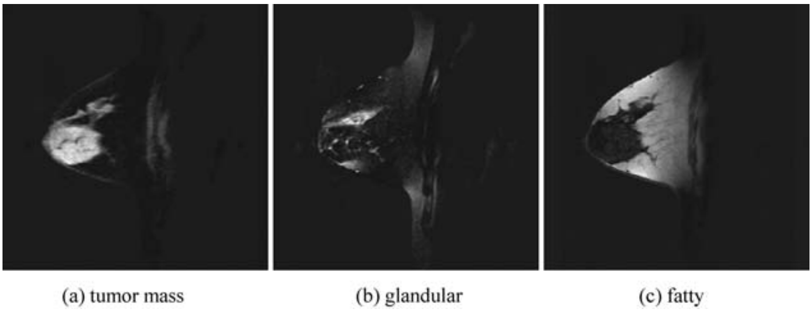
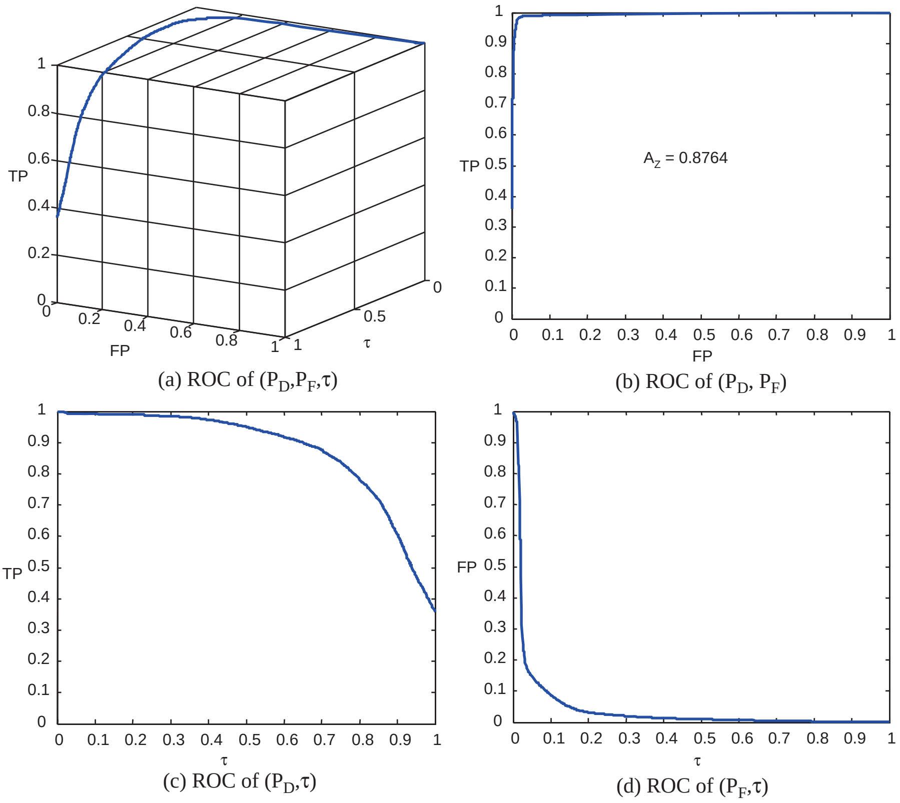

(a) tumor mass (b) glandular (c) fatty

Figure 3.16 Detection results of three breast tissues by CEM.

Under such a circumstance, only the 3D ROC curve could provide adequate analysis for medical diagnosis.

3.6.2.2 Brain Tissue Classification

The synthetic images to be used for experiments in this section were the axial T1, T2, and proton density MR brain images (with 5-mm section thickness, 0% noise, and 0% intensity



(a) ROC of (P_D, P_F, τ) (b) ROC of (P_D, P_F)

(c) ROC of (P_D, τ) (d) ROC of (P_F, τ)

Figure 3.17 3D ROC curve and three 2D ROC curves for tumor mass.

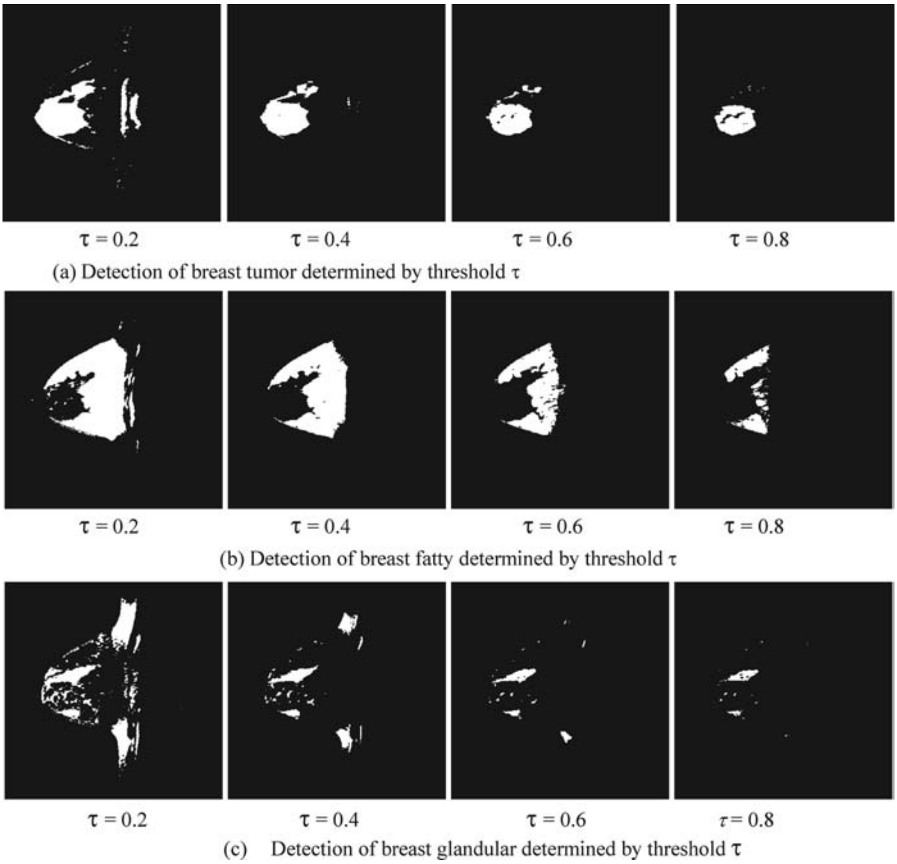

τ = 0.2
τ = 0.4
τ = 0.6
τ = 0.8
(a) Detection of breast tumor determined by threshold τ
τ = 0.2
τ = 0.4
τ = 0.6
τ = 0.8
(b) Detection of breast fatty determined by threshold τ
τ = 0.2
τ = 0.4
τ = 0.6
τ = 0.8
(c) Detection of breast glandular determined by threshold τ

Figure 3.18 Detection results for each of the three tissues determined by the threshold τ specified by (53).

nonuniformity) resulting from the MR imaging simulator of McGill University, Montreal, Canada (available at _www.bic.mni.mcgill.ca/brainweb/). The image volume provided separate volumes of tissue classes, such as CSF, GM, WM, bone, fat, and background. The use of these web MR brain images allows researchers to reproduce our experiments for verification. Figure 3.19 shows three

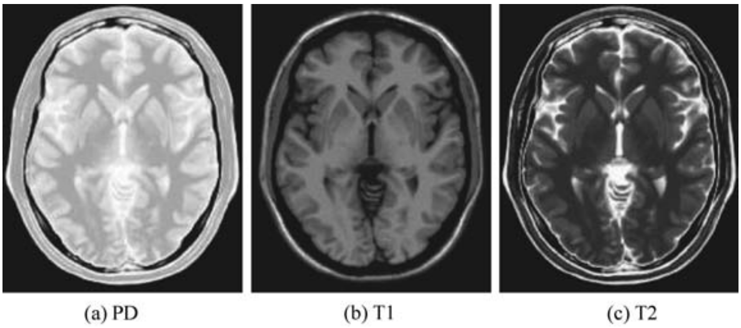


Figure 3.19 Three MR brain images.

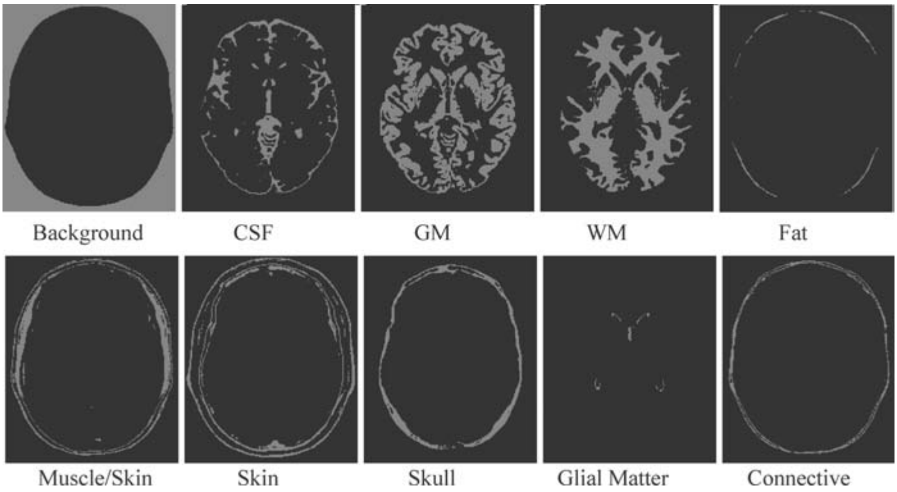


Figure 3.20 Ground truth of brain tissue substances for images in Figure 3.19.

MR brain images with specifications provided in www.bic.mni.mcgill.ca/brainweb/ where the first image is acquired by Modality = PD, Protocol = ICBM, Phantom_name = normal, Slice_thickness = 5 mm, Noise = 0%, INU (intensity nonuniformity) = 0%, the second image by Modality = T1, Protocol = ICBM, Phantom_name = normal, Slice_thickness = 5 mm, Noise = 0%, INU = 0% and the third image by Modality = T2, Protocol = ICBM, Phantom_name = normal, Slice_thickness = 5 mm, Noise = 0%, INU = 0%.

Figure 3.20 provides the ground truth, which is also available in the website www.bic.mni.mcgill.ca/brainweb/ for brain tissue substances in the images in Figure 3.19 that will be used to verify the results obtained for our experiments.

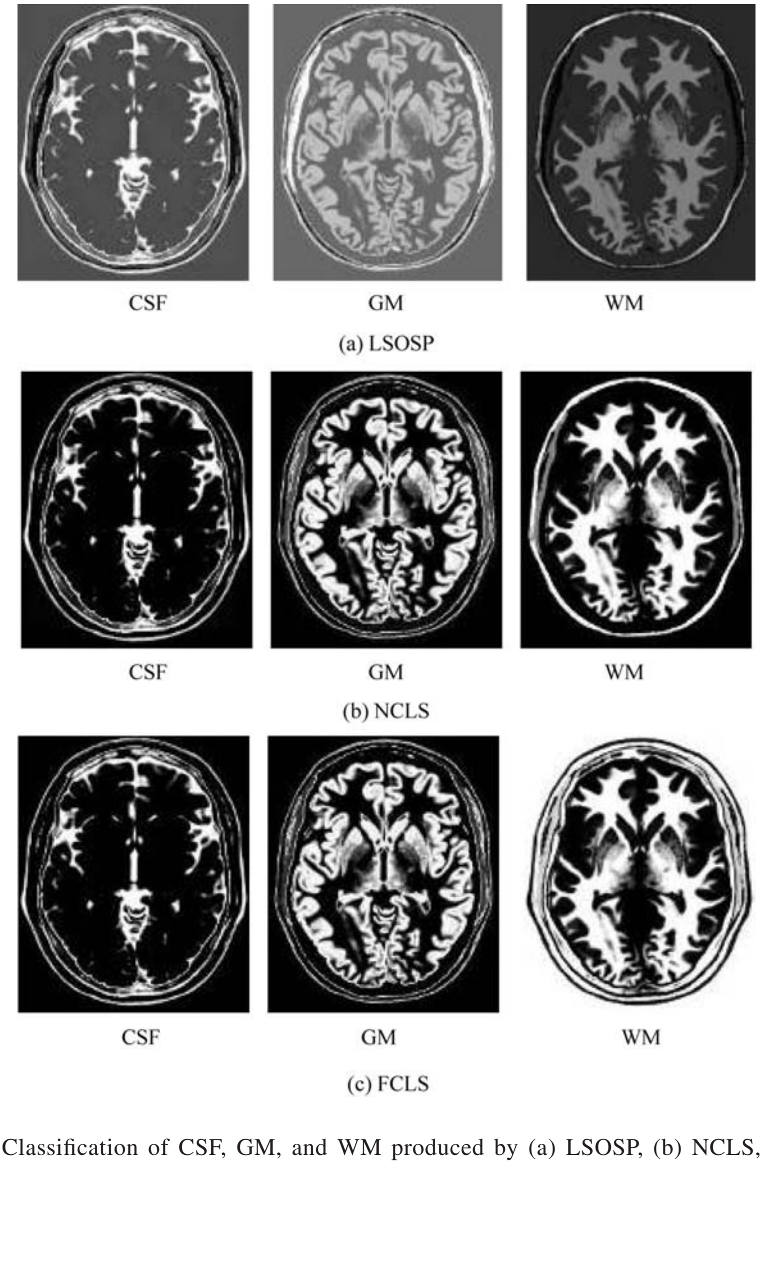
Since only GM, WM, and CSF are major tissues of interest in brain image classification, all other brain tissues in Figure 3.20 are considered as unwanted signatures for suppression. In this case, the SDIN model (3.36) was specified by $\mathbf{S} = [\mathbf{CSF\ GM\ WM}]$ as the signal matrix and $\mathbf{I} = [\mathbf{B\ fat\ muscle\ skin\ skull\ glial\ connective}]$ as the interference matrix which included all brain tissues in Figure 3.20 other than CSF, GM, and WM where the background signature **B** was included as part of the interference matrix **I**. With this interpretation the SBN in (3.45) was considered as a special case of the SDIN. Also, it has been shown in Figures 3.10 and 3.11 that the SBN model was not as good as the SDIB model. In this case, only the SDIN model was considered to be used for LSMA. An LSMA technique is then performed via (3.44) to unmix CSF, GM, and WM where $\mathbf{d}$ = one specific brain tissue to be classified and **U** made up of the other two brain tissues. Similar to the application of hyperspectral image classification presented in Section 3.6.1.2, LSOSP, NCLS, and FCLS were used to unmix the CSF, GFM, and WM where an orthogonal subspace projector $P^{\perp}_{[\mathbf{U}\,\mathbf{I}]}$ was used to reject all unwanted signatures prior to detection of **d** by a matched filter. Figure 3.21 shows classification results unmixed by LSOSP, NCLS, and FCLS where the results produced by NCLS seemed to be the best by visual assessment.

To conduct a quantitative study, we further used 3D ROC analysis for performance evaluation. The mean classification rates of CSF, GM, and WM were calculated based on (3.39) and (3.40) and used to plot their corresponding 3D ROC curves along with three 2D ROC curves in Figure 3.22, where NCLS clearly performed better than the other two methods.

Table 3.4 A_z obtained for LSOSP, NCLS, and FCLS

	LSOSP	NCLS	FCLS
A_z	0.847063	0.916500	0.805803

Table 3.4 also tabulates their AUCs, A_z, calculated for the 2D ROC curves in Figure 3.22(b) produced by LSOSP, NCLS, and FCLS in Figure 3.21; the best classification rate was actually NCLS which yielded $A_z = 0.916500$ compared to LSOSP and NCLS which produced $A_z = 0.847063$ and 0.805803, respectively.

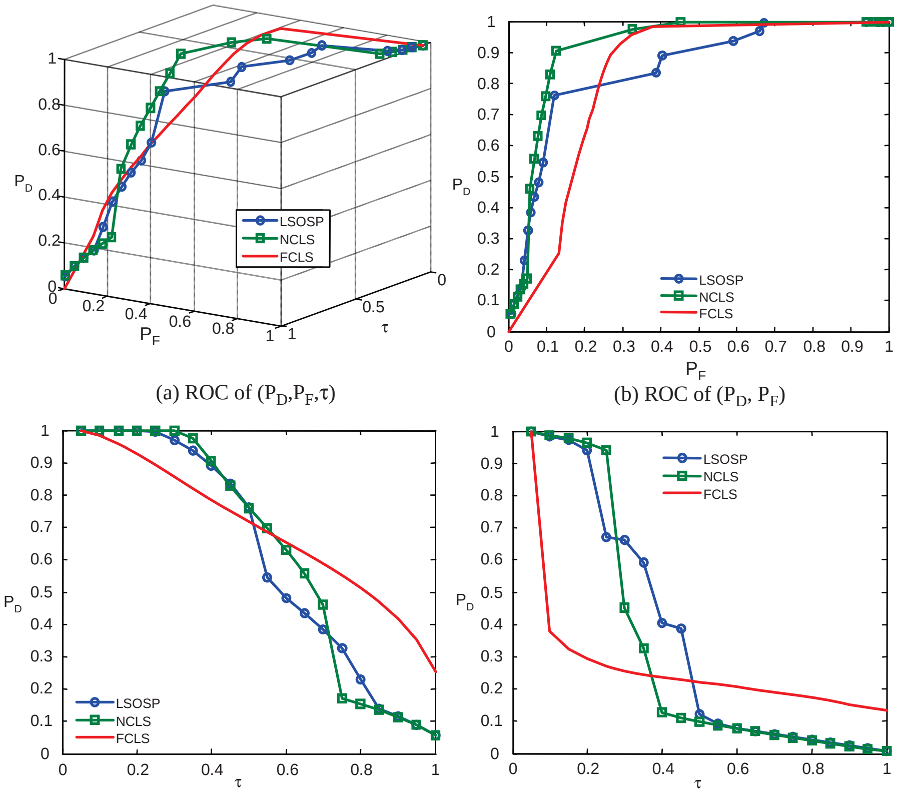


(a) LSOSP

(b) NCLS

(c) FCLS

Figure 3.21 Classification of CSF, GM, and WM produced by (a) LSOSP, (b) NCLS, and (c) FCLS methods.



(a) ROC of (P_D, P_F, τ)

(b) ROC of (P_D, P_F)

(c) ROC of (P_D, τ)

(d) ROC of (P_F, τ)

Figure 3.22 3D ROC of LSOSP, NCLS, and FCLS.

Interestingly, FCLS in Figure 3.21(c) did not perform better than LSOSP as it did for hyperspectral linear unmixing in Figures 3.10(c) and 3.11(c). This is mainly because FCLS was designed to quantify rather than classify brain tissues. Furthermore, for NCLS, the best threshold τ seemed to be around 0.4 according to the three 2D ROC curves in Figures 3.22(b) and 3.22(d). These experiments further provided evidence of usefulness of 3D ROC analysis.

3.6.3 Chemical/Biological Agent Detection

This section describes a third application of 3D ROC analysis in chemical/biological agent detection. After 9/11 the threat from chemical and biological warfare (CBW) agents has become reality, especially contamination and pollution in water supplies during water point selection, production, storage, and distribution to consumers. To address such a need the U.S. Army Joint Service Agent Water Monitor (JSAWM) program is particularly interested in developing a hardware-designed device, named JSAWM HHA (hand-held assay) (Chang, 2006; Liu et al., 2005), based on lateral flow immunoassay technology to determine threat levels of the CBW agents of the incidents. Two tickets with/without an agent signal shown in Figures 3.23(a) and 3.23(b) are developed by the ANP Inc. for HHA and can be read either by human eyes or by an optical scanner. The detection of water ticket samples shown in Figure 3.23(c) was carried out by the software developed at the

Table 3.5 Ticket samples of four signals with various concentrations used for experiments

Concentration	Signal 1	Signal 2	Signal 3	Signal 4	Number of samples
1	0.00E + 00	0.00E + 00	0.00E + 00	0.00E + 00	6
2	3.00E + 01	6.00E + 01	2.00E + 01	2.00E + 01	6
3	3.00E + 01	1.00E + 02	4.00E + 01	4.00E + 01	6
4	6.67E + 01	1.33E + 02	6.00E + 01	3.33E + 01	6
5	1.00E + 02	2.00E + 02	1.00E + 02	7.50E + 01	6
6	1.33E + 02	2.33E + 02	2.00E + 02	1.00E + 02	6
7	2.00E + 02	3.00E + 02	3.00E + 02	1.33E + 02	6
8	3.00E + 02	1.00E + 03	3.00E + 02	3.00E + 02	6
9	3.00E + 02	3.00E + 03	1.00E + 03	1.00E + 03	6
10	1.00E + 03	1.00E + 04	3.00E + 03	3.00E + 03	6
11	2.00E + 03	3.00E + 04	1.00E + 04	1.00E + 04	6
12	3.00E + 03	1.00E + 05	2.00E + 04	4.00E + 04	6

University of Maryland, Baltimore County (UMBC). According to the ground truth provided by the ANP, Table 3.5 tabulates the ticket samples of four signals with various concentrations that were provided by the ANP, Inc.

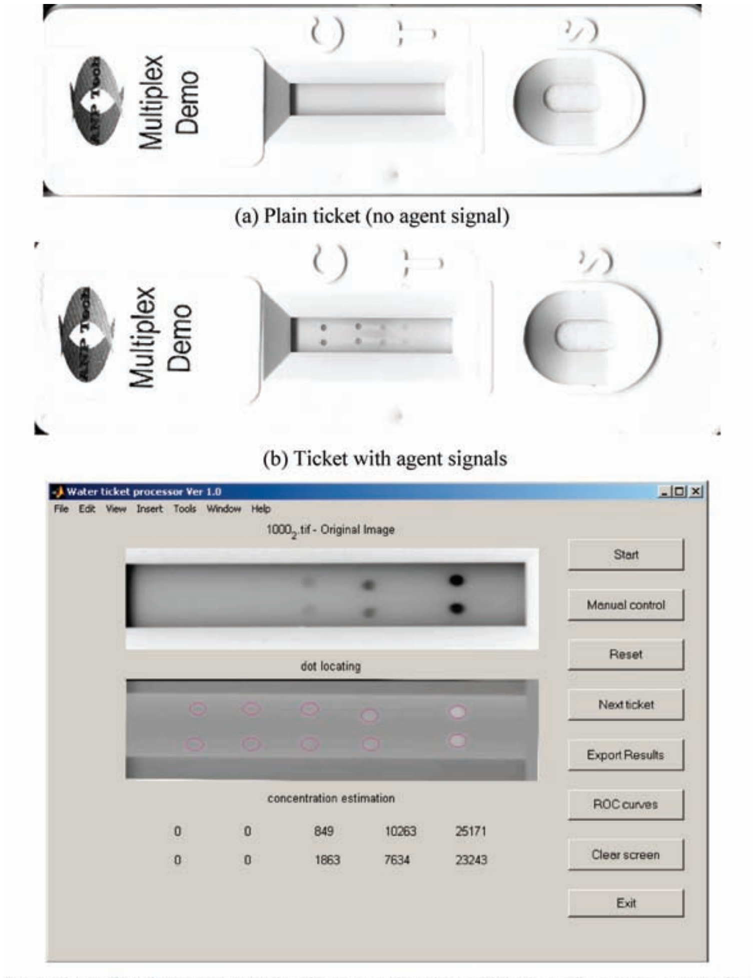
These ticket samples were further used for experiments. To generate ROC curves for this particular application for HHA tickets, we calculated the P_D and the P_F of each agent signal $\mathbf{m}_j$ (signal j) for $j = 1, \ldots, 4$ in the HHA ticket image, where the $\mathbf{m}_j$ was given by different combinations of agent signals such as vaccinia, coxiella, ricin, or bot tow. Setting a concentration threshold for each agent signal to be detected was an important task in chemical and biological applications, where the presence of a threat usually depends on whether its concentration or abundance is above a certain tolerance threshold. If the concentration is greater than the threshold, we claim that the agent signal is detected. Otherwise, we consider that no agent signal is detected.

Since Figure 3.23 shows detected abundance fractions produced by the designed image detection algorithms, it requires the threshold τ to perform target detection. Let $a\%$ be the abundance percentage used as the desired cutoff threshold value. Now for each target signature $\mathbf{m}_j$ we use (3.24) to normalize its detected abundance fraction $\hat{\alpha}_j(\mathbf{r})$ in Figure 3.23 for each single ticket sample $\mathbf{r}$ to $\tilde{\alpha}_j(\mathbf{r})$ by defining

$$\tilde{\alpha}_j(\mathbf{r}) = \frac{\hat{\alpha}_j(\mathbf{r}) - \min_{\mathbf{r}} \hat{\alpha}_j(\mathbf{r})}{\max_{\mathbf{r}} \hat{\alpha}_j(\mathbf{r}) - \min_{\mathbf{r}} \hat{\alpha}_j(\mathbf{r})}. \tag{3.47}$$

As a result, the values of $\tilde{\alpha}_j(\mathbf{r})$ always lie in the range of [0,1]. With $a\%$ as a thresholding criterion we can define an abundance percentage mixed-to-pure pixel converter (APMPCV) with $a\%$-threshold, referred to as $a\%$MPCV, $\chi_{a\%\mathrm{MPCM},\mathbf{m}_j}(\mathbf{r})$ (Chang, 2003a, Chapter 9) as follows:

$$\chi_{a\%\mathrm{MPCM},\mathbf{m}_j}(\mathbf{r}) = \begin{cases} 1 & \text{if} \quad \hat{\alpha}_j(\mathbf{r}) \geq \tau = \dfrac{a}{100} \\ 0 & \text{otherwise.} \end{cases} \tag{3.48}$$



(a) Plain ticket (no agent signal)

(b) Ticket with agent signals

```
0    0    849     10263    25171
0    0    1863    7634     23243
```

(c) A view of UMBC-developed software performing estimation of agent concentration

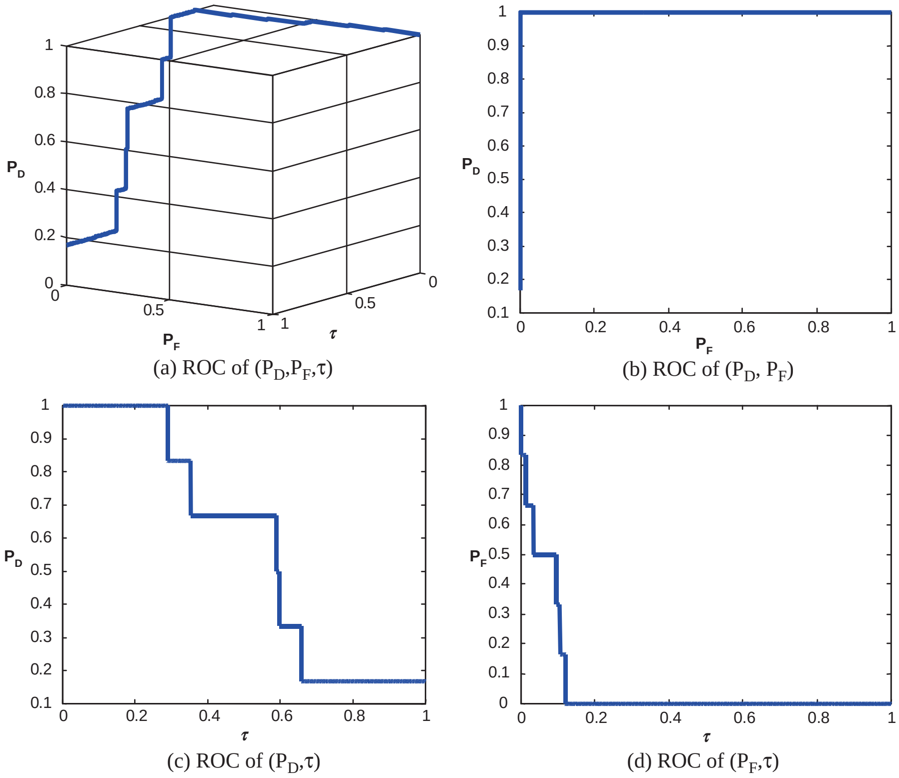
Figure 3.23 An example of tickets with/without agent signals.

If the detected abundance fraction of a target signature, $\mathbf{m}_j$, $\tilde{\alpha}_j(\mathbf{r})$ exceeds $\tau = a\%/100$ within the sample **r**, then the **r** will be assigned to this particular target. So, a "1" produced by (3.48) indicates that the sample **r** is classified as the desired target signature $\mathbf{m}_j$; otherwise, it is detected as the absence of the target signature.

Using (3.47) and (3.48) and threshold $\tau = a\%/100$ with $a\%$ ranging from 0% to 100% we can produce 3D ROC curves of ($\bar{R}_{\mathrm{D}}$,$\bar{R}_{\mathrm{F}}$,τ) for each of four signals shown in Figures 3.24–3.27 along with their AUCs, A_z, tabulated in Table 3.6.

Table 3.6 A_z calculated for four signals Figures 3.24(b)–3.27(b)

	Signal 1	Signal 2	Signal 3	Signal 4
A_z	1	0.6389	0.9167	0.9722



(a) ROC of (P_D,P_F,τ)

(b) ROC of (P_D, P_F)

(c) ROC of (P_D,τ)

(d) ROC of (P_F,τ)

Figure 3.24 3D and 2D ROC curves produced by signal 1.

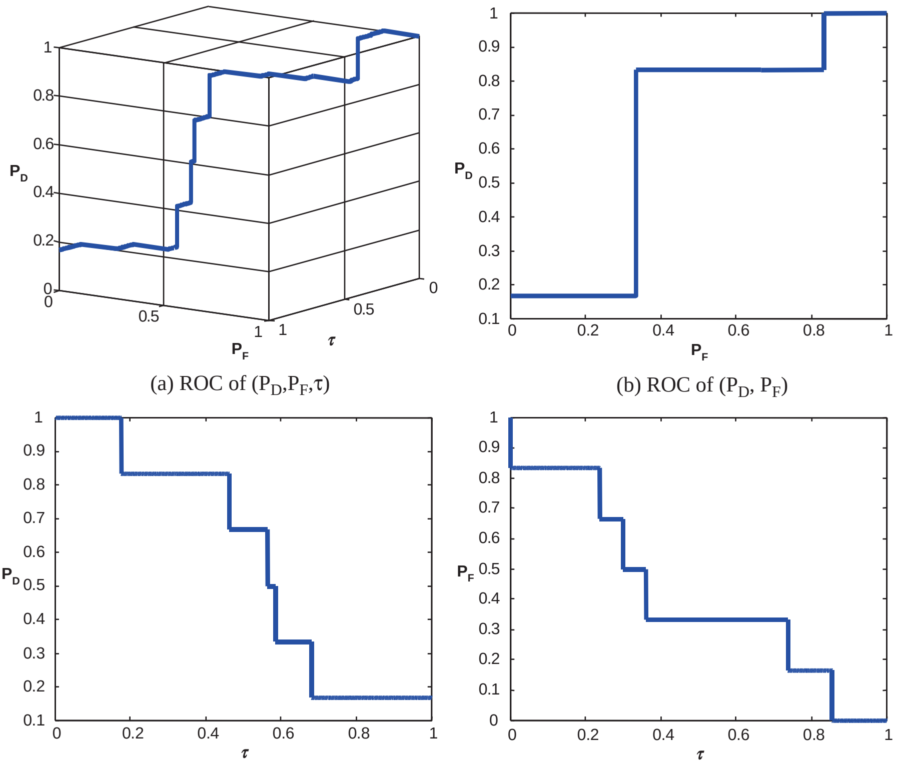
Since Table 3.6 was calculated based on A_z under the 2D ROC curve of ($\bar{R}_{\mathrm{D}}$,$\bar{R}_{\mathrm{F}}$) which did not include concentration as a parameter for performance evaluation, in this case, the 2D ROC curves of ($\bar{R}_D$,τ) and ($\bar{R}_F$,τ) must be factored in analysis of signal sensitivity to tickets. This phenomenon can be explained by the 2D ROC curves of ($\bar{R}_F$,τ) in Figures 3.24(d), 3.25 (d), 3.26(d), and 3.27(d) which provided additional information. That is exactly the reason why we need to introduce a 3D ROC curve to include concentration as a parameter for performance evaluation.

It should be noted that Figures 3.24–3.27 and Table 3.6 were obtained by real ticket samples. As an alternative, we could also use the statistics obtained from the real ticket samples in Table 3.5 to generate Gaussian-fitted 3D and 2D ROC curves. Figures 3.28–3.31 show the Gaussian fitted 3D and 2D ROC curves of Figures 3.24–3.27 along with their corresponding AUCs, A_z, tabulated in Table 3.7.

Like Table 3.6, the best performance based on A_z in Table 3.7 was still produced by signal 1 and the worst came from signal 2. If we further compare the results in Table 3.6 against

Table 3.7 A_z obtained for four signals from Figures 3.28(b)–3.31(b)

	Signal 1	Signal 2	Signal 3	Signal 4
A_z	0.9637	0.6406	0.8887	0.9306



(a) ROC of (P_D,P_F,τ)

(b) ROC of (P_D, P_F)

(c) ROC of (P_D,τ)

(d) ROC of (P_F,τ)

Figure 3.25 3D and 2D ROC curves produced by signal 2.

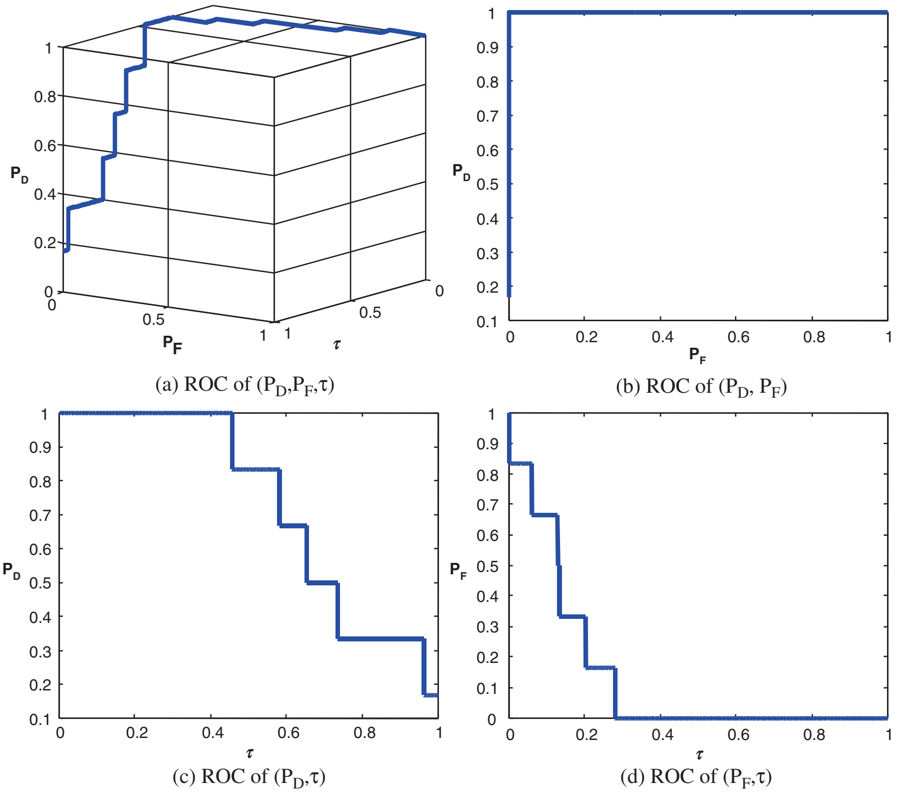
those in Table 3.5 the errors obtained from the ROC generated by real data and those generated by Gaussian-fitted data described in Sections 3.5.2 and 3.5.3 were within the range of 0.04, which indicated the appropriateness of Gaussian-fitted data in this particular application.

3.6.4 Biometric Recognition

In the three applications described earlier, the threshold τ is considered as an independent parameter and P_{D} and P_{F} are treated as dependent parameters as functions of τ. As a matter of fact, according to (2.6) or (3.5), τ is actually determined by the costs c_{ij} defined by

$$\tau = \frac{\pi_0(c_{10} - c_{00})}{\pi_1(c_{10} - c_{11})} \tag{3.49}$$

and prior probabilities π_j which are unknown and cannot be specified in any means beforehand in these applications. However, in some applications the costs may be specified by particular needs. One such application is the evaluation of a biometric recognition system which is generally conducted based on the costs used by the system (Du and Chang, 2007, 2008). To reflect the role that the costs play in the threshold τ, a 3D ROC curve may be better plotted based on



(a) ROC of (P_D,P_F,τ)

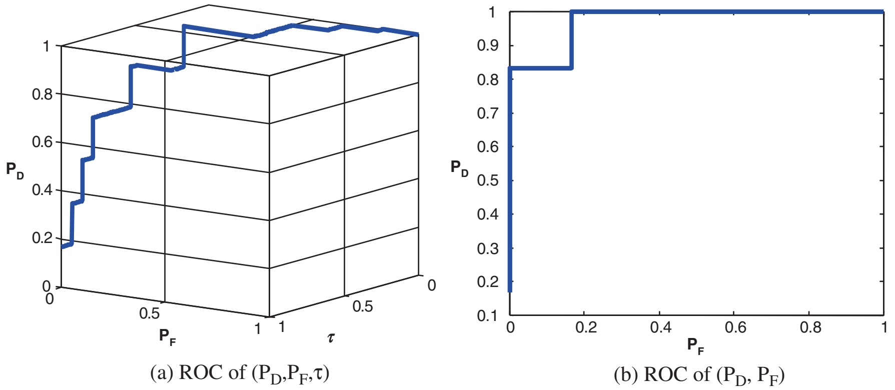
(b) ROC of (P_D, P_F)

(c) ROC of (P_D,τ)

(d) ROC of (P_F,τ)

Figure 3.26 3D and 2D ROC curves produced by signal 3.

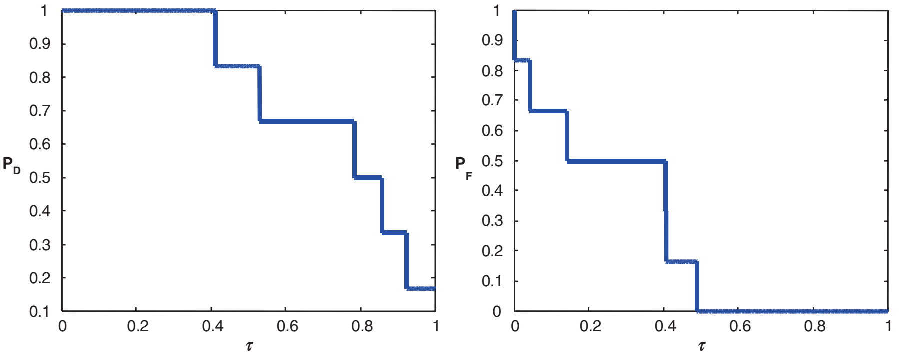
three parameters, P_{D}, P_{F}, and costs via (2.7), (2.9)–(2.10) instead of P_{D}, P_{F}, and τ via (3.49), (3.2)–(3.3). More specifically, P_{D} is replaced by false rejection rate (FRR) which is actually the missed probability P_{M} defined by (3.27) to reflect how a biometric system is sensitive to false rejection rate rather than detection rate. If we further assume that π_j for two hypotheses



(a) ROC of (P_D,P_F,τ)

(b) ROC of (P_D, P_F)

Figure 3.27 3D and 2D ROC curves produced by signal 4.

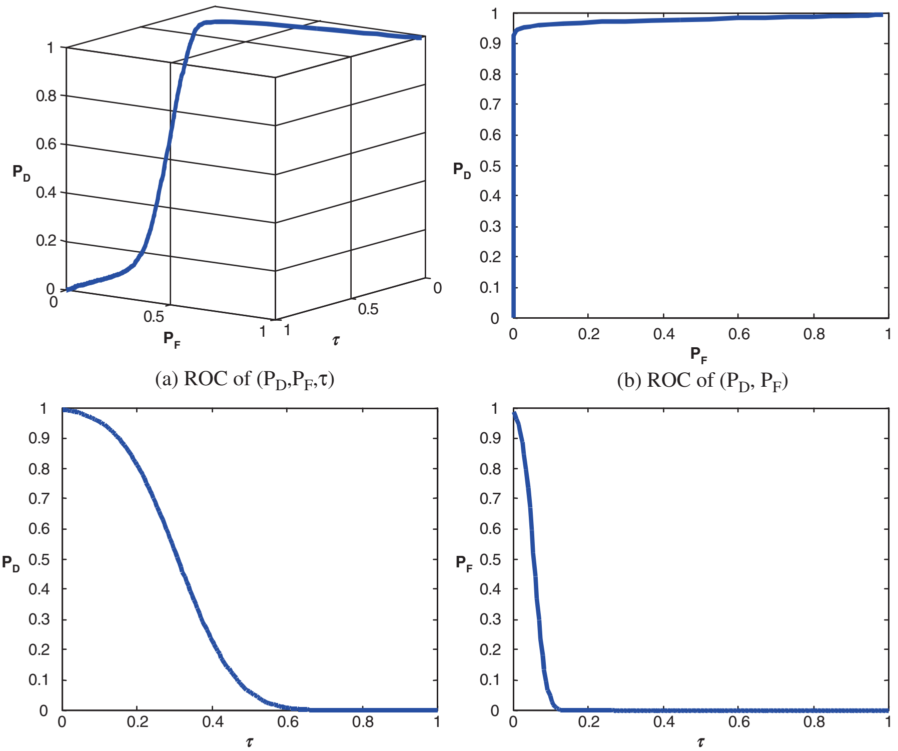


(c) ROC of (P_D,τ) (d) ROC of (P_F,τ)

Figure 3.27 (*Continued*).

are equally likely, this implies that two systems evaluated under each hypothesis have equal chance to be used for the application. As a result, the averaged risk $r(\delta)$ defined by (2.4) and the cost specified by (3.49) become

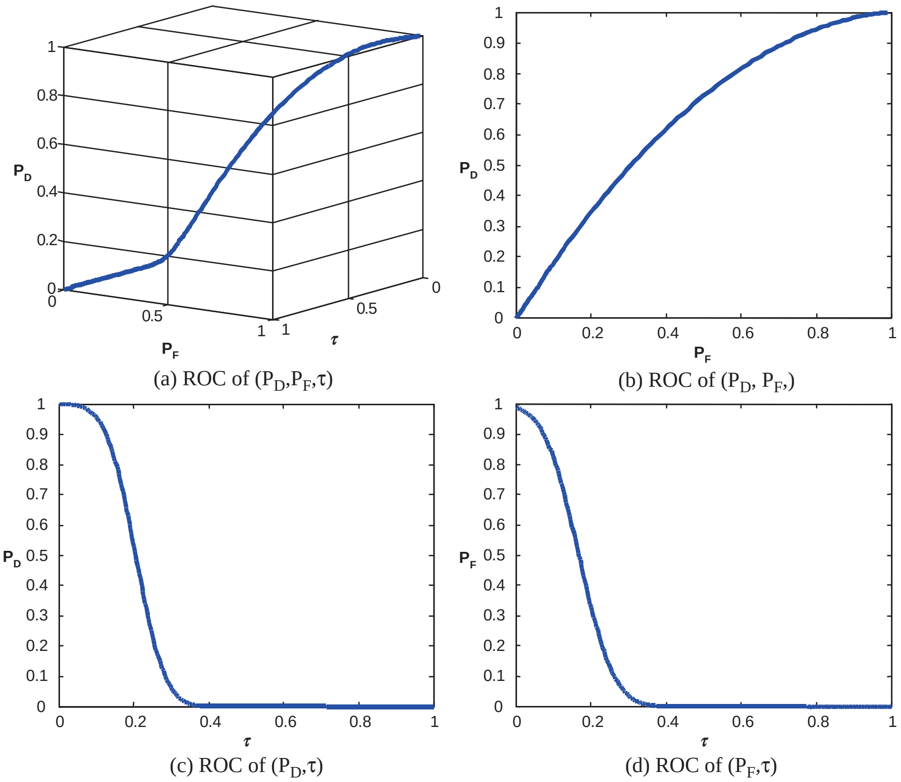
$$r(\delta) = (1/2)R_0(\delta) + (1/2)R_1(\delta), \tag{3.50}$$



(a) ROC of (P_D,P_F,τ) (b) ROC of (P_D, P_F)

(c) ROC of (P_D,τ) (d) ROC of (P_F,τ)

Figure 3.28 Gaussian-fitted 3D and 2D ROC curves produced by signal 1.



(a) ROC of (P_D,P_F,τ)

(b) ROC of (P_D, P_F,)

(c) ROC of (P_D,τ)

(d) ROC of (P_F,τ)

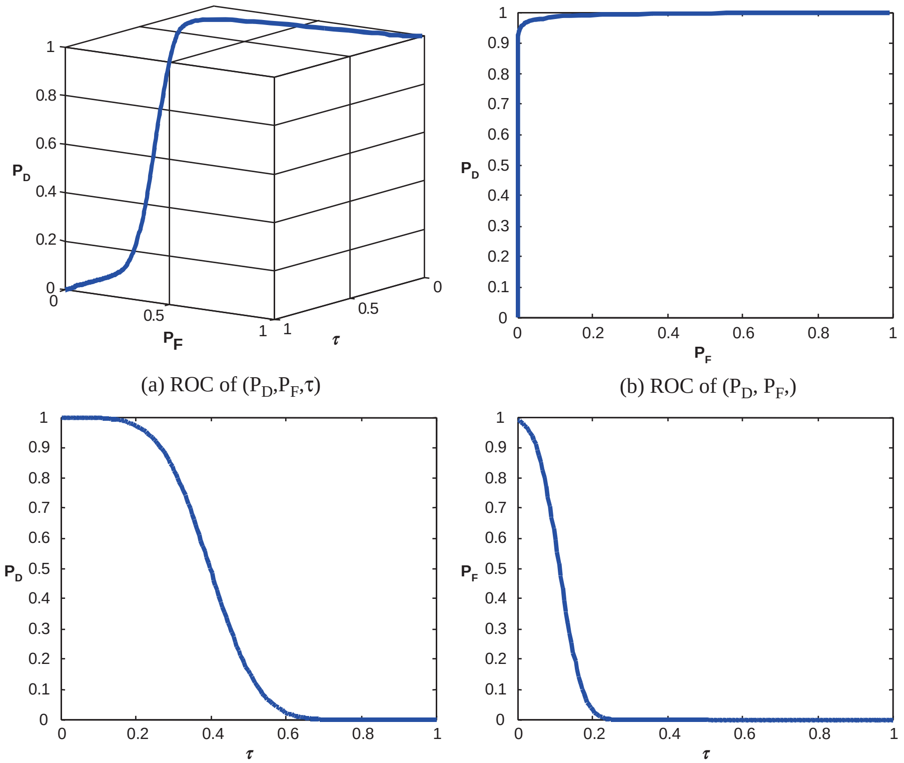
Figure 3.29 Gaussian-fitted 3D and 2D ROC curves produced by signal 2.

$$\tau = \frac{c_{10} - c_{00}}{c_{10} - c_{11}}, \tag{3.51}$$

respectively. Since we generally assume that there is no cost for TN and TP in which case $c_{00} = c_{11} = 0$, (60) can be further reduced to

$$\text{cost} = r(\delta) = (1/2)c_{10}P_0(\Gamma_1) + (1/2)c_{01}P_1(\Gamma_0), \tag{3.52}$$

which is reduced to a cost function of P_{D} and P_{F} specified by (3.2) and (3.3), respectively. As shown in Du and Chang (2007, 2008), a 3D ROC curve, referred to as the 3D cost curve of (FRR, P_{F}, *cost*), can then be plotted by the three parameters, FRR = 1 − P_{D}, P_{F}, and cost specified by (3.52) instead of the threshold τ. Consequently, in addition to the three 2D ROC curves of (P_{D}, P_{F}), (P_{D},τ), (P_{F},τ) that are already introduced in the 3D ROC analysis, there are three more additional 2D ROC curves of (P_{D}, *cost*), (P_{F}, *cost*), and (τ, *cost*) that can be further derived by introducing (3.50) as a cost via (3.51) parameter. Furthermore, upon evaluating a biometric recognition system, users may care more about relationships among *Security*, *Convenience*, τ, and *cost*. Under such circumstances, various 3D ROC curves, called 3D combinational curves in Du and Chang (2007, 2008), can be plotted based on these four parameters of interest, FER = 1 − P_{M}, P_{F}, τ, and *cost* along with their corresponding 2D combinational ROC curves by defining



(a) ROC of (P_D,P_F,τ) (b) ROC of $(P_D, P_F,)$

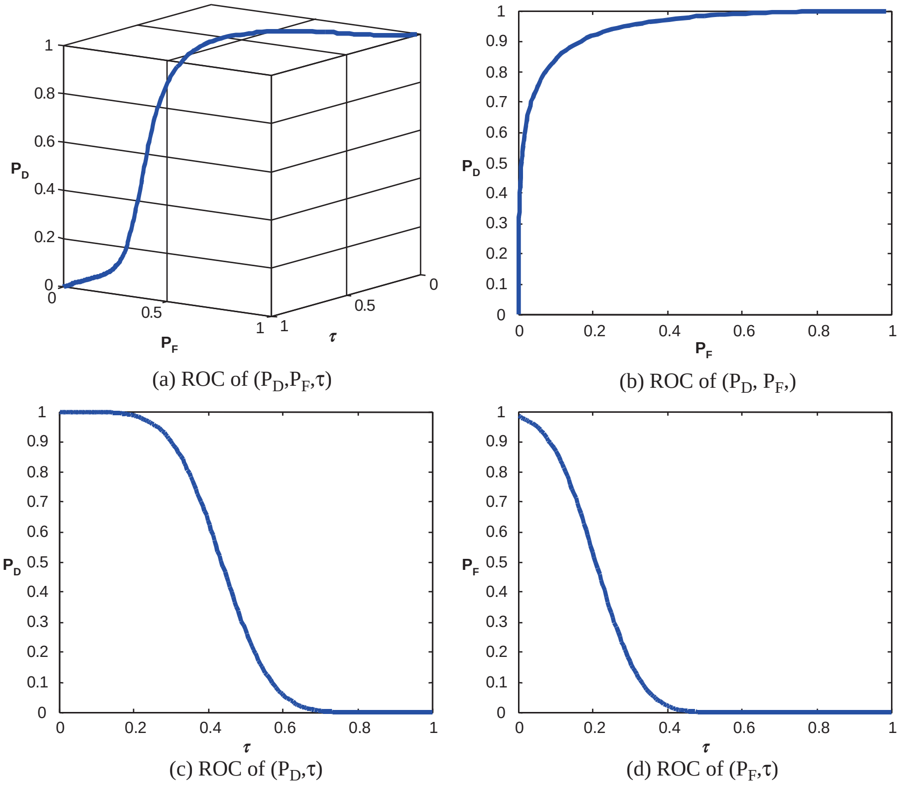
(c) ROC of (P_D,τ) (d) ROC of (P_F,τ)

Figure 3.30 Gaussian fitted 3D and 2D ROC curves produced by signal 3.

the system security by $1-P_F=P_{TN}$ via (3.13) and the system convenience by $1-\text{FRR}=P_D$. For example, in the context of *Security*, *Convenience*, τ, and *cost*, a 3D combinational performance ROC curve and a 3D combinational performance cost curve can be specified by a 3D ROC curve of (*Convenience*, *Security*, τ) and a 3D ROC curve of (*Convenience*, *Security*, *cost*), respectively, along with their six corresponding 2D combinational ROC curves of (*Convenience*, *Security*), (*Security*, τ), (*Convenience*, τ), and (*Security*, *cost*), (*Convenience*, cost), and (*cost*, τ). For example, a 2D combinational ROC curve of (*Convenience*, Security) = (P_D, P_{TN}) represents a compromise of a biometric system between security and convenience. Since the utility of 3D ROC analysis in the evaluation of biometric recognition systems has been investigated and explored in great detail in Du and Chang (2008) and extensive experimental results are also conducted therein, no experiments are included here to avoid duplication.

3.7 Conclusions

This chapter presents a 3D ROC analysis for multiple-signal detection and classification. Its idea arises from the fact that the performance of a detector is generally measured by a likelihood ratio test which is indeed a real-valued function, both detection power and false alarm rate are actually determined by a value that thresholds a real-valued detector. This problem is not only a pattern classification, where classification accuracy is also determined by a threshold used for clustering



(a) ROC of (P_D,P_F,τ)

(b) ROC of (P_D, P_F,)

(c) ROC of (P_D,τ)

(d) ROC of (P_F,τ)

Figure 3.31 Gaussian fitted 3D and 2D ROC curves produced by signal 4.

data. Therefore, including a threshold as a parameter to account for performance analysis seems more realistic and effective. To substantiate its utility in versatile applications, four examples representing a wide range of applications are presented for demonstration, which are linear spectral unmixing and target detection for hyperspectral data, medical diagnosis in tumor detection, and tissue characterization for magnetic resonance images, chemical/biological agent detection for water monitoring, and evaluation of biometric recognition systems.

4

Design of Synthetic Image Experiments

Many hyperspectral imaging algorithms have been developed for various applications such as spectral unmixing, subpixel detection, quantification, endmember extraction, classification, compression, as well as many more yet to explored. While each algorithm deserves its own right, it is very difficult to compare them one against another without a fair common ground. This chapter makes an attempt to design a set of standardized synthetic images for hyperspectral target analysis which simulate various scenarios so that different algorithms can be validated and evaluated on the same setting with completely controllable environments. Here, the term "target" used here is generic and simply indicates an object of interest in data analysis where a real target is specified by a certain application such as endmembers, anomalies, and man-made objects. Two types of scenarios are developed to simulate how a target can be inserted into the image background. One is called target implantation (TI) which implants a target by removing the background pixels they intend to replace. This type of scenario is of particular interest in endmember extraction where pure signatures can be simulated and inserted into the background with a target of guaranteed 100% purity. The other is called target embeddedness (TE) which embeds a target by superimposing it over the background pixels they intend to insert. This type of scenario can be used to simulate signal detection models where the noise and background pixels are additive, that is, signal detection in additive noise. It is worth noting that TE does not satisfy abundance sum-to-one constraint (ASC) due to superimposition of an inserted target pixel over a background pixel. Furthermore, for each type of target insertion three scenarios are designed to simulate different levels of target knowledge by adding a Gaussian noise. To make these six scenarios (three for TI and three for TE) standardized data sets for experiments, the data used to generate synthetic images can be chosen from a database or a spectral library available in the public domain or on website to avoid biased data being used for validation. By virtue of these designed six scenarios, an algorithm can be evaluated objectively and compared impartially to other algorithms in the same environment with completely controllable target knowledge. To further demonstrate how these six scenarios can be used for performance evaluation and analysis, various algorithms developed for applications of subpixel detection, mixed pixel classification/quantification, and endmember extraction are used for comparison.

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

4.1 Introduction

When a new algorithm is designed and developed, a frequently asked question is that "How does it perform compared to other algorithms?". In other words, if one walks in with a new algorithm saying that his algorithm performs better than other existing algorithms, how do we substantiate his claim? This is particularly true for a new area such as hyperspectral imaging where new algorithms keep emerging and popping up in a fast pace and each algorithm claims to be better than others. Many users have been struggling and wrestling this issue when they come to select candidate algorithms for hardware architecture design and development such as field programmable gate array (FPGA). Despite the fact that computer-simulated data such as Monte Carlo simulations have been used for this purpose, on many occasions the computer simulations generally go far beyond reality and can be only used for proof-of-concept. In order to move to the next level, more realistic data are needed for further evaluation. The same dilemma occurs in medical community also where the so-called phantoms are designed and developed based on real data using controllable parameters to simulate real environments before experiments can be conducted for real data *in vivo*. One good example is the magnetic resonance (MR) brain web library provided by MR imaging simulator of McGill University, Montreal, Canada (available at www.bic.mni.mcgill.ca/brainweb/) (see Chapter 32). It seems natural that a similar approach to McGill's synthetic images can also be adopted for hyperspectral imaging. This chapter takes this challenge and designs so-called synthetic images that serve the same purpose as phantoms designed for medical imaging.

Before doing so several issues must be addressed. First, it is important to note that none of the algorithms can claim its superiority over other algorithms without specifying criteria to be used for optimality and applications for which they are designed. More specifically, in what sense of optimality and in what application can an algorithm perform better than other algorithms? Second, there should be a database or a spectral library available in the public domain that allows users to perform impartial assessments and objective comparative studies and analyses. Additionally, all the experiments conducted should be repeatable for validation so that any claimed algorithm can be verified by others using the same data set and identical environment under which the experiments are conducted. Only in this way it can prevent users from being accused of the use of their own data sets to make their own cases. Most important of all is how to design an effective and objective evaluation process to compare algorithms without controversy and subjectivity. Interestingly, to the author's best knowledge, no such effort has ever been made in the past. In this chapter, we investigate and focus on the first and third issues since the second issue can be resolved by many data sets available on website now. Besides, the third issue is also closely related to the first issue which is completely determined by applications. In this case, both issues will be investigated in a coherent manner when it comes to design of experiments.

This chapter first addresses the third issue of how to design and develop a creditable evaluation process. Unlike most computer simulations which generally use laboratory data sets via a set of parameters such as Monte Carlo simulations, here we use real data sets to design synthetic images that can simulate real images with certain properties that we would like to explore. Because hyperspectral imaging sensors can effectively capture targets that generally appear in a mixed form or at subpixel level, it is important to design a process of how such targets can be inserted into an image scene at our discretion. Two types of target insertion are considered, target implantation (TI) and target embeddedness (TE), where the former implants targets by removing the background pixels they intend to replace as opposed to the latter which embeds targets by superimposing the targets over the background pixels they intend to insert. While TI is of particular interest in endmember extraction since pure signatures can be simulated and inserted into the background with guaranteed 100% purity, TE is particularly useful in target detection where the noise and background are assumed to be additive. For each type of target insertion three scenarios can be designed to simulate different levels of target knowledge by adding a Gaussian noise. In order for the designed six scenarios (three scenarios for TI, TI1, TI2, TI3 and three scenarios for TE, TE1, TE2, TE3) to be used as standardized data sets for experiments, the data sets used to generate synthetic images can be chosen from existing databases or spectral libraries available in the public domain. Therefore, no particular data sets are required to simulate these synthetic images. By virtue of these particularly designed six scenarios, an algorithm can be assessed and evaluated objectively and compared fairly to other algorithms on the same setting.

4.2 Simulation of Targets of Interest

As noted earlier, a target used in this book is referred to as an object whose existence can be spectrally characterized by certain properties, for example, statistics of spectral correlation across the wavelength range used for data acquisition. Generally two types of targets are of interest in hyperspectral target analysis, subsample targets, and mixed-sample targets, as discussed in Chapter 2.

4.2.1 Simulation of Synthetic Subsample Targets

First, we simulate a subsample target. Assume that a subsample target is specified by a signature, **p**, for example, panel signature in Figures 1.8–1.10, 1.12(c), (d) and 1.16–three subsample targets with $\frac{3}{4}$, $\frac{1}{2}$, and $\frac{1}{4}$ sample sizes, respectively. Figure 4.1(a) shows how a subsample target $\mathbf{t}_1$ with $\frac{3}{4}$ sample size is simulated by the panel signature, **p**. To simulate the subsample target $\mathbf{t}_1$, we first simulate one-sample vector specified by a background signature **b** and three-sample vectors specified by **p** to form a four-sample square panel as shown at the bottom layer of Figure 4.1(a).

The four-sample square panel is then shrunk to its $\frac{1}{4}$ size by averaging all four sample vectors to a single four-subsample square panel with the same spatial resolution 1.56 m shown at the top layer of Figure 4.1(a) where each subsample vector in the shrunk single-sample square panel is only $\frac{1}{4}$ size of its corresponding sample vector at the bottom layer of Figure 4.1(a) as a result of $1/4(\mathbf{p}+\mathbf{p}+\mathbf{p}+\mathbf{b})$. This shrinking process is actually a linear mixture of two signatures, **p** and **b**, with abundance fractions corresponding to $\frac{3}{4}$ and $\frac{1}{4}$, respectively, and similar to the process used in Chang *et al.* (2004). The shrunk single four-subsample vector at the top layer of Figure 4.1(a) is the desired sample vector $\mathbf{p}_1$ that contains a subsample target $\mathbf{t}_1$ of $\frac{3}{4}$ sample size specified by the panel signature **p**. Similarly, two-sample vectors $\mathbf{p}_2$ and $\mathbf{p}_3$ shown in Figure 4.1(b) and,(c) are also simulated in the same fashion as results of $1/4(\mathbf{p}+\mathbf{p}+\mathbf{b}+\mathbf{b})$ and $1/4(\mathbf{p}+\mathbf{b}+\mathbf{b}+\mathbf{b})$, respectively, where $\mathbf{p}_2$ contains a subsample target $\mathbf{t}_2$ of $\frac{1}{2}$ sample size specified by creosote leaves and $\mathbf{p}_3$ contains $\mathbf{t}_1$ of $\frac{1}{4}$ sample size specified by creosote leaves.

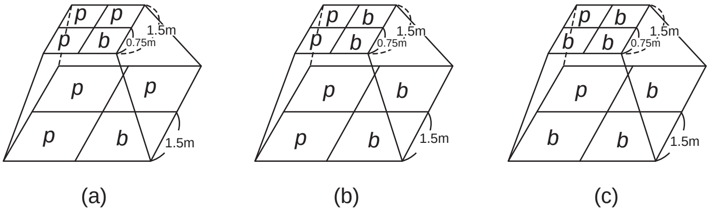


Figure 4.1 Simulations of three subsample target panels: (a) subsample target panel, $\mathbf{p}_1$ with $\frac{3}{4}$ of $\mathbf{p}+\frac{1}{4}$ of **b**; (b) subsample target panel, $\mathbf{p}_2$ with 1/2 of $\mathbf{p}+1/2$ of **b**; (c) subsample target panel $\mathbf{p}_3$ with $\frac{1}{4}$ of $\mathbf{p}+\frac{3}{4}$ of **b**.

4.2.2 Simulation of Synthetic Mixed-Sample Targets

In general, simulating a synthetic mixed-sample target can be done in a similar way as the subsample targets are simulated in Figure 4.1. However, there is a simple method to do it. If we assume that the abundance fraction of a signature present in a sample vector is proportional to its size embedded in a sample vector, a subsample target of $^1/_4$ sample size can be simulated by 25% of its abundance provided that the abundance of a fully occupied sample vector is assumed to be 100%. In light of this interpretation, the three subsample targets, $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$, in Figure 4.1 can be re-expressed as $\mathbf{p}_1 = 75\%\mathbf{p} + 25\%\mathbf{b}$, $\mathbf{p}_2 = 50\%\mathbf{p} + 50\%\mathbf{b}$, $\mathbf{p}_3 = 25\%\mathbf{p} + 75\%\mathbf{b}$. As a result, we can simulate any arbitrary sample vector $\mathbf{p}$ linearly mixed by a number of signatures, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ with their corresponding abundance fractions specified by $\alpha_1, \alpha_2, \ldots, \alpha_p$ with $\alpha_j \geq 0$ for all $0 \leq j \leq p$ and $\sum_{j=1}^{p} \alpha_j = 1$ as $\mathbf{p} = \sum_{j=1}^{p} \alpha_j \mathbf{m}_j$.

4.3 Six Scenarios of Synthetic Images

Section 4.2 describes how to simulate subsample targets or mixed-sample targets according to their characteristics. In this section, we will discuss on how to simulate synthetic images with target panels inserted in accordance with certain desired properties.

4.3.1 Panel Simulations

First, the real image scene with reflectance data shown in Figure 1.12(c) is used to simulate panels of interest where the reflectance spectra of five USGS ground-truth mineral spectra: alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M) are used to simulate 25 panels of various sizes that are arranged in a 5 × 5 matrix as shown in Figure 4.2.

Each row of the five panels in Figure 4.2 is simulated by the same mineral signature and each column of five panels has the same size. Among 25 panels are five 4 × 4-pure pixel panels, $\mathrm{p}^{i}_{4\times4}$ for $i = 1, \ldots, 5$ in the first column, five 2 × 2-pure pixel panels, $\mathrm{p}^{i}_{2\times2}$ for $i = 1, \ldots, 5$ in the second column, five 2 × 2-mixed pixel panels, $\left\{\mathrm{p}^{i}_{3,jk}\right\}_{j=1,k=1}^{2,2}$ for $i = 1, \ldots, 5$ in the third column, five

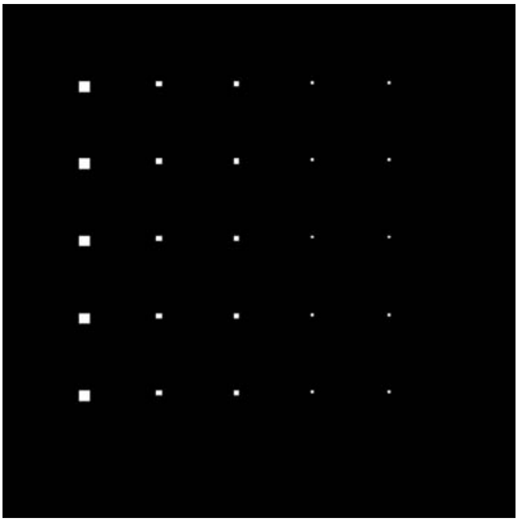


Figure 4.2 25 simulated panels.

Table 4.1 Simulated 20 mixed panel pixels in the first column

Row 1	$p_{3,11}^{1} = 0.5A + 0.5B$	$p_{3,12}^{1} = 0.5A + 0.5C$
	$p_{3,21}^{1} = 0.5A + 0.5K$	$p_{3,22}^{1} = 0.5A + 0.5M$
Row 2	$p_{3,11}^{2} = 0.5B + 0.5A$	$p_{3,12}^{2} = 0.5B + 0.5C$
	$p_{3,21}^{2} = 0.5B + 0.5K$	$p_{3,22}^{2} = 0.5B + 0.5M$
Row 3	$p_{3,11}^{3} = 0.5C + 0.5A$	$p_{3,12}^{3} = 0.5B + 0.5C$
	$p_{3,21}^{3} = 0.5C + 0.5K$	$p_{3,22}^{3} = 0.5C + 0.5M$
Row 4	$p_{3,11}^{4} = 0.5K + 0.5A$	$p_{3,12}^{4} = 0.5K + 0.5B$
	$p_{3,21}^{4} = 0.5K + 0.5C$	$p_{3,22}^{4} = 0.5K + 0.5M$
Row 5	$p_{3,11}^{5} = 0.5M + 0.5A$	$p_{3,12}^{5} = 0.5M + 0.5B$
	$p_{3,21}^{5} = 0.5M + 0.5C$	$p_{3,22}^{5} = 0.5M + 0.5K$

subpixel panels, $p_{4,1}^{i}$ for $i = 1, \ldots, 5$ in the fourth column, and five subpixel panels, $p_{5,1}^{i}$ for $i = 1, \ldots, 5$ in the fifth column. The purpose of introducing five panels in the third column is to conduct a study and analysis on the five mineral signatures with different mixtures in a pixel. Table 4.1 tabulates the mixing details of mineral composition in the 20 panels in the third column, while subpixel panels in the fourth and fifth columns are simulated with their simulated abundance fractions tabulated in Table 4.2, where the background (BKG) is simulated by the sample mean of the real cuprite image scene in Figure 1.12(a).

According to Tables 4.1 and 4.2, the simulated synthetic image in Figure 4.2 has a total of 130 panel pixels present in the scene, 80 pure pixel panels in the first column, 20 pure pixel panels in the second column, 20 mixed panel pixels in the third column, five 50%-abundance subtarget panel pixels in the fourth column, and five 25%-abundance subpixel target panel pixels in the fifth column. These 130 pixel panels include a total of 26 spectrally distinct signatures (5 pure mineral signatures in the first and second columns, 20 mixed signatures in the third column 5 along with 1 BKG signature). Figure 4.3 graphically plots the abundance fractions of all these 130 panel pixels where (a-e) indicate the five mineral signatures, A, B,C,K, and M, used to simulate 26 panel pixels in each of five rows in Figure 4.2 respectively.

By virtue of the 25 simulated panels in Figure 4.2, two target insertions, TI and TE, can be designed to be used for experiments conducted in this book.

1. *Target implantation (TI):* Three scenarios to implant target pixels into the background are simulated in such a manner that the target pixels are inserted into the background while their corresponding background pixels are removed. This type of scenario is mainly designed to simulate pure target pixels for extraction. The utility of TI includes applications such as endmember extraction (Chapters 7–11), mixed pixel detection, classification, and quantification (Chapters 12–18).

Table 4.2 Abundance fractions of subpixel panels in the fourth and fifth columns

Row	Fourth column	Fifth column
1	$p_{4,11}^{1} = 0.5A + 0.5BKG$	$p_{5,11}^{1} = 0.25A + 0.75BKG$
2	$p_{4,11}^{2} = 0.5B + 0.5BKG$	$p_{5,11}^{2} = 0.25B + 0.75BKG$
3	$p_{4,11}^{3} = 0.5C + 0.5BKG$	$p_{5,11}^{3} = 0.25C + 0.75BKG$
4	$p_{4,11}^{4} = 0.5K + 0.5BKG$	$p_{5,11}^{4} = 0.25K + 0.75BKG$
5	$p_{4,11}^{5} = 0.5M + 0.5BKG$	$p_{5,11}^{5} = 0.25M + 0.75BKG$

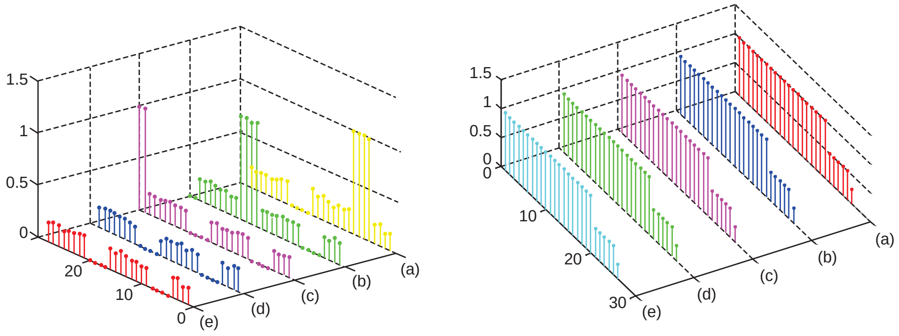


Figure 4.3 Graphical plots of abundance fractions of 130 panel pixels in Figure 4.2.

2. *Target embeddedness (TE):* Three scenarios to embed target pixels into the background are simulated in such a manner that the targets are inserted into the background by adding the targets directly to their corresponding background pixels. In other words, target pixels are superimposed over their corresponding background pixels instead of removing their corresponding background pixels to accommodate the target pixels as the way target implantation does. This type of scenario is primarily designed for target detection where a binary hypothesis testing problem is cast for detection (see Chapter 2) with the null hypothesis H_0 representing background with additive noise against the alternative hypothesis H_1 representing signal (target pixels) plus background (background pixels) with additive noise. It can also be used to test whether or not an endmember extraction algorithm can extract the most purest signatures, which also turn out to be embedded target pixels but do not have 100% purity of signatures. A salient difference between TI and TE is worth being mentioned. Since the three TE scenarios insert targets by adding target pixels to and superimposing over background pixels instead of replacing background pixels as the way the three TI scenarios do for their target insertion, the abundance fraction of the pixel into which a target pixel is embedded is not summed to one. These three TE scenarios violated the abundance sum-to-one constraint (ASC) generally imposed on linear spectral mixture analysis (LSMA), and thus, they cannot be used for quantification as shown in Chang et al. (2010) and Figure 4.14 (Section 4.4.2.2). Nevertheless, they are well suited for detection-in-noise model analysis.

4.3.2 Three Scenarios for Target Implantation (TI)

Three interesting scenarios for TI, Scenario TI1, Scenario TI2, and Scenario TI3, presented in this section, are designed for applications in target extraction such as endmember extraction (see Part II) and target quantification. The 25 panels in Figure 4.2 are used as targets of interest and implanted in a synthetic image scene with size of 200 × 200 pixel vectors in a way that the targets to be implanted replace their corresponding background pixels. Each of these three scenarios is described as follows.

4.3.2.1 Scenario TI1 (Clean Panels Implanted into Clean Background)

This scenario assumes that the image background is clean and simulated by only one BKG signature. The 25 clean panels simulated in Figure 4.2 are then implanted in the background by replacing their corresponding background pixels with the clean panel pixels. The resulting image is a synthetic image shown in Figure 4.4 with clean panels implanted in the clean background image scene.

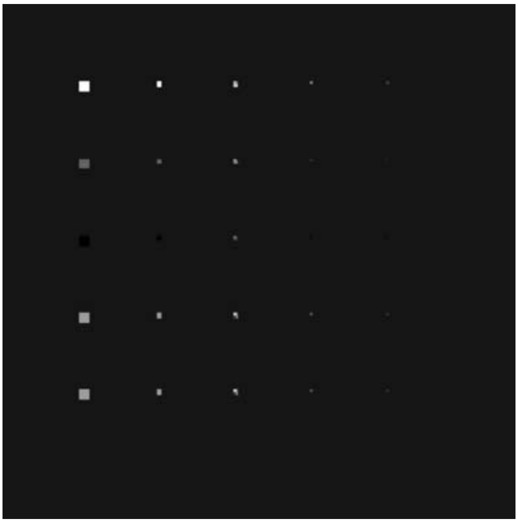


Figure 4.4 Synthetic image simulated by Scenario TI1.

4.3.2.2 Scenario TI2 (Clean Panels Implanted into Noisy Background)

Practically, Scenario TI1 does not exist because of no noise present in the data. Scenario TI2 is more realistic when the noise-free background in Scenario TI1 is replaced with a noisy background image which is corrupted by an additive Gaussian noise to achieve a signal-to-noise ratio (SNR) = 20:1 defined as 50% signature (i.e., reflectance/radiance) divided by the standard deviation of the noise in Harsanyi and Chang (1994). Then clean targets are implanted into such simulated noisy background image. So, the resulting synthetic image has clean targets implanted in a noisy background as shown in Figure 4.5. This scenario simulates a case that true clean targets are indeed present in a noisy image background for target extraction.

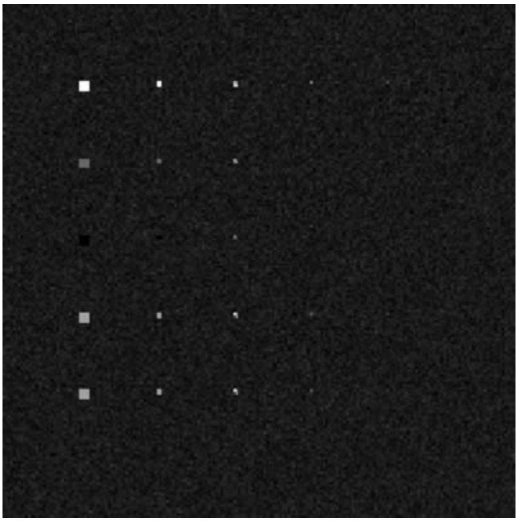


Figure 4.5 Synthetic image simulated by Scenario TI2.

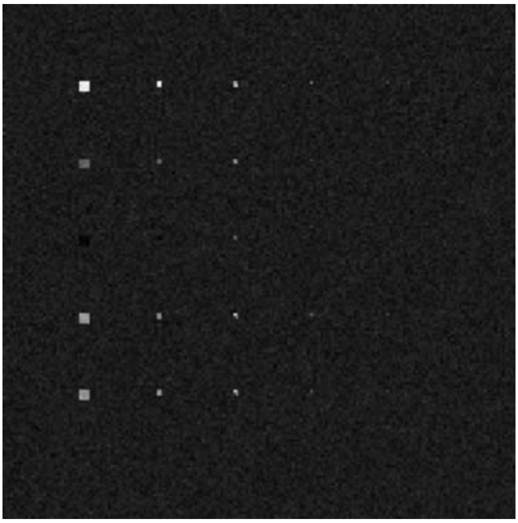


Figure 4.6 Synthetic image simulated by Scenario TI3.

The synthetic image scene in this scenario is the same as the one in Scenario TI1 except that the image background is not clean, but rather corrupted by an additive Gaussian noise with SNR = 20:1.

4.3.2.3 Scenario TI3 (Gaussian Noise Added to Clean Panels Implanted into Clean Background)

Scenario TI3 is the same as Scenario TI1 except that a Gaussian noise is added to TI1 to achieve an SNR = 20:1. So, in this synthetic image, the clean targets and clean background image are both corrupted by an additive Gaussian noise with SNR = 20:1 as shown in Figure 4.6. It is also similar to Scenario TI2 but the implanted targets are now noise-corrupted compared to the clean targets implanted in Scenario TI2.

This scenario simulates a case that the implanted targets are not original true targets and have been contaminated and corrupted by noise. As a consequence of noise corruption, all the pure pixels, mixed pixels, and subpixels are contaminated. So, technically, those panel pixels of 100% purity as endmembers are no longer pure. However, these implanted targets are still considered to be purest and closest to the original targets compared to other pixels present in the image scene. So, they can still be considered as targets of interest. This scenario is designed to test and evaluate how sensitive a target extraction algorithm can be when clean targets are corrupted by noise.

4.3.3 Three Scenarios for Target Embeddedness (TE)

In the previous sections, three scenarios for TI are simulated by inserting the 25 panels into the image scenes by removing background pixels to accommodate these 25 panels for target implantation. As an alternative, this section simulates another type of target insertion, called TE, which inserts the 25 panels into an image scene with a size of 200 × 200 pixel vectors by adding the 25 panels directly to an image scene in such a way that the 25 panels are simply superimposed over their corresponding background pixels. In other words, unlike target implantation in Scenarios TI1, TI2, and TI3 which replaced background pixels with 25 implanted panel pixels, the

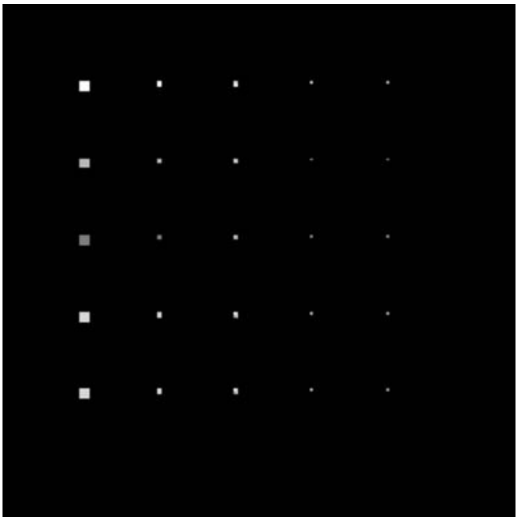


Figure 4.7 Synthetic image simulated by Scenario TE1.

following three scenarios embed 25 panel pixels by adding panel pixels to background pixels via superimposition. In this case, a pixel which contains an embedded panel pixel also contains a background pixel and thus, its abundance is the sum of abundance of the embedded panel pixel and background pixel. The three counterparts of Scenarios TI1, TI2, and TI3 can also be simulated as TE1, TE2, and TE3, respectively. The design of TE serves different applications such as signal detection (see Chapter 2) and discrimination including subpixel target detection and discrimination, mixed pixel classification, and identification.

4.3.3.1 Scenario TE1 (Clean Panels Embedded in Clean Background)

Scenario TE1 is the same as Scenario TI1 except that the targets implanted into the background are now embedded into the background as shown in Figure 4.7.

More specifically, the targets are inserted into the background in such a manner that the targets are added to their corresponding background pixels without removing them similar to Scenario TI1. This scenario simulates an idealistic case for signal detection where two hypotheses represent background versus signal plus background, that is, H_0: BKG versus H_1: clean signal + BKG. As a result, there are no pure pixels in this particular scenario since pure pixels in the first and second columns are no longer pure due to its inclusion of background pixels. So, technically, there are no pure signatures or endmembers in the image scene, but there are 25 spectrally distinct panel signatures plus one BKG signature.

4.3.3.2 Scenario TE2 (Clean Panels Embedded in Noisy Background)

In analogy with TI2, Scenario TE2 simulates a practical signal detection problem where clean targets are embedded into a noisy background simulated by a background signature with an additive Gaussian noise to achieve an SNR = 20:1. In other words, instead of replacing background pixels with the clean targets as done in Scenario TI2, the targets are actually embedded into and superimposed over clean background pixels. So, in this case, the resulting synthetic image has clean targets embedded into a noisy background as shown in Figure 4.8.

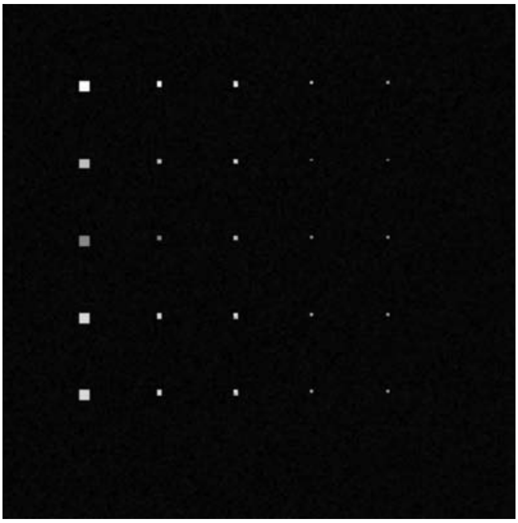


Figure 4.8 Synthetic image simulated by Scenario TE2.

This scenario simulates a case for target detection where two hypotheses represent noisy background against signal plus noisy background, that is, H_0: BKG + noise versus H_1: clean signal + BKG + noise. It is tricky to simulate this scenario since the noisy background pixels are replaced by their corresponding clean background pixels, while SNR must be retained at the given level.

4.3.3.3 Scenario TE3 (Gaussian Noise Added to Clean Panels Embedded in Background)

Scenario TE3 is the same as Scenario TE1 except that a Gaussian noise is added to Scenario TE1 to achieve SNR = 20:1 as shown in Figure 4.9.

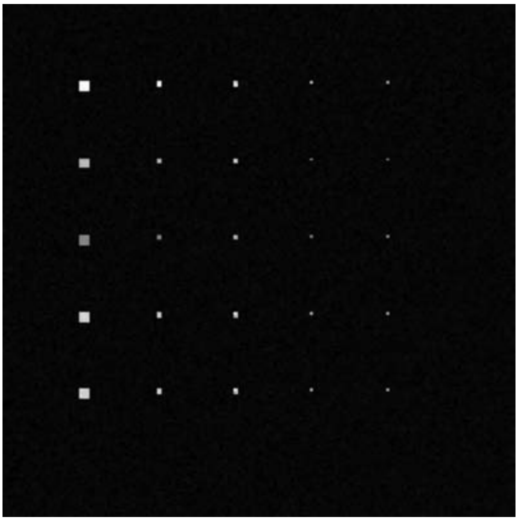


Figure 4.9 Synthetic image simulated by Scenario TE3.

It is also similar to Scenario TE2 with only difference that the embedded targets are noise corrupted compared to the clean targets embedded in Scenario TE2. More specifically, in Scenario TE3 the clean targets and clean background image are both corrupted by an additive Gaussian noise with SNR = 20:1. So, the 25 spectrally distinct panel signatures are corrupted by noise and BKG signature. This scenario simulates a case that two hypotheses represent noisy background against noisy signal with noisy background, i.e., H_0: noisy BKG versus H_1: noisy signal + noisy BKG.

Finally, several remarks on the above designed six synthetic images are noteworthy:

1. Most panel pixels in Figures 4.4–4.9 are visible. This may lead to a belief that these six scenarios are not useful or appropriate for experiments. The truth is that what we see from images is generally not what we will expect. Specifically, what we see is only qualitative and not quantitative, a task that a computer algorithm can do well while human being cannot. Although all target panels either implanted or embedded in synthetic images are clearly visible, it does not mean that an algorithm can detect these target panels well. This is exactly what we need from these scenarios to show that "can an algorithm accomplish what human eyes cannot do or do better?" Unfortunately, on many occasions visual assessment may even mislead conclusions. This phenomenon will be demonstrated in the following experiments where human eye inspection can only provide qualitative assessment but not quantitative measure. One such example was demonstrated in Chang and Wang (2008) and Figure 4.14 (Section 4.4.2.2), where a fully constrained least-squares (FCLS) method (Heinz and Chang, 2001; Chang, 2003a) could not estimate abundance fractions of any target panel in the second to fifth rows in Scenario TE2 even when the embedded target panels are clean and known precisely *a priori*. However, if we examine Scenario TE2 shown in Figure 4.8 closely, all the 130 embedded target panels are clearly visible by visual inspection. Why was FCLS unable to estimate abundance fractions of the embedded target panels correctly? The simple reason of why FCLS failed in Scenario TE2 is not that FCLS was ineffective but rather that the simulated embedded target panels in Scenario TE2 do not satisfy ASC imposed by FCLS. To resolve this dilemma, the nonnegativity constraint least squares (NCLS) method (Chang and Heinz, 2000; Chang, 2003a) was implemented. The NCLS-estimated abundance fractions of all 130 target panels in the TE2 scenario turned out to be very accurate as demonstrated by Chang and Wang (2008) and Figure 4.14 (Section 4.4.2.2). This simple example shows how unreliable human visual inspection can be. It also further explains why the three TE scenarios, which are simulated for various signal detection models involving pure, mixed, and subpixel targets, can be used to evaluate effectiveness of signal detection techniques such as NCLS (Chang and Heinz, 2001), while the three TI scenarios, which contain various simulated pure, mixed, and subpixel targets, can be used to evaluate effectiveness of endmember extraction algorithms such as FCLS used for accurate abundance fraction estimation (Heinz and Chang, 2001). The above six scenarios are precisely designed for these purposes.
2. The TI and TE scenarios introduced above bridge a gap between computer simulations such as a Mote Carlo method and real images and provide basic understanding of real images under complete controllable environments via a set of designed parameters. They can be further used to simulate more sophisticated scenarios such as two or more BKG signatures or different noise distributions or some examples in Chapter 18.
3. It should be noted that since different spectral bands have different signal energies, in order for each spectral band to achieve the same level of SNR defined as 50% signature (i.e., reflectance/radiance) divided by the standard deviation of the noise in Harsanyi and Chang (1994), zero-mean Gaussian noises with different variances are used and added to different bands for this purpose.

4. In hyperspectral imagery noise is generally non-Gaussian. This is mainly due to the fact that many unknown subtle substances such as clutters and interferers uncovered by hyperspectral imaging sensors are actually interference and not noise, in which case these unwanted interferers should be considered as structure noise instead of random noise. If all such unknown substances are removed in the image data, which is the case in these six scenarios, it leaves only random noise. Under this circumstance, the Gaussian noise is the most appropriate assumption, which is exactly the case assumed in signal processing and communications. In light of this interpretation, it is reasonable to simulate Gaussian for Scenarios TI and TE, because the simulated image background is clean.

4.4 Applications

The usefulness of the synthetic image-simulated six scenarios three applications are presented for illustration.

4.4.1 Endmember Extraction

Endmember extraction has received considerable interest in recent years and is probably one of the most important and crucial steps in hyperspectral image analysis since endmembers provide unique spectral information that is very valuable for data exploitation. Many algorithms have been developed and reported in the literature. Two most popular and widely used endmember extraction algorithms, pixel purity index (PPI) (Boardman, 1994) and N-finder algorithm (N-FINDR) (Winter, 1999a,b) with details in Chapter 7, were used for evaluation by the six designed scenarios. Since there are only five pure signatures, which are A, B, C, K, and M, dimensionality reduction required for PPI and N-FINDR was performed by the maximum noise fraction (MNF) transform (Green *et al.*, 1998) to reduce the original data space to five dimensions. The results produced by the PPI using 500 skewers and N-FINDR are shown in Figures 4.10 and 4.11, respectively, where all pixels with PPI counts greater than zero are shown and marked by yellow pixels. Since there is no noise in TI1 and TE1, PCA instead of MNF was performed for dimensionality reduction.

According to Figure 4.10, PPI was able to extract all five pure mineral signatures in all scenarios except TI1 and TE1. In particular, in TE1 PPI counts of all background pixels produced by the PPI were constant and greater than the PPI counts of the five pure mineral signatures because no noise was present in the data and the background dominates the entire data in which case it was considered as a pure signature. Similar results were also found by N-FINDR except one interesting finding which showed that N-FINDR could not extract the pure "calcite" signature in all TE scenarios. This is because the sample mean is used to simulate the image background and the signature of "calcite" is very close and similar to its signature in the sense of spectral similarity compared to other four mineral signatures. So, in this case, calcite was considered as a corrupted background signature so that once the background signature was extracted, the calcite could not be extracted. To see this, Figure 4.12 plots the spectral signatures of all the five minerals and the sample mean

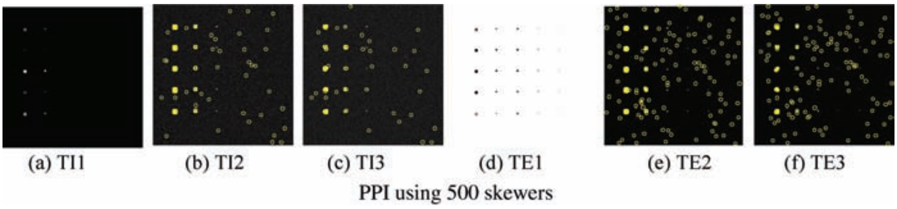


Figure 4.10 Endmember extraction by PPI.

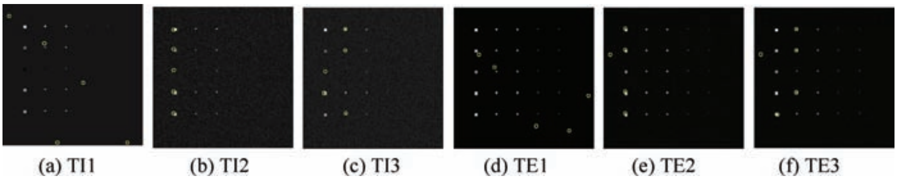


Figure 4.11 Five endmembers extracted by N-FINDR.

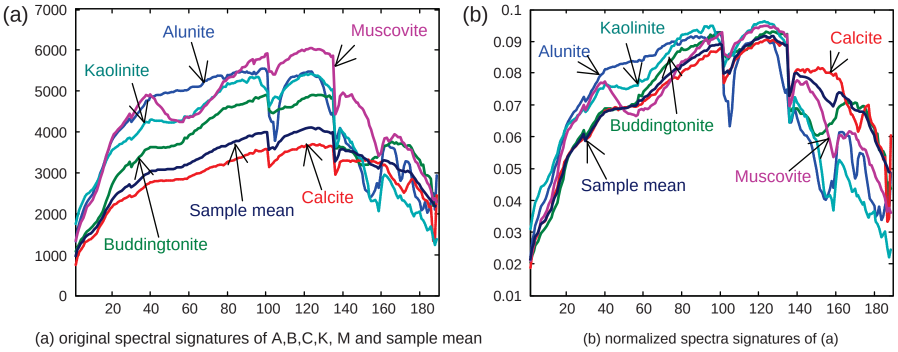


Figure 4.12 Spectra of A, B, C, K, M mineral signatures in the cuprite image scene and its sample mean signature.

and their normalized spectral signatures to show the similarity among their spectral shapes where the sample mean in Figure 4.12(a) and the calcite have nearly the same shapes from band 1 to band 140 in Figure 4.12(b).

The above experiments conducted based on Scenarios TI and TE also demonstrated several interesting results of how panel pixels extracted by PPI and N-FINDR in correspondence to five mineral signatures, which could not be observed by real image experiments. For example, N-FINDR successfully extracted all the panel pixels corresponding to five mineral signatures but in different manners where all the five extracted panel pixels in TI2 were from the first column compared to TI3 with two panel pixels from the first column and three panel pixels from the second column. A similar phenomenon was also observed in TE2 and TE3 except that N-FINDR successfully extracted the signature of calcite in TI2 and TI3 but failed to do so in Scenarios TE2 and TE3.

4.4.2 Linear Spectral Mixture Analysis (LSMA)

To perform LSMA, a linear mixing model is generally required where the complete target signature knowledge must be known *a priori*. It should be noted that in addition to the five endmembers discussed in Section 4.1, the background signature must be included in unmixing even when the background signature is mixed. This is because the background signature also represents a distinct spectral class in the data and cannot be excluded from being considered as an important signature to form the model. In this case, we assume that six distinct target signatures, which are five mineral signatures, A (alunite), B (buddingtonite), C (calcite), K (kaolinite), and M (muscovite) and a background signature, are the desired component signatures $\{\mathbf{m}_j\}_{j=1}^{6}$ to be used to form a linear mixing model $\mathbf{r} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n}$, where r is an image pixel, $\mathbf{M} = [\mathbf{m}_1\mathbf{m}_2 \cdots \mathbf{m}_6]$ is the target signature matrix with the abundance vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_6)^T$ specified by their corresponding abundance fractions $\{\alpha_j\}_{j=1}^{6}$, and $\mathbf{n}$ is a model correction term.

4.4.2.1 Mixed Pixel Classification

There are many mixed pixel classification methods available in the literature. One of the most widely used techniques is the so-called orthogonal subspace projection (OSP) developed by Harsanyi and Chang (1994) which has shown great success in various applications. Since OSP is developed as a signal detection technique and does not factor in abundance estimation, a least-squares OSP (LSOSP) was proposed by Tu *et al.* (1997) by including an abundance estimation error correction term in OSP as shown in Chang (1998). Figure 4.13 shows unmixed classification results by LSOSP along with their unmixed abundance fractions of each of the five mineral signatures where the labels of (a), (b), (c), (d), and (e) in quantification results corresponding to (A), (B), (C), (K), and (M) mineral signatures, respectively, and the LSOSP-unmixed abundance fractions were very close to true simulated abundance fractions for five mineral signatures.

4.4.2.2 Mixed Pixel Quantification

Despite that LSOSP has demonstrated its ability in unmixing abundance fractions as shown in Figure 4.13, it is an unconstrained spectral unmixing method which does not impose ASC $\sum_{j=1}^{6} \alpha_j = 1$ and abundance nonnegativity constraint $\alpha_j \geq 0$ for $1 \leq j \leq 6$. Consequently, the LSOSP-unmixed abundance fractions were not necessarily true fractions even though their unmixed fractions were more accurate than those produced by the OSP. So, for the purpose of mixed pixel quantification, these two constraints must be imposed on LSOSP. One such algorithm is the so-called FCLS method developed by Heinz and Chang (2001). Figures 4.14 graphically plot quantification results produced by FCLS for six scenarios where the labels of (a), (b), (c), (d), and (e) in quantification results corresponding to (A), (B), (C), (K), and (M) mineral signatures, respectively.

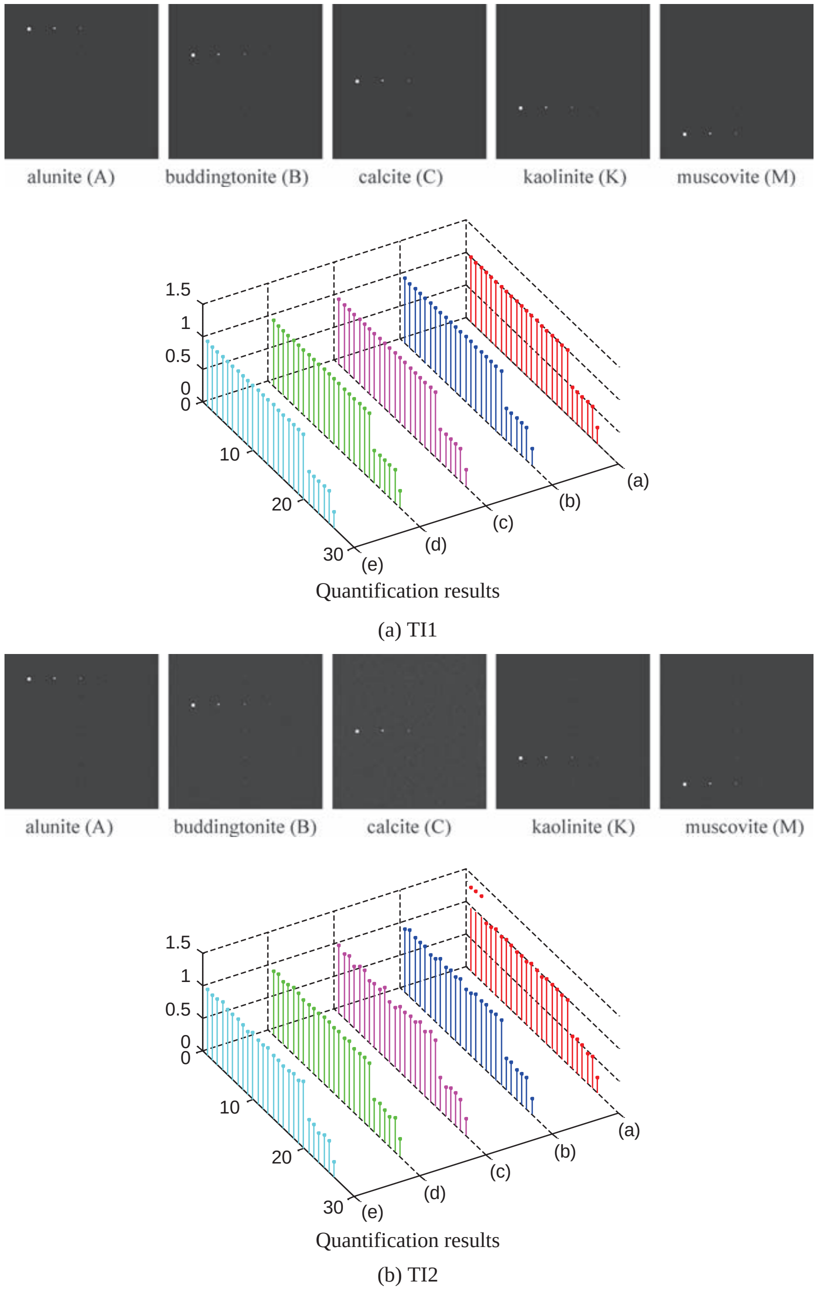
A very interesting and intriguing observation can be made from the results of three TE scenarios in Figure 4.14(c)–(f) where FCLS completely failed in quantifying all the five mineral signatures by throwing all abundance fractions to a single mineral signature, Muscovite. There is a reason for it. Since TE scenarios do not satisfy ASC, FCLS was forced to perform constrained quantification in which case it weighed all abundance fractions on the Muscovite due to the fact that the Muscovite has the most spectrally distinct signature among the five mineral signatures. This experiment demonstrated an important fact that constrained methods only worked effectively when the problems to be considered satisfy required constraints. These experiments further demonstrated the advantages of using synthetic images over real images.

4.4.3 Target Detection

Unlike the mixed pixel classification/quantification which requires complete knowledge of target signatures assumed to be in the data, the target detection only needs a certain level of partial target knowledge. In this section two types of target detection are considered, subpixel target detection which only needs the knowledge of the target signature of interest and anomaly detection which does not need any target knowledge.

4.4.3.1 Subpixel Target Detection

One of the most powerful subpixel target techniques is the constrained energy minimization (CEM) developed by Harsanyi (1993). Its various forms have been investigated in Chang (2002b). CEM only assumes that the target of interest is given and designated as the desired target signature, **d**, while discarding all other knowledge including background knowledge. By specifying one of the five mineral signatures as a desired target signature, **d**, Figure 4.15 shows the detection results of panel pixels that were simulated by the particular signature **d**.



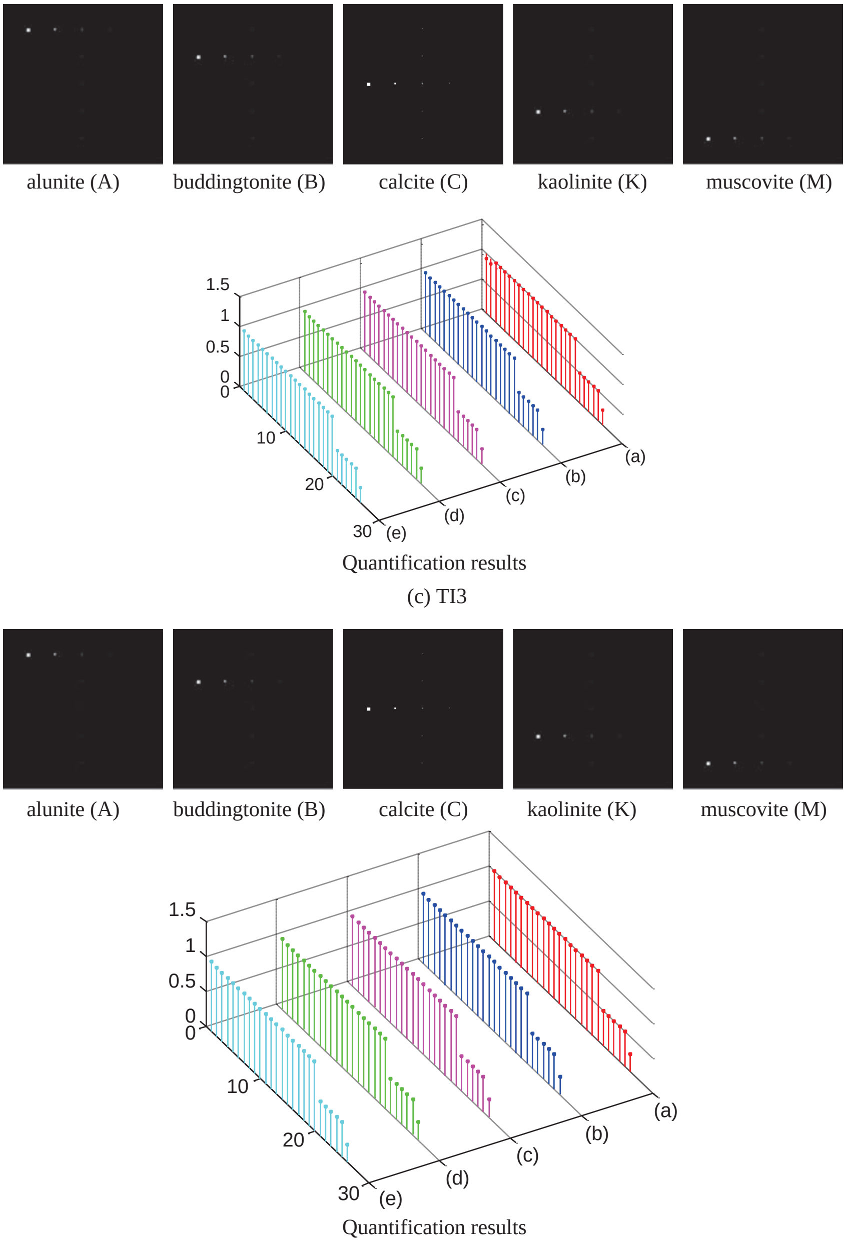
Quantification results

(a) TI1

Quantification results

(b) TI2

Figure 4.13 LSOSP-mixed pixel classification results for six scenarios.



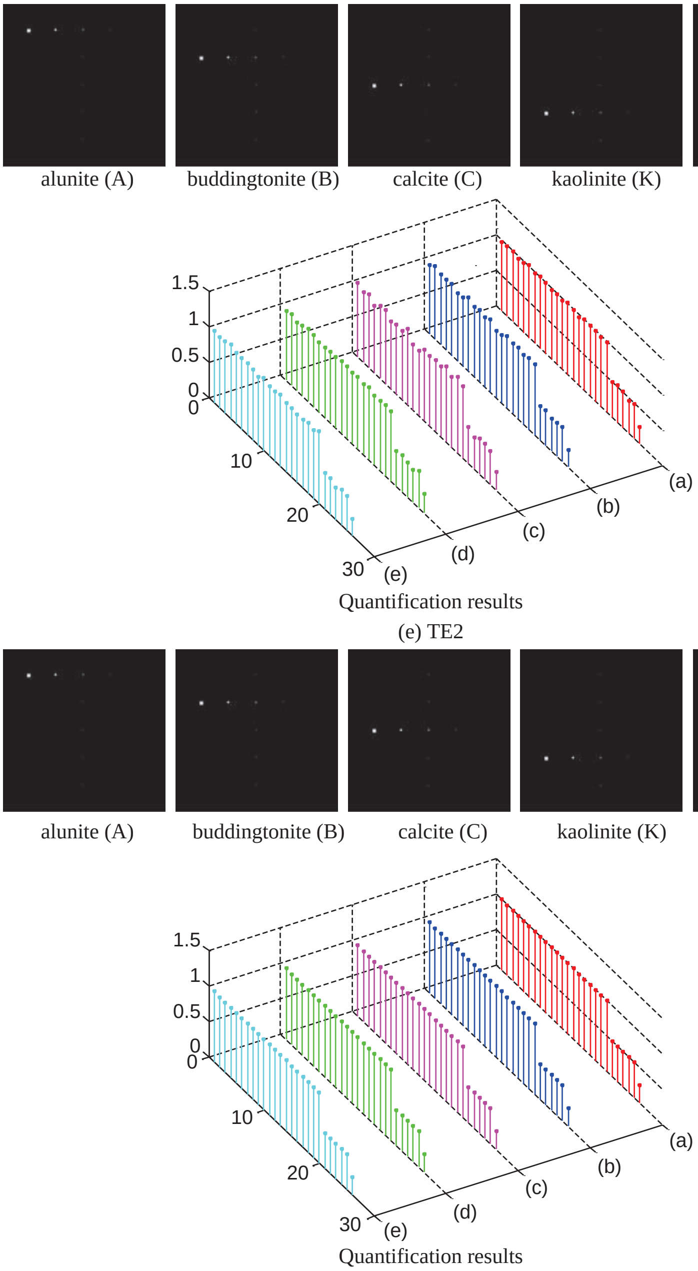
Quantification results

(c) TI3

Quantification results

(d) TE1

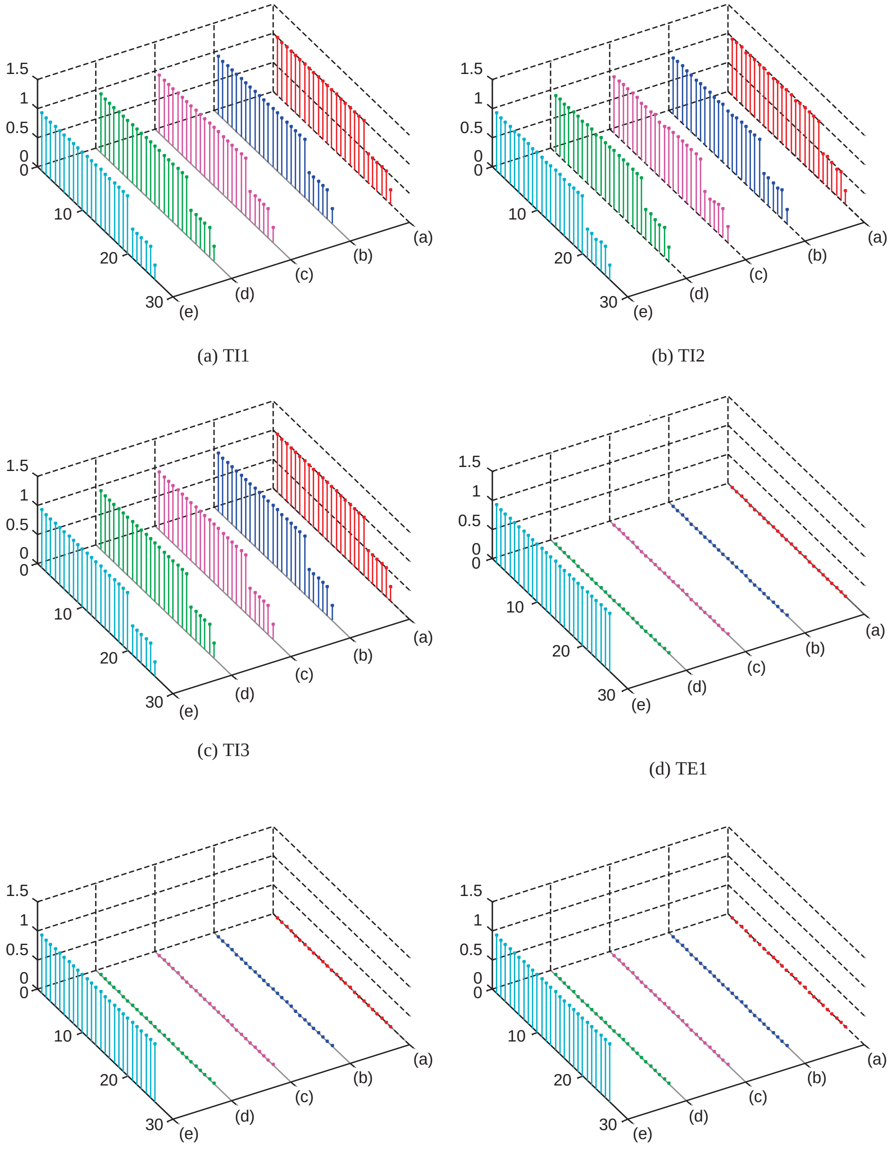
Figure 4.13 (*Continued*).



(e) TE2

(f) TE3

Figure 4.13 (*Continued*).



(a) TI1

(b) TI2

(c) TI3

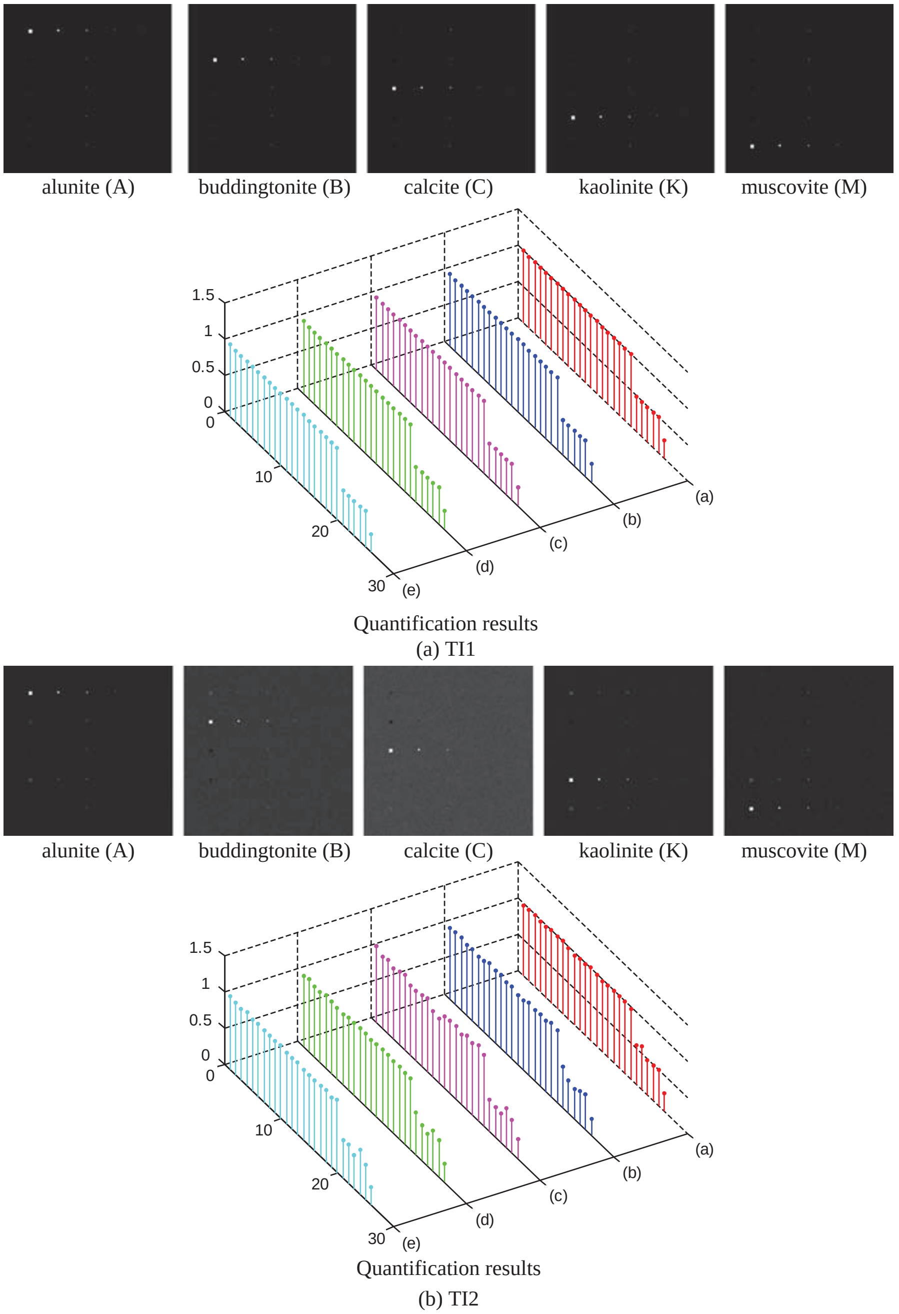
(d) TE1

(e) TE2

(f) TE3

Figure 4.14 FCLS-mixed pixel quantification results for six scenarios.

As we can see from the results in Figure 4.15, CEM also performed target detection very effectively including subpixel detection in the fourth and fifth columns. Most interestingly, CEM could detect small amounts of abundance fractions of other signatures simulated in mixed panel pixels, specifically panel pixels in the second row and the third column. Comparing the results in Figures 4.13 and 4.14, it can be clearly seen that LSMA made use of other signatures as unwanted signatures to suppress their interfering effects instead of detecting their abundance fractions as



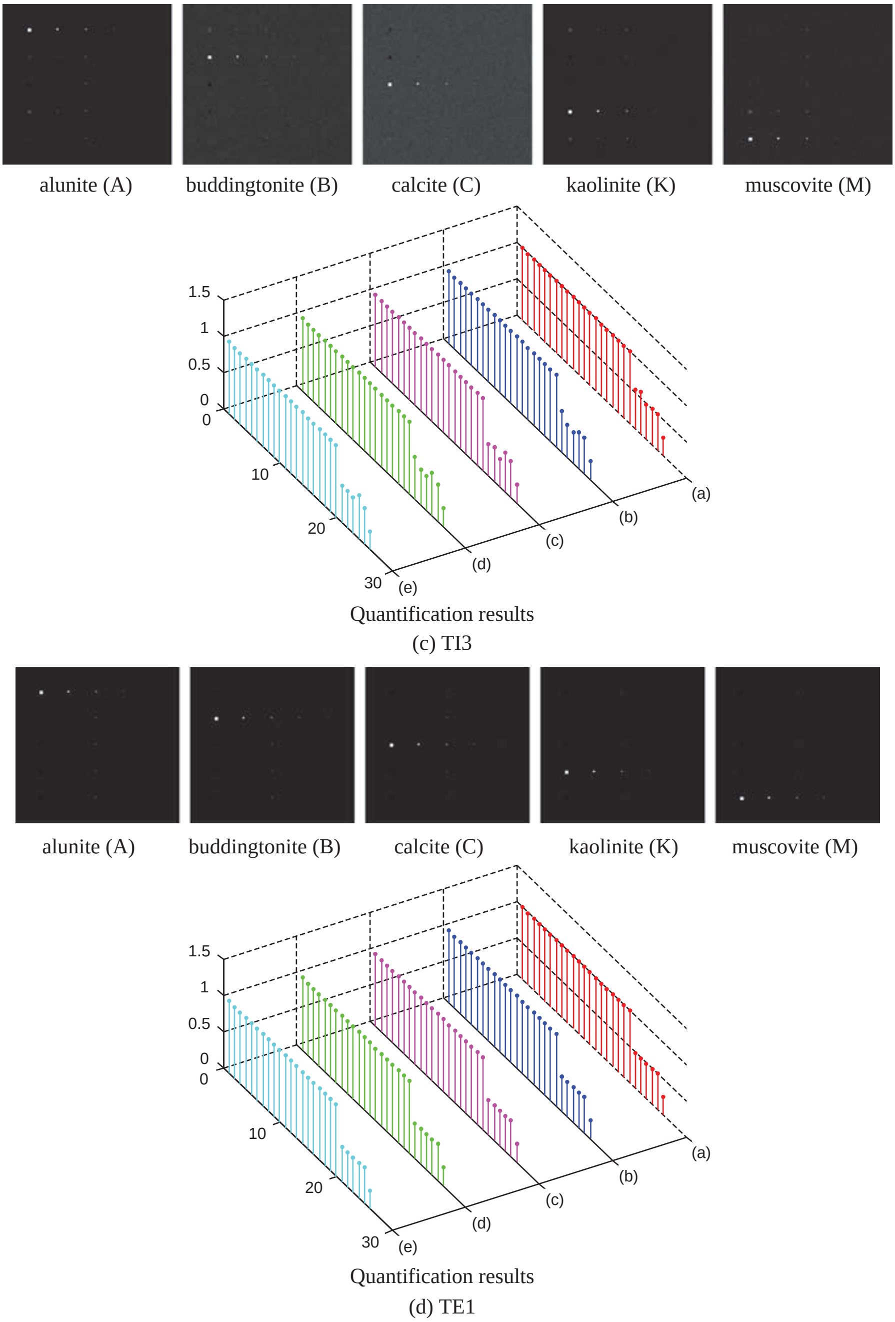
Quantification results

(a) TI1

Quantification results

(b) TI2

Figure 4.15 CEM detection results for six scenarios.



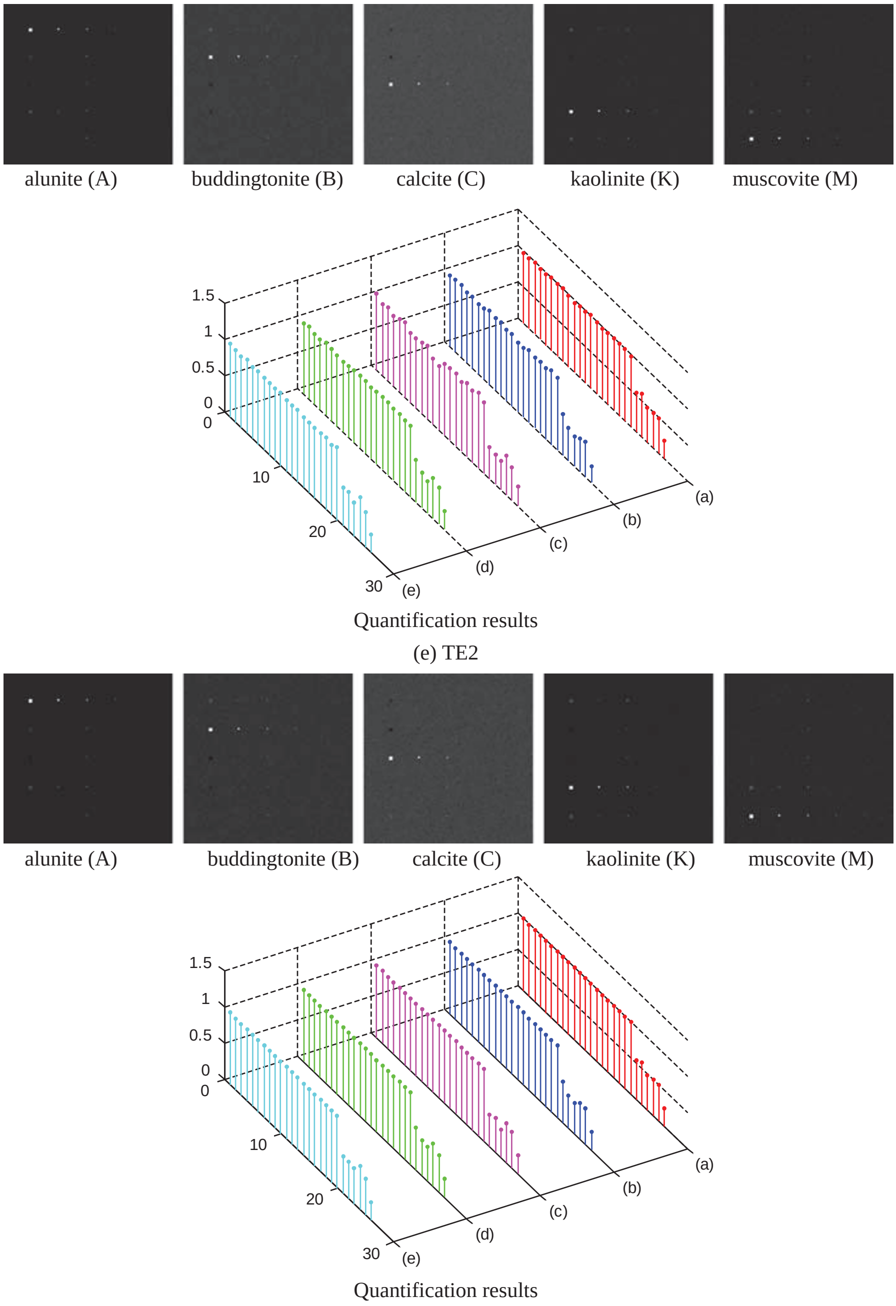
Quantification results

(c) TI3

Quantification results

(d) TE1

Figure 4.15 (*Continued*).



(e) TE2

(f) TE3

Figure 4.15 (*Continued*).

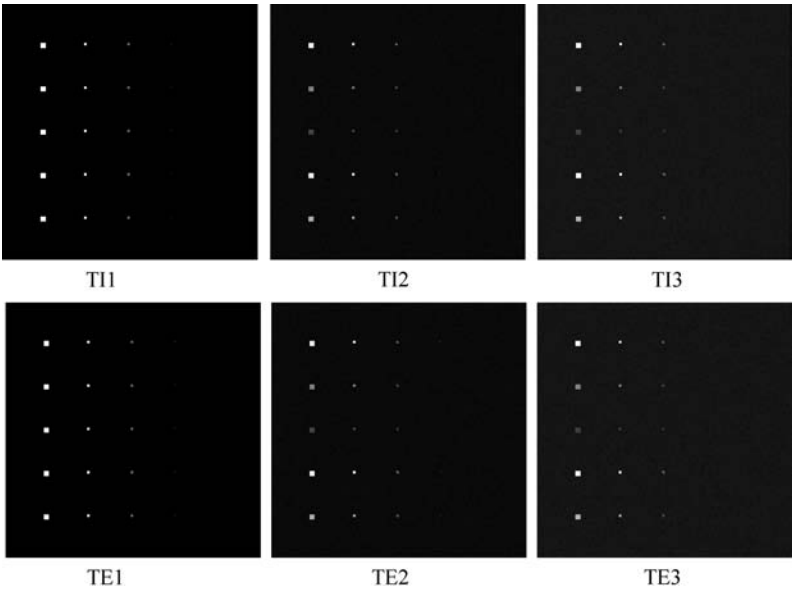


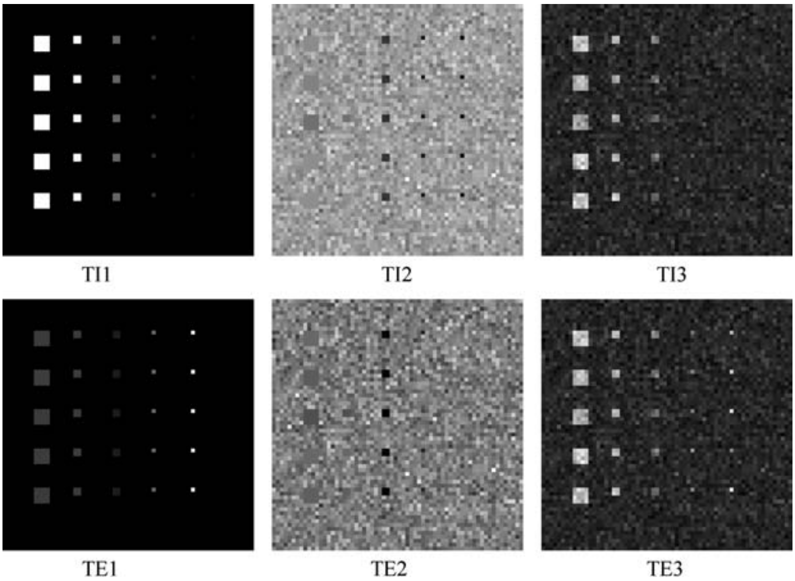
Figure 4.16 Anomaly detection by RXD for six scenarios with an image size of 200 × 200 pixel vectors.

CEM did. Obviously, real image experiments cannot provide such evidence because there is no complete prior endmember knowledge for verification.

4.4.3.2 Anomaly Detection

When CEM is implemented, it requires the specific knowledge of the desire target signature of interest (Chang, 2003a). In many applications such as surveillance there is no prior knowledge regarding which targets we are looking for and which targets we are interested in. In this case, target detection must be performed without appealing for any prior knowledge. One widely used detection algorithm is developed by Reed and Yu (1990), referred to as RXD. Figure 4.16 shows the results of the six scenarios produced by RXD which detects all panel pixels in the first three columns but misses all the subpixel panels in the fourth and fifth columns.

Now, if we operated RXD on the same six scenarios with a small image size of 64 × 64 pixel vector where the same 25 panels simulated in Figure 4.2 were also inserted into these six scenarios with the image background, Figure 4.17 shows their RXD-detected results. An immediate finding by comparing the results in Figure 4.17 to those in Figure 4.16 led to an interesting observation. That is, the target panels of sizes 2 × 2 and 1 × 1 that were detected in TI2 and TE2 by RXD in Figure 4.16 as anomalies now became undetectable in TI2 and TE2 for RXD as shown in Figure 4.17, in which case they were no longer considered as anomalies in Figure 4.17. Moreover, the performance of operating RXD on scenarios of TI and TE in Figure 4.16 was nearly the same and so was for scenarios of TI3 and TE3 in Figure 4.16. But this was not the case for RXD operating on a smaller image scene with the same identical 25 panels in Figure 4.17 where RXD had complete opposite results for TI1 and TE1 and quite different results for TI3 and TE3. Why did the same RXD produce different results for the same set of 25 panels inserted into the same image background with the only difference in the size of the processed image scenes? This simple example sheds light on the utility of the designed six scenarios which shows a tricky issue in anomaly detection, "what is really meant by anomaly?", a topic to be discussed in Chapter 18, Chang and Hsueh (2006) and Chang (2013).



TI1 TI2 TI3

TE1 TE2 TE3

Figure 4.17 Anomaly detection by RXD for six scenarios with an image size of 64 × 64 pixel vectors.

4.5 Conclusions

Many hyperspectral imaging algorithms have been designed and developed for data exploitation in the past. It seems that there is a lack of standardized data sets that can be used to objectively compare one algorithm to another for performance evaluation and analysis. In other words, if one claims his algorithm to be better than any other algorithm, without a standardized data set it will be very difficult to substantiate such a claim and validate the results. This chapter investigates this issue and further designs six scenarios that can be used as a standardized data set to simulate various scenarios. However, it should be noted that the six scenarios serve only as a purpose of how to deign synthetic images. Many other scenarios can also be simulated on the basis of the same concept such as those explained in Chapter 18. To illustrate how these scenarios can be carried out for algorithm performance analysis, three applications are included as illustrative examples, which are endmember extraction, spectral unmixing for mixed pixel classification/quantification, and subpixel target detection, each of which requires different levels of target signature knowledge.

5

Virtual Dimensionality of Hyperspectral Data

The term of virtual dimensionality (VD) was first coined in Chang (2003a) as a new concept defined as the number of spectrally distinct signatures in hyperspectral imagery. It was later published in Chang and Du (2004) and has received considerable interest since then. There are reasons of why VD has become a widespread and acceptable concept in hyperspectral imaging community. First, due to significantly improved spectral and spatial resolutions a hyperspectral image sensor can now uncover many unknown subtle material substances, referred to as signal sources that cannot be identified by *a priori* knowledge or visual inspection. Determining the number of such substances in the data is very challenging and extremely difficult, if not impossible. Second, there exists no concept in hyperspectral imaging similar to intrinsic dimensionality (ID) (Fukunaga, 1982, 1990) used in statistical signal processing, pattern recognition, and classification that can be used for hyperspectral data exploitation where ID, also known as effective dimensionality (ED), is defined as the minimal number of parameters used to characterize data. Third, the techniques developed for determining ID in multivariate data have been shown to be inapplicable to hyperspectral data. Finally and most importantly, before VD was introduced, there were few techniques available in the literature that can be used to effectively determine the number of unknown signal sources in hyperspectral data. VD was originally proposed to address these issues. Like many other new concepts VD has evolved through several stages where it has been examined and studied extensively for its use in various applications. As expected, some controversial issues also arise in how VD is used and interpreted. This chapter revisits and reexamines the concept of VD and further explores its utility in hyperspectral data exploitation, while clarifying some issues that may have misled users to misinterpreting VD.

5.1 Introduction

How to represent multidimensional data in an appropriate form in some sense of optimality is very challenging. There are two key issues involved. One is how to determine the number of basic elements, referred to as p, present in the data. The other is how to find these p basic elements. While these two issues may be addressed separately, they can also be treated as one issue if they are tied together in a certain form. This chapter investigates VD from these two aspects.

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

The VD proposed in Chang (2003a) is essentially developed from the first aspect without constructing the basic elements. According to the definition provided in Fukunaga (1990) for ID, it is defined as the minimal number of parameters required to account for the observed properties of the data that must be specifically defined. As a matter of fact, the properties observed in the data are generally determined by these basic elements that vary with applications. In other words, when high-dimensional data sets are observed, the data sample vectors are generally represented in a certain form of dimensionality. Most commonly it is the dimensionality of a data sample vector, referred to as data dimensionality, which is defined by the number of coordinates used to represent the data sample vector. As an alternative, in many applications in order for a data set to be represented more effectively, the original data dimensions are usually transformed to components via a transformation where each data component is a result of a transformed data dimension. The resulting dimensionality is referred to as component dimensionality. For example, in color image processing the red (R), green (G), and blue (B) are used to represent an image in color space where the data dimensions are specified by three colors, R,G,B. However, the representation by the (R,G,B) coordinate is not well suited for describing colors in terms of practical or human interpretation. In this case, a more effective way to represent a color image is to transform the R-G-B model to a new data coordinate system, called Hue (H), Saturation (S), and Intensity (I) as new data dimensions. Similarly, a data set can also be represented by new data components via the principal components analysis (PCA) where each original data dimension is transformed to a component specified by a data variance. In remote sensing data an array of sensors specified by a range of wavelengths has been used for data collection in which case a data dimension acquired by each spectral channel is called a spectral band dimension. As a result, the spectral dimensionality of a remotely sensed data set is determined by the number of spectral bands or channels used for data acquisition. So, as long as data representation is concerned, various types of data coordinate systems can be used for this purpose. How well a data representation can characterize data-observed properties is determined by an appropriate selection of a data representation system. Unfortunately, ID only provides a very abstract notion and does not specify the parameters to be used for data characterization. As a consequence, ID is practically not useful. This is because ID decouples the issue in determining the number of parameters from the issue of what parameters are used to account for data-observed properties where both issues are indeed closely tied together and must be treated as one single issue.

VD is defined in Chang (2003a) as well as Chang and Du (2004) as the number of spectrally distinct signatures in hyperspectral data without specifying these signatures. It was originally developed along the same line as ID is defined. But it has three salient differences from ID. Firstly, unlike ID that does not specify data properties, VD is particularly designed for hyperspectral data by data spectral properties characterized by so-called nonliteral information. Secondly, while ID requires the number of parameters to be minimal, the number of spectrally distinct signatures defined in VD is not necessarily minimal. This number can vary and is actually determined by a particular application. For example, the number of endmembers is different from the number of anomalies. However, when it comes to data representation such as linear spectral mixture analysis (LSMA), VD can be considered as the minimal number of spectral signatures required to form a linear model to perform linear spectral unmixing, in which case such signatures are referred to as virtue endmembers (VEs) that are the basic elements required to form a linear mixing model used by LSMA. Thirdly, even for a given particular application VD may also vary with a specific designed technique. Unfortunately, being unaware of these differences some users have misinterpreted the use of VD as a one-size-fits-all definition for all applications. So, when VD does not work for a particular application, VD takes the blame. In fact, the technique used to find VD should be the one that takes responsibility. In order to further explore the concept of VD the following section revisits and reinterprets VD in terms of a two-category dichotomy, data characterization and data representation.

5.2 Reinterpretation of VD

Since VD was introduced in Chang (2003a), it has shown promise in various applications, just to name a few, dimensionality reduction (Wang and Chang, 2006), band selection (Chang and Wang, 2006), and endmember extraction (Nascimento and Dias, 2005; Chang and Plaza, 2006; Chang et al. 2006). However, it also gives rise to some controversial issues caused by users' misinterpretation of VD. The first misinterpretation of VD is to tie VD to the technique that was originally developed by Harsanyi et al. (1994a), referred to as Harsanyi–Farrand–Chang (HFC) method for VD estimation. When the HFC method does not perform effectively, users blame VD for its inapplicability. A second misinterpretation is caused by the fact that VD does not address the second issue mentioned in the introduction, that is, how to find the p basic elements as a whole. More specifically, VD must be tuned to various applications that define basic elements. In other words, a different application may need a different set of basic elements in which case a different value of p is also required. A third misinterpretation results from a misconception that the techniques developed for finding ID should also be applicable to finding VD. Unfortunately, this is not true. Since ID does not specify any data properties, a default assumption about the data properties for ID is data variances in which case PCA becomes a classical approach to determine ID. Unlike ID, VD is specifically designed to preserve the information provided by spectral dimensions, particularly signal sources characterized by "*spectral*" not "*spatial*" properties. In this case, PCA that finds its applicability for ID is no longer effective for VD as demonstrated in Chang (2003a) and Chang and Du (2004). To address this issue, a new technique called the HFC method, designed by Harsanyi *et al.* (1994a), has been particularly developed. Finally, the most common misinterpretation is to tie VD together with the HFC method and consider the HFC method as the only technique used to determine VD. So, when the HFC method does not work, users promptly conclude that VD is wrongly defined. This is similar to what users believe that PCA is the only method that can be used to determine ID. When PCA does not work, users then complain that ID is incorrectly defined. Apparently, this is a serous misinterpretation. The matter of truth is that ID and VD only define the number of parameters and the number of signal sources in the data, but do not define what exactly these parameters and signal sources are. How to find these parameters or signal sources is another issue. Accordingly, what the HFC method is to VD is exactly the same as what PCA is to ID. The PCA and HFC methods are only one of many methods that can be used to determine ID and VD, respectively, but not the only method to be used for this purpose. If one method does not work effectively, a new method should be sought. This has nothing to do with definitions.

This chapter explores the aforementioned issues and further proposes two aspects to determine VD. The first one is based on characterization of data-observed properties regardless of what basic elements are. This is similar to that used to define ID. The second one is based on data representation, which treats determination of VD as a part of finding appropriate basic elements to represent the data to be processed in some sense of optimality such as linear representation. In this case, the issue of determining VD is tied with the issue of finding an optimal set of basic elements for data representation as one single issue. In what follows, each of these two aspects is discussed in detail.

5.3 VD Determined by Data Characterization-Driven Criteria

Many criteria have been reported in the literature to estimate the number of signal sources from various perspectives. This section briefly reviews criteria that use characterization of data properties to determine VD. The first criterion, referred to as the eigenvalue distribution criterion, calculates the cumulative eigenvalues to account for the energy contributed by signals sources, which is determined by a given error threshold ε. This criterion involves finding eigenvalues of a sample data covariance/correlation matrix. A second criterion is also eigenbased component analysis that includes singular value decomposition (SVD) and PCA, both of which find a smallest singular value or eigenvalue bounded below from a given error threshold ε. A third criterion is referred to as factor analysis (FA)-based Malinowski's error theory where four measures, real error (RE), extracted error (XE), imbedded error (IE), and empirical indicator function (EIF), are used to implement the FA criterion. A fourth criterion is information theoretic criterion (ITC) that includes an information criterion (AIC) and minimum description length (MDL), both of which are developed based on the logarithm of likelihood functions. A fifth criterion is a Gershgorin radius that is developed by separating the Gershgorin disks formed by Gershgorin radii into two classes, signal class and noise class. The error threshold ε is used to determine how well these two classes will be separated. A sixth criterion makes use the Neyman–Pearson (NP) detection theory to estimate the number of signal sources.

5.3.1 Eigenvalue Distribution-Based Criteria

Before we proceed, some necessary notations need to be defined. Let $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N\}$ be a set of N data sample vectors where $\mathbf{r}_i = (r_{i1}, r_{i2}, \ldots, r_{iL})^T$ is an L-dimensional vector with $1 \leq i \leq N$. Assume that $\boldsymbol{\mu} = (1/N)\sum_{i=1}^{N} \mathbf{r}_i$ is the sample data global mean vector, and $\mathbf{K}_{L\times L}$ is the sample data covariance matrix formed by $\mathbf{K}_{L\times L} = (1/N)\sum_{i=1}^{N} (\mathbf{r}_i - \boldsymbol{\mu})(\mathbf{r}_i - \boldsymbol{\mu})^T$. Suppose that $\{\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L\}$ are the eigenvalues of $\mathbf{K}_{L\times L}$ arranged in descending order and $\{\mathbf{v}_1 \geq, \mathbf{v}_2 \geq, \cdots \geq, \mathbf{v}_L\}$ are their associated orthonormal eigenvectors. Similarly, let $\mathbf{R}_{L\times L}$ be the sample data correlation matrix formed by $\mathbf{R}_{L\times L} = (1/N)\sum_{i=1}^{N} \mathbf{r}_i\mathbf{r}_i^T$ with its eigenvalues specified by $\{\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_L\}$ in descending order and their corresponding orthonormal eigenvectors $\{\hat{\mathbf{v}}_1 \geq, \hat{\mathbf{v}}_2 \geq, \cdots \geq, \hat{\mathbf{v}}_L\}$.

The ID definition neither specifies what observed data properties need to be characterized nor provides a means of how ID is determined. So, a general approach to finding ID is PCA where eigenvalues are used to determine the number of principal components, which is the value of p. In this case, the observed data properties characterized by ID are data variances specified by eigenvalues. There are usually two ways to find ID using eigenvalues. One is to plot all eigenvalues in descending order and find a sudden drop at a certain eigenvalue that determines ID. The other is to calculate the ratio of a cumulative sum of descending eigenvalues to the total sum of eigenvalues and the ratio is then used to determine ID. So, a straightforward extension to VD is to follow similar approaches to determining VD.

One simplest and commonly used criterion is to use the sum of eigenvalues of $\mathbf{K}_{L\times L}$ or $\mathbf{R}_{L\times L}$ to estimate the number of signal sources. Three measures can be derived for this purpose. In order to make the error threshold for different measures in the same range for comparison, the covariance eigenvalues $\{\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L\}$ and correlation eigenvalues $\{\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_L\}$ are normalized by $\tilde{\lambda}_l = \frac{\lambda_l}{\sum_{l=1}^{L}\lambda_l}$ and $\tilde{\hat{\lambda}}_l = \frac{\hat{\lambda}_l}{\sum_{l=1}^{L}\hat{\lambda}_l}$ for $1 \leq l \leq L$ as probability vectors, respectively.

5.3.1.1 Thresholding Energy Percentage

One is to calculate the cumulative sum of eigenvalues to represent how much energy is contributed by signal sources. In other words, the cumulative distribution resulting from probability vectors $[\tilde{\lambda}_1, \tilde{\lambda}_2, \ldots, \tilde{\lambda}_L]$ defined above is used to specifiy how much energy percentage

$a\%$ is contributed by signal sources. In this case, VD is determined by finding the smallest number p that yields

$$\mathrm{VD}_{\hat{\lambda}}^{\mathrm{eigen}}(a\%) = \arg\left\{\min_{1\leq l\leq L}\left[\sum\nolimits_{l=1}^{p} \tilde{\lambda}_l \geq \frac{a}{100}\right]\right\} \text{ for a given percentage } a\% \tag{5.1}$$

5.3.1.2 Thresholding Difference between Normalized Correlation Eigenvalues and Normalized Covariance Eigenvalues

Analogous to (5.1) we can also define a meausre by calculating the difference between two corresponding normalized covariance and correlation eigenvalues, $\tilde{\hat{\lambda}}_l - \tilde{\lambda}_l$, as follows:

$$\mathrm{VD}_{\hat{\lambda}-\lambda}^{\mathrm{eigen}}(\varepsilon) = \arg\left\{\max_{1\leq l\leq L}\left[\tilde{\hat{\lambda}}_l - \tilde{\lambda}_l \geq \varepsilon\right]\right\} \text{ for a given threshold } \varepsilon \tag{5.2}$$

5.3.1.3 Finding First Sudden Drop in the Normalized Eigenvalue Distribution

An alternative to (5.2) is to calculate the difference between two consecutuve normalized covariance eigenvalues, $\tilde{\lambda}_{l-1} - \tilde{\lambda}_l$, or normalized correlation eigenvales $\tilde{\hat{\lambda}}_{l-1} - \tilde{\hat{\lambda}}_l$, by

$$\mathrm{VD}_{\lambda}^{\mathrm{eigen}}(\varepsilon) = \arg\left\{\max_{2\leq l\leq L}\left[\tilde{\lambda}_{l-1} - \tilde{\lambda}_l \geq \varepsilon\right]\right\} \text{ for a given threshold } \varepsilon \tag{5.3}$$

or

$$\mathrm{VD}_{\hat{\lambda}}^{\mathrm{eigen}}(\varepsilon) = \arg\left\{\max_{1\leq l\leq L-1}\left[\tilde{\hat{\lambda}}_{l-1} - \tilde{\hat{\lambda}}_l \geq \varepsilon\right]\right\} \text{ for a given threshold } \varepsilon \tag{5.4}$$

5.3.2 *Eigen-Based Component Analysis Criteria*

Another commonly used approach is to find eigen-components via eigenvalue decomposition. Two such techniques are discussed.

5.3.2.1 Singular Value Decomposition (SVD)

SVD is one of most important signal processing techniques. Assume that the matrix $\mathbf{B}_{L\times L} = \mathbf{A}_{L\times N}(\mathbf{A}_{L\times N})^T$ where $\mathbf{A}_{L\times N} = [\mathbf{r}_1\mathbf{r}_2\cdots\mathbf{r}_N]$ is a data matrix formed by N data sample vectors $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N\}$. The singular values of the matrix $\mathbf{B}_{L\times L}$ can be found and arranged in the following descending order:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q > 0 = \lambda_{q+1} = \lambda_{q+2} = \cdots = \lambda_L \tag{5.5}$$

$$\sqrt{\lambda_1} \geq \sqrt{\lambda_2} \geq \cdots \geq \sqrt{\lambda_q} > 0 = \sqrt{\lambda_{q+1}} = \sqrt{\lambda_{q+2}} = \cdots = \sqrt{\lambda_r} \tag{5.6}$$

where $\sqrt{\lambda_1} \geq \sqrt{\lambda_2} \geq \cdots \geq \sqrt{\lambda_q}$ in (5.6) turns out to be the square root of nonnegative eigenvalues of $\mathbf{K}_{L\times L}$ or $\mathbf{R}_{L\times L}$. If we interpret eigenvalues as variances, the singular values are simply their standard deviations. Detailed discussion on SVD can be found in Section 6.2.1.3 of Chapter 6.

According to (5.6) it is the value of p that corresponds to VD needed to be estimated and defined as

$$\mathrm{VD}_{\lambda}^{\mathrm{SVD}}(\varepsilon) = \arg\left\{\max_{1\le l\le L}\left[\sqrt{\tilde{\lambda}_l} \ge \varepsilon\right]\right\} \text{ for a given threshold } \varepsilon \tag{5.7}$$

5.3.2.2 Principal Components Analysis (PCA)

A similar and very popular component transform to SVD is PCA. Assume that $\{\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_L\}$ is a set of eigenvalues arranged in descending order. We calculate the ratio of a cumulative sum of the eigenvalues in descending order to the total sum of eigenvalues, $R_\lambda(p) = \left(\sum_{j=1}^{p}\lambda_j\right)/\left(\sum_{j=1}^{L}\lambda_j\right)$, and determine VD by setting a prescribed percentage $a\%$, that is,

$$\mathrm{VD}^{\mathrm{PCA}}(a\%) = \arg\{\min_p[R_\lambda(p) \ge a/100]\} \tag{5.8}$$

In this case, selecting an appropriate $a\%$ in (5.8) as a cutting threshold is also challenging.

5.3.3 Factor Analysis: Malinowski's Error Theory

Malinowski's error theory (Malinowski, 1977a, 1977b) is developed based on FA, which is a mathematical technique developed for solving multidimensional problems by expressing a data sample as a linear sum of a finite number of product terms referred to as factors. It has been widely used in various fields of chemistry such as chromatography, spectrophotometry, nuclear magnetic resonance, and mass spectroscopy (Malinowski, 1977a, 1977b). Therefore, a factor used in FA is similar to a principal component used in PCA. So, like PCA, FA also finds eigenvalues of a sample covariance matrix and groups all eigenvalues into two categories called primary set of eigenvalues and secondary set of eigenvalues in accordance with three types of errors illustrated in Figure 5.1: RE resulting from the difference between the pure and raw data, XE resulting from the difference between the FA-reproduced data and raw data, and IE resulting from the difference between the pure data and FA-reproduced data, each of which can be defined as follows:

$$\mathrm{RE}(p) = \left[\frac{\sum_{l=p+1}^{L}\lambda_l}{N(L-p)}\right]^{1/2} \tag{5.9}$$

$$\mathrm{IE}(p) = \left[\frac{p\sum_{l=p+1}^{L}\lambda_l}{LN(L-p)}\right]^{1/2} \tag{5.10}$$

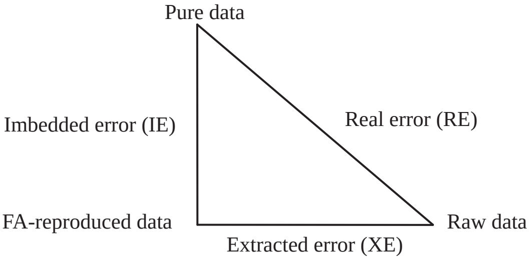


Figure 5.1 Relationship among three types of errors.

$$\mathrm{XE}(p) = \left[\frac{\sum_{l=p+1}^{L}\lambda_l}{LN}\right]^{1/2} \tag{5.11}$$

where L is the total number of spectral bands and N is the total number of data samples with $L \leq N$.

While the primary set consists of eigenvalues corresponding to RE and IE that cannot be removed by any technique, the secondary set is made up of XE that can be removed by a custom-designed mathematical technique for data improvement. So, if we interpret pure data, raw data, and FA-reproduced data as reflectance data, real radiance data, and processed data, respectively, endmembers represent pure signatures compared to data samples that are real signatures in the data, while the processed pixels are pixels used for data representation. In light of this interpretation, (5.9)–(5.11) can be used as criteria to find an optimal number of factors as follows:

$$\mathrm{VD}_{\mathrm{RE}}^{\mathrm{FA}} = \arg\left\{\min_{1\leq p\leq L}[\mathrm{RE}(p)]\right\} = \arg\left\{\min_{1\leq p\leq L}\left(\left[\frac{\sum_{l=p+1}^{L}\lambda_l}{N(L-p)}\right]^{1/2}\right)\right\} \tag{5.12}$$

$$\mathrm{VD}_{\mathrm{IE}}^{\mathrm{FA}} = \arg\left\{\min_{1\leq p\leq L}[\mathrm{IE}(p))]\right\} = \arg\left\{\min_{1\leq p\leq L}\left(\left[\frac{p\sum_{l=p+1}^{L}\lambda_l}{LN(L-p)}\right]^{1/2}\right)\right\} \tag{5.13}$$

$$\mathrm{VD}_{\mathrm{XE}}^{\mathrm{FA}} = \arg\left\{\min_{1\leq p\leq L}[\mathrm{XE}(p))]\right\} = \arg\left\{\min_{1\leq p\leq L}\left(\left[\frac{\sum_{l=p+1}^{L}\lambda_l}{LN}\right]^{1/2}\right)\right\} \tag{5.14}$$

Another criterion different from (5.12)–(5.14) is the EIF derived by Malinowski (1977b) and defined as

$$\mathrm{EIF}(p) = (1/(L-p))^2\left[\frac{\sum_{l=p+1}^{L}\lambda_l}{N(L-p)}\right]^{1/2} \tag{5.15}$$

The optimal number of factors to produce the smallest EIF(p) in (5.15) is the one solving the following equation:

$$\mathrm{VD}_{\mathrm{EIF}}^{\mathrm{FA}} = \arg\left\{\min_{1\leq p\leq L}[\mathrm{EIF}(p)]\right\} = \arg\left\{\min_{1\leq p\leq L}\left[\frac{\sum_{l=p+1}^{L}\lambda_l}{N(L-p)^5}\right]^{1/2}\right\} \tag{5.16}$$

5.3.4 Information Theoretic Criteria (ITC)

A similar issue is also encountered in statistical signal processing, specifically in passive array processing where determination of the number of signal sources has been a challenging problem. Several methods have been proposed in the past for this purpose. Two most widely used criteria to estimate the number of signal sources arriving at an array of sensors are AIC suggested by Akaike (1974) and MDL proposed by Schwarz (1978) and Rissanen (1978), which can be used for model selection. They can be derived from information theoretic criteria. The formulas given below for AIC and MDL are obtained by Wax and Kailath (1985).

5.3.4.1 AIC

$$\begin{aligned} \text{AIC}(p) = & -2\log\left(\prod_{j=p+1}^{L} \lambda_j^{1/(L-p)} \Big/ \left[(1/(L-p)) \sum_{j=p+1}^{L} \lambda_j\right]\right)^{(L-p)/N} \\ & +2p(2L-p) \end{aligned} \tag{5.17}$$

where p is the number of free parameters that specifies a family of probability density functions and $\{\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_L\}$ are eigenvalues generated by the sample covariance matrix $\mathbf{K}_{L\times L}$:

$$\text{VD}_{\text{AIC}}^{\text{ITC}} = \arg\left\{\min_{1\leq p\leq L}\text{AIC}(p)\right\} \tag{5.18}$$

5.3.4.2 MDL

$$\begin{aligned} \text{MDL}(p) = & -\log\left(\prod_{j=p+1}^{L} \lambda_j^{1/(L-p)} \Big/ \left[(1/(L-p)) \sum_{j=p+1}^{L} \lambda_j\right]\right)^{(L-p)/N} \\ & +(1/2)p(2L-p)\log N \end{aligned} \tag{5.19}$$

where p is the number of signal sources determined by finding the smallest number of p that solves the following optimization problem:

$$\text{VD}_{\text{MDL}}^{\text{ITC}} = \arg\left\{\min_{1\leq p\leq L}\text{MDL}(p)\right\} \tag{5.20}$$

It is worth noting that there are two underlying assumptions on these two criteria. The noise is assumed to be (1) independent identically distributed (i.i.d.) and (2) Gaussian.

Intuitively, it seems that AIC and MDL can be directly applied to estimation of VD. The truth is that the concept of VD is more sophisticated than just a simple estimate considered in array processing. Such implication may be too naïve since the issue in determining the number of signal sources of interest in hyperspectral imagery is much more complicated than that in array processing. Firstly, in passive sensor array processing the only issue is to separate signal sources of interest from the noise and there is no need of differentiating one signal source from another. In hyperspectral data, signal sources of interest vary with different applications. Signal source distinction is crucial in hyperspectral data analysis. Secondly, the number of signal sources estimated in array processing is a fixed value for data to be processed regardless of what types of signal sources are of interest in different applications. However, different types of signal sources play a key role in hyperspectral data exploitation. However, the AIC- and MDL-estimated values are fixed at a constant. After all its only interest is determination of signal sources arriving at an array of passive sensors. These two issues are indeed crucial in developing the concept of VD because the number of signal sources of interest must vary and be determined by applications but not by a single value as a constant. In this case, the criteria used in array processing may not be applicable to hyperspectral imagery.

5.3.5 Gershgorin Radius-Based Methods

One of major disadvantages of using AIC and MDL is their assumption of Gaussian noise. It is well known that the noise in remotely sensed imagery is generally not Gaussian. In order to alleviate this dilemma, this section presents another criterion that only assumes that the noise is i.i.d., but not necessarily Gaussian. The idea is to develop a method that can find locations of eigenvalues of the sample covariance matrix so that all the eigenvalue can be separated into two classes in accordance with their locations: signal class and noise class. By counting the number of eigenvalues in the signal class we can estimate the number of signal sources in the data. The idea is briefly described as follows.

Assume a complex $L \times L$ matrix $\mathbf{A} = \lfloor a_{ij} \rfloor$ with the (i, j) element denoted by a_{ij}. For each row i, $1 \leq i \leq L$, we define a parameter ρ_i by

$$\rho_i = \sum\nolimits_{j=1, j \neq i}^{L} |a_{ij}| \tag{5.21}$$

Then the so-called Gershgorin disk $R_i(\mathbf{A})$ is defined by

$$R_i(\mathbf{A}) = \{z \in C | \rho_i \geq |z - a_{ii}|\} \tag{5.22}$$

Gershgorin Circle Theorem (Moon and Stirling, 2000, p. 325)
The eigenvalues of an $L \times L$ matrix $\mathbf{A} = \lfloor a_{ij} \rfloor$ all lie in the union of the Gershgorin disks of $\mathbf{A}$:

$$\lambda(\mathbf{A}) \subset \cup_{i=1}^{L} R_i(\mathbf{A}) = G(\mathbf{A}) \tag{5.23}$$

Furthermore, if any Gershgorin disk $R_i(\mathbf{A})$ is disjoint from the other Gershgorin disks of $\mathbf{A}$, then it contains exactly one eigenvalue of $\mathbf{A}$. By extension, the union of any k of these disks that do not intersect the remaining $m - k$ circles must contain precisely k of the eigenvalues, counting multiplicities.

Let $\mathbf{K}_{L \times L}$ be reexpressed as

$$\mathbf{K}_{L \times L} = \lfloor a_{ij} \rfloor = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1L} \\ a_{21} & a_{22} & \ddots & a_{2L} \\ \vdots & \ddots & \ddots & \vdots \\ a_{L1} & a_{L2} & \cdots & a_{L \times L} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{(L-1)\times(L-1)} & \mathbf{a} \\ \mathbf{a}^T & a_{L \times L} \end{bmatrix} \tag{5.24}$$

where $\mathbf{a} = \lfloor a_{1L}, a_{2L}, \ldots, a_{(L-1)L} \rfloor^T$ and $\mathbf{K}_{(L-1)\times(L-1)}$ is given by

$$\mathbf{K}_{(L-1)\times(L-1)} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1L} \\ a_{21} & a_{22} & \ddots & a_{2L} \\ \vdots & \ddots & \ddots & \vdots \\ a_{L1} & a_{L2} & \cdots & a_{(L-1)\times(L-1)} \end{bmatrix} \tag{5.25}$$

with the Gerschgorin radii defined by (5.21).

Now assume that $\{\lambda'_1 \geq \lambda'_2 \geq \cdots \geq \lambda'_{L-1}\}$ are eigenvalues obtained from $\mathbf{K}_{(L-1)\times(L-1)}$ and $\{\mathbf{v}'_1 \geq \mathbf{v}'_2 \geq \cdots \geq \mathbf{v}'_{L-1}\}$ are their corresponding orthonormal eigenvectors. Then,

$$\lambda_1 \geq \lambda'_1 \geq \lambda_2 \geq \lambda'_2 \geq \cdots \geq \lambda_{L-1} \geq \lambda'_{L-1} \geq \lambda_L \tag{5.26}$$

The eigenmatrix defined by $\boldsymbol{\Lambda}'_{(L-1)\times(L-1)} = \lfloor \mathbf{v}'_1 \mathbf{v}'_2 \cdots \mathbf{v}'_{L-1} \rfloor$ is a unitary matrix that transforms $\mathbf{K}_{(L-1)\times(L-1)}$ into a diagonal matrix denote by $\mathbf{D}'_{(L-1)(L-1)}$ with all the eigenvalues $\lambda'_1, \lambda'_2, \ldots, \lambda'_{L-1}$ in its diagonal line such that

$$\mathbf{D}'_{(L-1)\times(L-1)} = \left(\boldsymbol{\Lambda}'_{(L-1)\times(L-1)}\right)^T \mathbf{K}_{(L-1)\times(L-1)} \boldsymbol{\Lambda}'_{(L-1)\times(L-1)} \tag{5.27}$$

Now, we define $\tilde{\mathbf{\Lambda}}_{L\times L} = \begin{bmatrix} \mathbf{\Lambda}'_{(L-1)\times(L-1)} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$. Then

$$
\begin{aligned}
\tilde{\mathbf{K}}_{L\times L} &= \tilde{\mathbf{\Lambda}}^{T}_{L\times L}\mathbf{K}_{L\times L}\tilde{\mathbf{\Lambda}}_{L\times L} \\
&= \begin{bmatrix} \left(\mathbf{\Lambda}'_{(L-1)\times(L-1)}\right)^{T}\mathbf{K}_{(L-1)\times(L-1)}\mathbf{\Lambda}'_{(L-1)\times(L-1)} & \left(\mathbf{\Lambda}'_{(L-1)\times(L-1)}\right)^{T}\mathbf{a} \\ \mathbf{a}^{T}\mathbf{\Lambda}'_{(L-1)\times(L-1)} & a_{LL} \end{bmatrix} \\
&= \begin{bmatrix} \mathbf{D}'_{(L-1)\times(L-1)} & \left(\mathbf{\Lambda}'_{(L-1)\times(L-1)}\right)^{T}\mathbf{a} \\ \mathbf{a}^{T}\mathbf{\Lambda}'_{(L-1)\times(L-1)} & a_{LL} \end{bmatrix} \\
&= \begin{bmatrix} \lambda'_1 & 0 & \cdots & 0 & \rho'_1 \\ 0 & \lambda'_2 & \ddots & 0 & \rho'_2 \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & \lambda^{1}_{L-1} & \rho'_{L-1} \\ \rho'_1 & \rho'_2 & \cdots & \rho'_{L-1} & a_{LL} \end{bmatrix}
\end{aligned}
\tag{5.28}
$$

According to (5.28), the Gershgorin disks obtained for $\tilde{\mathbf{K}}_{L\times L}$ are given by $\rho_i = |\rho'_i|$ for $1 \leq i \leq L-1$ and $\rho_L = 0$. If there are p endmembers, then (5.28) becomes

$$
\tilde{\mathbf{K}}_{L\times L} = \begin{bmatrix} \lambda'_1 & 0 & 0 & \cdots & 0 & 0 & \rho'_1 \\ 0 & \ddots & 0 & \cdots & \cdots & 0 & \vdots \\ 0 & \cdots & \lambda'_p & \ddots & \ddots & 0 & \rho'_p \\ \vdots & \ddots & 0 & \sigma_n^2 & 0 & \vdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & \sigma_n^2 & 0 \\ \rho'_1 & \cdots & \rho'_p & 0 & \cdots & 0 & a_{LL} \end{bmatrix}
\tag{5.29}
$$

where σ_n^2 is noise variance. As a result, the Gershgorin disks $\rho_i = 0$ for $p+1 \leq i \leq L-1$. According to (5.29), it is clear to show that all the eigenvalue have been divided into two classes: Gershgorin disks with nonzero radii that include eigenvalues resulting from signal sources and Gershgorin disks with zero radii resulting from noise. Consequently the number of signal sources can be estimated by counting the number of Gershgorin disks with nonzero radii.

Similarly, $\mathbf{R}_{L\times L}$ can also be expressed as

$$
\mathbf{R}_{L\times L} = \left[b_{ij}\right] = \begin{bmatrix} \mathbf{R}_{(L-1)\times(L-1)} & \mathbf{b} \\ \mathbf{b}^{T} & b_{L\times L} \end{bmatrix}
\tag{5.30}
$$

where the Gershgorin radii are defined by

$$
\hat{\rho}_i = \sum\nolimits_{j=1,j\neq i}^{L} |b_{ij}| \text{ for } 1 \leq i \leq L
\tag{5.31}
$$

Using the same arguments outlined by (5.28)–(5.31), we can also obtain

$$
\begin{aligned}
\tilde{\mathbf{R}}_{L\times L} = \bar{\mathbf{\Lambda}}_{L\times L}^{T}\mathbf{R}_{L\times L}\bar{\mathbf{\Lambda}}_{L\times L} &= \begin{bmatrix} \hat{\lambda}_1' & 0 & \cdots & 0 & \hat{\rho}_1' \\ 0 & \hat{\lambda}_2' & \ddots & 0 & \hat{\rho}_2' \\ \vdots & \ddots & \ddots & 0 & \vdots \\ 0 & \cdots & 0 & \hat{\lambda}_{L-1}' & \hat{\rho}_{L-1}' \\ \hat{\rho}_1' & \hat{\rho}_2' & \cdots & \hat{\rho}_{L-1}' & b_{LL} \end{bmatrix} \\
&= \begin{bmatrix} \hat{\lambda}_1' & 0 & 0 & \cdots & 0 & 0 & \hat{\rho}_1' \\ 0 & \ddots & 0 & \cdots & \cdots & 0 & \vdots \\ 0 & \cdots & \hat{\lambda}_p' & \ddots & \ddots & 0 & \hat{\rho}_p' \\ \vdots & \ddots & 0 & \sigma_n^2 & 0 & \vdots & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & \sigma_n^2 & 0 \\ \hat{\rho}_1' & \cdots & \hat{\rho}_p' & 0 & \cdots & 0 & a_{LL} \end{bmatrix}
\end{aligned}
\tag{5.32}
$$

where $\hat{\lambda}_1' \geq \hat{\lambda}_2' \geq \cdots \geq \hat{\lambda}_{L-1}'$ are $(L-1)$ eigenvalues obtained from $\mathbf{R}_{(L-1)\times(L-1)}$ with

$$\hat{\lambda}_1 \geq \hat{\lambda}_1' \geq \hat{\lambda}_2 \geq \hat{\lambda}_2' \geq \cdots \geq \hat{\lambda}_{L-1} \geq \hat{\lambda}_{L-1}' \geq \hat{\lambda}_L \tag{5.33}$$

and $\bar{\mathbf{\Lambda}}_{(L-1)\times(L-1)}' = \left[\bar{\mathbf{v}}_1'\bar{\mathbf{v}}_2'\cdots\bar{\mathbf{v}}_{L-1}'\right]$ is a set of orthonormal eigenvectors corresponding to $\hat{\lambda}_1' \geq \hat{\lambda}_2 \geq \cdots \geq \hat{\lambda}_{L-1}'$. As a result, the Gershgorin disks obtained for $\tilde{\mathbf{R}}_{L\times L}$ are given by $\hat{\rho}_i = |\hat{\rho}_i'|$ for $1 \leq i \leq L-1$ and the Gershgorin disk radii $\hat{\rho}_i = 0$ for $p+1 \leq i \leq L$.

5.3.5.1 Thresholding Gershgorin Radii

According to (5.32), the Gershgorin disks with radii greater than zero should embrace all signal eigenvalues, while the Gershgorin disks with zero radii should include all noise eigenvalues. Therefore, the number of signal sources is the largest number p that yields a nonzero Gershgorin radius greater than zero.

$$\mathrm{VD}_{\mathrm{disk}}^{\mathrm{GR}}(\varepsilon) = \arg\left\{\max\nolimits_{1\leq p\leq L}|\hat{\rho}_p'| \geq \varepsilon\right\} \text{ for a given threshold } \varepsilon \tag{5.34}$$

5.3.5.2 Thresholding Difference Gershgorin Radii between $\mathbf{R}_{L\times L}$ and $\mathbf{K}_{L\times L}$

Since the correlation matrix includes the sample mean, the correlation eigenvalue is generally greater than or equal to its corresponding covariance eigenvalue, that is, $\hat{\lambda}_i \geq \lambda_i$ for all $1 \leq i \leq L$. Therefore, the Gershgorin radii of $\tilde{\mathbf{R}}_{L\times L}$ in (5.30), $|\hat{\rho}_i'|$, are usually greater than their corresponding Gershgorin radii of $\tilde{\mathbf{K}}_{L\times L}$ in (5.32), $|\rho_i'|$, for $1 \leq i \leq p$. Using this fact, the number of signal sources can be estimated by finding the smallest number p to solve the following optimization problem:

$$\mathrm{VD}_{\mathrm{diff}}^{\mathrm{GR}}(\varepsilon) = \arg\left\{\min_{1 \le i \le L}\left[|\hat{\rho}_i'| - |\rho_i'| \le \varepsilon\right]\right\} \text{ for a given threshold } \varepsilon \tag{5.35}$$

This criterion is reduced to the criterion described in Section 5.3.1.2, thresholding difference between correlation eigenvalue and covariance eigenvalue.

5.3.6 HFC Method

Up to now, all the criteria described above manipulate eigenvalues to come up with a means of determining VD. However, in order for eigenvalues to appropriately determine VD the signal sources must be effectively characterized by eigenvalues, that is, data variances. In hyperspectral imagery signal sources such as endmembers, anomalies considered to be interesting are actually insignificant in terms of eigenvalues due to their small sample pools. Consequently, their contributions to eigenvalues are generally limited and usually very little. As expected, using eigenvalues as a criterion may not be an effective measure in determining VD. In order to address this issue Harsanyi *et al.* (1994a) proposed an approach, referred to as the HFC method, to resolve this dilemma.

The idea is very simple. It assumes that hyperspectral signatures of interest are unknown, nonrandom and deterministic signal sources and noise is white Gaussian. Under this circumstance the signal sources will be only contributed to the first-order statistics, which is the data sample mean. This assumption seems reasonable. Despite that the noise in hyperspectral imagery is generally not Gaussian such a non-Gaussian noise is generally caused by unknown interferers that are considered as structure noise, for example, clutters or small unidentified disturbed sources. If these structure noises are removed from the noise, the remainder must be completely random in which case it must be white Gaussian. This is why the noise in communications is generally assumed to be white Gaussian.

In order to materialize this concept, it first calculates the sample autocorrelation matrix, $\mathbf{R}_{L\times L} = \sum_{i=1}^{N} \mathbf{r}_i \mathbf{r}_i^T$, and sample autocovariance matrix, $\mathbf{K}_{L\times L} = \sum_{i=1}^{N} (\mathbf{r}_i - \boldsymbol{\mu})(\mathbf{r}_i - \boldsymbol{\mu})^T$. Then for each $1 \le l \le L$ where the L is the number of spectral channels it finds the difference between their corresponding eigenvalues, that is, $\{\hat{\lambda}_1 \ge \hat{\lambda}_2 \ge \cdots \ge \hat{\lambda}_L\}$ and $\{\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_L\}$, which are two sets of eigenvalues generated by $\mathbf{R}_{L\times L}$ and $\mathbf{K}_{L\times L}$, called correlation eigenvalues and covariance eigenvalues, respectively. If a hyperspectral signal source is present in the data, there should be some spectral dimension l with $1 \le l \le L$ such that $\hat{\lambda}_l > \lambda_l$ due to the fact that the signal source will contribute to the sample mean in the sample correlation matrix $\mathbf{R}_{L\times L}$ but not to the sample covariance matrix $\mathbf{K}_{L\times L}$ that has removed the sample mean.

By assuming that signal sources are nonrandom unknown positive constants and noise is white with zero mean, we can expect that

$$\hat{\lambda}_l > \lambda_l \text{ for } l = 1, \ldots, \mathrm{VD} \tag{5.36}$$

and

$$\hat{\lambda}_l = \lambda_l \text{ for } l = \mathrm{VD} + 1, \ldots, L \tag{5.37}$$

Using (5.36) and (5.37), the eigenvalues in the lth spectral channel can be related by

$$\begin{array}{l} \hat{\lambda}_l > \lambda_l > \sigma_{n_l}^2 \text{ for } l = 1, \ldots, \mathrm{VD} \\ \text{and} \\ \hat{\lambda}_l = \lambda_l = \sigma_{n_l}^2 \text{ for } l = \mathrm{VD} + 1, \ldots, L \end{array} \tag{5.38}$$

where $\sigma_{n_l}^2$ is the noise variance in the lth spectral channel.

In order to determine the VD, Harsanyi *et al.* (1994a) formulated the VD determination problem as a binary hypothesis problem as follows:

$$\begin{array}{ll} H_0 : z_l = \hat{\lambda}_l - \lambda_l = 0 & \\ \text{versus} & \text{for } l = 1, 2, \ldots, L \\ H_1 : z_l = \hat{\lambda}_l - \lambda_l > 0 & \end{array} \tag{5.39}$$

where the null hypothesis H_0 and the alternative hypothesis H_1 represent the case that the correlation eigenvalue is equal to its corresponding covariance eigenvalue and the case that the correlation eigenvalue is greater than its corresponding covariance eigenvalue, respectively. In other words, when H_1 is true (i.e., H_0 fails), it implies that there is an endmember contributing to the correlation eigenvalue in addition to noise, since the noise energy represented by the eigenvalue of $\mathbf{R}_{L\times L}$ in that particular component is the same as the one represented by the eigenvalue of $\mathbf{K}_{L\times L}$ in its corresponding component.

Let $\{\hat{\lambda}_l\}_{l=1}^{L}$ be the eigenvalues of the data sample correlation matrix $\mathbf{R}_{L\times L}$ and $\{\lambda_l\}_{l=1}^{L}$ be the eigenvalues of the data sample covariance matrix $\mathbf{K}_{L\times L}$. If we assume that the noise in each spectral dimension has zero mean and variance, $\sigma_{n_l}^2$, then $\hat{\lambda}_l = \mu_l^2 + \sigma_{n_l}^2$ and $\lambda_l = \sigma_{n_l}^2$ where μ_l is the sample mean of the lth spectral dimension. Furthermore, suppose that the signatures of interest in hyperspectral image analysis are material substances that generally cannot be identified *a priori* or by visual inspection, such as endmembers, anomalies, man-made objects, and so on. So, the set of data sample vectors specified by a hyperspectral signature of such type, s_l in the lth spectral dimension, denoted by Δ_{s_l}, generally cannot be too large and its number, $|\Delta_{s_l}|$, is relatively small. So, the variance $\sigma_{s_l}^2$ specified by the sample statistics resulting from intrapixel spectral variability of s_l within Δ_{s_l} in terms of second-order statistics is considered to be very small and insignificant, approximately $\sigma_{s_l}^2 \approx 0$. In this case, the sample mean can be approximated by its signal energy s_l^2, that is, $\mu_l^2 \approx s_l^2$. Consequently, the data variance of the lth spectral dimension, σ_l^2, is nearly the same as its noise variance, $\sigma_{n_l}^2$. That is, $\hat{\lambda}_l = \mu_l^2 + \sigma_l^2 \approx s_l^2 + \sigma_{n_l}^2$ and $\lambda_l = \sigma_l^2 = \sigma_{s_l}^2 + \sigma_{n_l}^2 \approx \sigma_{n_l}^2$. In light of this interpretation a binary hypothesis testing problem can be formatted as follows.

Despite the fact that the $\hat{\lambda}_l$ and λ_l in (5.36)–(5.38) are unknown constants, according to Anderson (1984), we can model each pair of eigenvalues, $\hat{\lambda}_l$ and λ_l, under hypotheses H_0 and H_1 as random variables by the asymptotic conditional probability densities given by

$$p_0(z_l) = p(z_l|H_0) \cong N(0, \sigma_{z_l}^2) \text{ for } l = 1, 2, \ldots, L \tag{5.40}$$

and

$$p_1(z_l) = p(z_l|H_1) \cong N(\mu_l, \sigma_{z_l}^2) \text{ for } l = 1, 2, \ldots, L \tag{5.41}$$

respectively, where μ_l is an unknown constant and the variance $\sigma_{z_l}^2$ for $l = 1, 2, \ldots, L$ is given by

$$\sigma_{z_l}^2 = \mathrm{var}\left[\hat{\lambda}_l - \lambda_l\right] = \mathrm{var}\left[\hat{\lambda}_l\right] + \mathrm{var}[\lambda_l] - 2\mathrm{cov}\left(\hat{\lambda}_l, \lambda_l\right) \tag{5.42}$$

It has been shown that when the total number of samples N is sufficiently large, $\mathrm{var}\left[\hat{\lambda}_l\right] \cong 2\hat{\lambda}_l^2/N$ and $\mathrm{var}[\lambda_l] \cong 2\lambda_l^2/N$. Therefore, the noise variance $\sigma_{z_l}^2$ in (5.38) can be estimated and approximated using (5.42).

From (5.38), (5.41), and (5.42), we define the false alarm probability and detection power (i.e., detection probability) as follows:

$$P_F = \int_{\tau_l}^{\infty} p_0(z)dz \tag{5.43}$$

$$P_D = \int_{\tau_l}^{\infty} p_1(z)dz \tag{5.44}$$

An NP detector for $\hat{\lambda}_l - \lambda_l$, denoted by $\delta^{\mathrm{NP}}(l)$ for the binary composite hypothesis testing problem specified by (5.39), can be obtained by maximizing the detection power P_D in (5.44), while the false alarm probability P_F in (5.43) is fixed at a specific value, α, which determines the threshold value τ_l in (5.43) and (5.44). It is a randomized decision rule given by

$$\delta^{\mathrm{NP}}(z_l) = \begin{cases} 1, \text{ if } \Lambda(z_l) > \tau_l \\ 1 \text{ with probability } \kappa, \text{ if } \Lambda(z_l) = \tau_l \\ 0, \text{ if } \Lambda(z_l) < \tau_l \end{cases} \tag{5.45}$$

where the likelihood ratio test $\Lambda(z_l) = p_1(z_l)/p_0(z_l)$ with $p_0(z_l)$ and $p_1(z_l)$ given by (5.40) and (5.41), respectively.

So, a case of $\hat{\lambda}_l - \lambda_l > \tau_l$ indicates that $\delta^{\mathrm{NP}}(z_l)$ fails the test, in which case there is signal energy assumed to contribute to the eigenvalue, $\hat{\lambda}_l$, in the lth spectral dimension. It should be noted that the test for (5.39) must be performed for each of L spectral dimensions. Therefore, for each pair of $\hat{\lambda}_l - \lambda_l$, the threshold τ is different and should be dependent on spectral dimension, that is τ_l. VD resulting from the binary hypothesis testing problem specified by the NP detector $\delta^{\mathrm{NP}}(z_l)$ given by (5.45) is referred to as $\mathrm{VD}_{\mathrm{HFC}}^{\mathrm{NP}}(P_F)$ and given by

$$\mathrm{VD}_{\mathrm{HFC}}^{\mathrm{NP}}(P_F) = \sum\nolimits_{l=1}^{L} [\delta_{\mathrm{NP}}(z_l)] \text{ for a given false alarm probability } P_F \tag{5.46}$$

where P_F is a predetermined false alarm probability and $[x]$ is defined as the largest integer less than or equal to x, that is, $\lfloor \delta(z_l) \rfloor = 1$ only if $\delta^{\mathrm{NP}}(z_l) = 1$ and $\lfloor \delta(z_l) \rfloor = 0$ if $\delta^{\mathrm{NP}}(z_l) < 1$. More specifically, (5.38) can be described by

$$\begin{aligned} &\hat{\lambda}_{s(l)} > \lambda_{s(l)} \text{ for } l = 1, 2, \ldots, \mathrm{VD} \\ &\text{and} \\ &\hat{\lambda}_{ns(l)} = \lambda_{ns(l)} \text{ for } l = \mathrm{VD}+1, \ldots, L \end{aligned} \tag{5.47}$$

where $s(l)$ and $ns(l)$ are the lth spectral dimensions that contain a signal source, that is, $\delta^{\mathrm{NP}}\left(z_{s(l)}\right) = 1$ and no signal source, that is, $\delta^{\mathrm{NP}}\left(z_{ns(l)}\right) < 1$, respectively. In general, $s(l) \neq l$ and $ns(l) \neq l$. Without loss of generality we can arrange for the first VD spectral dimensions to contain signal sources by setting $s(l) = l$ and the rest of spectral dimensions to contain no signal sources by assuming $ns(l) = l$. Using (5.47) the eigenvalues in the lth spectral dimension can be reexpressed by

$$\begin{aligned} &\hat{\lambda}_l > \lambda_l > \sigma_{n_l}^2 \text{ for } l = 1, \ldots, \mathrm{VD} \\ &\text{and} \\ &\hat{\lambda}_l = \lambda_l = \sigma_{n_l}^2 \text{ for } l = \mathrm{VD}+1, \ldots, L \end{aligned} \tag{5.48}$$

where $\sigma_{n_l}^2$ is the noise variance in the lth spectral dimension.

As noted above, the signature variance $\sigma_{s_l}^2$ is generally very small and can be influenced by interband covariances. To resolve this impact a preprocessing of noise whitening via a noise covariance matrix estimated by a technique developed by Roger and Arnold (1996) can be performed prior to the HFC method. The resulting HFC method is referred to as the noise-whitened HFC (NWHFC) method and the resulting VD is defined as $\mathrm{VD}_{\mathrm{NWHFC}}^{\mathrm{NP}}(P_F)$.

Several remarks on the HFC/NWHFC method are noteworthy.

1. In order for the HFC/NWHFC method to work effectively, the spectrally distinct signatures defined in VD are those that do not have significant contributions to data variances. Such signatures are generally characterized by three unique features. Firstly, the probabilities of their occurrence are usually low. Secondly, when such signatures are present, their samples are not too many. Thirdly, due to a small variance that is generally negligible such signatures are usually unique in terms of spectral distinction. In other words, the variance of a signature, s_l, $\sigma_{s_l}^2$, is generally very small due to its very small sample size of data sample vectors characterized by Δ_{s_l}. Therefore, the presence of s_l can only be detected by s_l^2 not $\sigma_{s_l}^2$.
2. By virtue of (5.48) the effectiveness of the HFC/NWHFC method is actually determined by the strength of the signature energy, s_l^2, which only contributes to the sample mean of the lth spectral dimension.

The HFC/NWHFC method does not provide a means of finding spectral signatures it detects; hence, these signatures are determined by various algorithms designed for different applications. As a result, VD estimated by the HFC/NWHFC method may work well for some applications but may perform poorly for other applications. In other words, the HFC/NWHFC method should not be considered as a one-size-fits-all technique for all applications.

5.3.7 Discussions on Data Characterization-Driven Criteria

When VD is performed to determine the value of p, a premise assumes that there is no *a priori* knowledge about the data. Fortunately, hyperspectral imagery provides a wealth of spectral information generated by hundreds of contiguous spectral bands. Accordingly, using such hyperspectral information as data characterization becomes the only means that can be used to characterize signal sources of interest. A natural approach is to find the sample spectral correlation among data sample vectors via either the sample spectral correlation matrix $\mathbf{R}_{L\times L}$ or sample covariance matrix $\mathbf{K}_{L\times L}$ to capture statistics of signal sources. However, such sample spectral statistics should not be confused with sample spatial statistics commonly used in spatial domain-based image processing techniques. This is because the sample correlation matrix $\mathbf{R}_{L\times L}$ is invariant to any permutation $\{P(i)\}_{i=1}^N$ of $\{i\}_{i=1}^N$, that is, $\mathbf{R}_{L\times L} = \sum_{i=1}^N \mathbf{r}_{P(i)}\mathbf{r}_{P(i)}^T$ where $\{\mathbf{r}_i\}_{i=1}^N = \left\{\mathbf{r}_{P(i)}\right\}_{i=1}^N$. This implies that $\mathbf{R}_{L\times L}$ is an intrasample correlation matrix, not an intersample spatial correlation matrix. Bearing this in mind it is not surprising to learn that all the data characterization-driven criteria described above are based on sample spectral statistics to determine VD.

As a matter of fact, all of the data characterization-driven criteria derived above have their merits in signal processing and communications and each one has it own right in a particular application. For example, the information theoretic criteria, AIC and MDL, have been shown to be effective in determining the number of signals arriving at an array of passive sensors and so is Gershgorin radius when signal sources are considered to be indistinguishable and noise is i.i.d.. The eigen-analysis based on eigenvalue distribution has been widely used to determine the number of principal components needed to be retained after PCA and so are both SVD and FA used in Malinowski's error theory. Unfortunately, as shown in the following experiments these approaches are generally ineffective. One major reason is that the signal sources of interest in hyperspectral imagery are mostly small objects that provide crucial and important information compared to those in multispectral imagery that are most likely patterns. In order to form a pattern there must be sufficient data sample vectors closely related in terms of spectral correlation to constitute statistics in which case its variance provides a good indication to represent this particular pattern. While this may be true for multispectral imagery the same logic cannot be applied to hyperspectral imagery. This is mainly due to the fact that there are too many spectrally distinct signal sources of interest in hyperspectral imagery whose sample pools are too few to produce creditable variances for data processing. As a consequence, using eigenvalues to characterize variances of signal sources in determining VD does not always work, specifically when signal sources of interest have a relatively small number of samples such as endmembers, anomalies, man-made objects, and so on.

Nevertheless, there is a catch between the HFC method and all other criteria. As noted in Chapter 1, an effective hyperspectral imaging technique should be derived directly from an aspect of hyperspectral imagery not multispectral imagery. This is also true for VD determination. Except the HFC method all other data characterization-driven criteria follow traditional eigen-based spatial domain approaches that use eigenvalues as data variance to characterize the data properties. The HFC method was derived exactly different from this point of view. In order to illustrate this critical difference we specifically use the PCA as an example to compare with the HFC method.

1. Despite that both PCA and the HFC method use eigenvalues to determine VD there is a key difference between them. Compared to PCA that uses eigenvalues as data variances to characterize the entire data as a whole the HFC method calculates the difference between correlation eigenvalues and covariance eigenvalues to capture variation in spectral sample mean for each of spectral dimensions, that is, one spectral dimension at a time.
2. The HFC method implements a binary hypothesis test for each spectral dimension to test if each of spectral dimensions can be used to accommodate one signal source as opposed to PCA that uses eigenvalues to determine the total number of signal sources instead of a signal source in an individual spectral dimension.
3. PCA differs from the HFC method in that PCA uses the second-order statistics such as data variance to determine VD while the HFC method uses the first-order statistics, which is the sample mean vector to determine VD. Therefore, unless a hyperspectral signature is a random signal that may contribute to second-order statistics PCA cannot be very effective in determining VD.
4. The signal sources of interest in hyperspectral data are generally "*spectrally*" distinct as defined by VD and usually do not have a large pool of sample vectors present in the data. As a result, the variances of these signal sources described by second-order statistics resulting from these sample vectors are relatively small and nearly negligible. On the other hand, these signal sources generally contributed to the sample mean vector as first-order statistics. As a consequence, data sample mean is more significant and crucial than the data variance. That is why background signatures instead of signal sources are generally extracted in the first few principal components produced by PCA due to their larger variances. On the contrary, the hyperspectral signatures of major interest are usually deterministic signal sources that contribute to first-order statistics. In this case, the HFC method can be very effective in determining a spectral dimension containing a signature. This also explains why the HFC method works effectively when signal signatures are spectrally distinct and their sample pools are small in which case PCA fails because the variances of such signal signatures are very small compared to that of background signatures.

5. Last but not least, one common issue arising in the use of VD is that users always believe in that VD and the HFC method are tied together as a pair and think that VD can only be found by the HFC method. This is certainly a misinterpretation of VD. In analogy with ID that defines the minimum number of parameters to represent high-dimensional data VD is a concept that defines an effective spectral dimensionality that is the number of "*spectrally distinct*" signatures required to characterize and represent a hyperspectral image. In fact, the effectiveness of VD is actually determined by various applications where different types of "*spectral signatures*" are of major interest and the techniques used to find these signatures are the key to its success. The HFC method is developed as "*one*" technique to determine VD in a similar manner as PCA is used as "*one*" technique to determine ID. So, when the HFC method does not work effectively, a new technique must be sought. It should not imply that VD is not correctly defined. This simply indicates that the assumption made by the HFC method is not appropriate.

5.4 VD Determined by Data Representation-Driven Criteria

In Section 5.3, VD is determined by data characterization where all the developed criteria provide no specific algorithms to find signal sources. Accordingly, VD remains the same if different applications are considered. Although some criteria can use a parameter such as error threshold ε or false alarm probability P_F to fine-tune VD, this practice still has its limitations in real applications. For example, the number of endmembers in endmember extraction is certainly different from the number of anomalies in anomaly detection. It seems that a constraint in using data characterization to determine VD is the lack of algorithms to find signal sources that generally vary with applications. In order to resolve this dilemma, one feasible approach is to use data representation to determine VD where the basic elements to construct the entire data are those signal sources that determine VD. In this case, VD is tied together with an algorithm to find these basic signal sources. The most commonly used data representation is a linear regression model in multivariate data analysis where data samples are modeled as linear combinations of a finite set of basic elements. For example, a real number can be expressed by a binary representation where the basic elements are integer powers of 2. A one-dimensional signal can also be represented by a set of sinusoidal functions known as Fourier transform/series or by a wavelet representation in multiple scales where basic elements are mother wavelets such as Haar wavelet, Mexican hat wavelet, or Daubechies' wavelets (Vetterli and Kovacevic, 1995). A two-dimensional image can be represented by a set of basic elements that are Walsh-Hadamard basis matrices (Gonzalez and Woods, 2002). Other linear data representations include data represented by PCA where basic elements are principal components specified by eigenvectors in terms of data variances, that is, eigenvalues, and Fisher's linear discriminant analysis (LDA) where data samples are represented by a set of basic elements specified by feature vectors (Duda and Hart, 1973). All aforementioned linear data representations share two key issues in common, namely, how to determine the number of basic elements, referred to as p, and how to select appropriate basic elements. Obviously, none of these two issues is a trivial matter, not mentioning nonlinear data representation, which is more complicated than linear data representation.

In what follows, two linear data representation-driven approaches are developed to determine VD.

5.4.1 Orthogonal Subspace Projection (OSP)

Assume that there are p signal sources, $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p\}$, present in the data to be processed and every data sample vector $\mathbf{r}_i$ can be expressed by a linear mixture of these p signal sources as follows:

$$\mathbf{r}_i = \mathbf{S}_p \boldsymbol{\alpha}_i + \mathbf{n}_i \tag{5.49}$$

where $\mathbf{S}_p = [\mathbf{s}_1\mathbf{s}_2 \cdots \mathbf{s}_p]$ is a signal matrix made up of the p signal sources, $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p\}$, and $\mathbf{n}_i$ can be interpreted as the noise vector or model error vector.

Let $\mathbf{P}_p = \mathbf{S}_p\left(\mathbf{S}_p^T\mathbf{S}_p\right)^{-1}\mathbf{S}_p^T$ be the p-signal projection matrix formed by the signal matrix $\mathbf{S}_p$. It can then be used to map all data sample vectors $\{\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_N\}$ into the space linearly spanned by the p signal sources, $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p\}$, denoted by $\langle \mathbf{S}_p \rangle = \langle \mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p \rangle$. In other words, every data sample vector $\mathbf{r}_i$ can be expressed by a linear mixture of the p signal sources, $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p\}$, specified by (5.49) via the p-signal projection matrix $\mathbf{P}_p$ where the noise $\mathbf{n}_i$ in (5.49) is included to account for the linear mixture model error. Since each sample vector produces a different model error and residual, the sample mean vector is used to represent an averaged model error and residual. In this case, from (5.49) the sample mean vector $\boldsymbol{\mu}$ can be expressed by

$$\begin{aligned} \boldsymbol{\mu} &= (1/N)\left\lfloor \mathbf{S}_p\left(\textstyle\sum_{i=1}^N \alpha_i\right) + \sum_{i=1}^N \mathbf{n}_i \right\rfloor \\ &= \left[\mathbf{S}_p\left((1/N)\textstyle\sum_{i=1}^N \alpha_i\right) + (1/N)\sum_{i=1}^N \mathbf{n}_i\right] = \mathbf{S}_p\bar{\alpha}_p + \bar{\mathbf{n}} \end{aligned} \tag{5.50}$$

where $\bar{\alpha}_p = (1/N)\sum_{i=1}^N \alpha_i$ and $\bar{\mathbf{n}} = (1/N)\sum_{i=1}^N \mathbf{n}_i$ with the covariance matrix given by $\mathbf{R}_{\bar{\mathbf{n}}} = (1/N)\sum_{i=1}^N \mathbf{n}_i\mathbf{n}_i^T$. Using (5.50) we can obtain

$$\begin{aligned} E\left[(\mathbf{P}_p\boldsymbol{\mu})(\mathbf{P}_p\boldsymbol{\mu})^T\right] &= E\left[\left(\mathbf{P}_p(\mathbf{S}_p\bar{\alpha}_p + \bar{\mathbf{n}})\left(\mathbf{P}_p(\mathbf{S}_p\bar{\alpha}_p + \bar{\mathbf{n}})\right)^T\right] \\ &= E\left[\mathbf{S}_p\bar{\alpha}_p\bar{\alpha}_p^T\mathbf{S}_p^T\right] + E\left[\mathbf{P}_p\bar{\mathbf{n}}\bar{\mathbf{n}}^T\mathbf{P}_p\right] \\ &= \mathbf{S}_p\bar{\alpha}_p\bar{\alpha}_p^T\mathbf{S}_p^T + \mathbf{P}_p^T\mathbf{R}_{\bar{\mathbf{n}}}\mathbf{P}_p \end{aligned} \quad . \tag{5.51}$$

There are two ways to find $\mathrm{VD}^{\mathrm{OSP}}$ as follows:

$$\begin{aligned} \mathrm{OSP}(p) &= \mathrm{trace}\left\{\mathbf{S}_p\bar{\alpha}_p\bar{\alpha}_p^T\mathbf{S}_p^T + \mathbf{P}_p^T\mathbf{R}_{\bar{\mathbf{n}}}\mathbf{P}_p\right\} \\ &= E\left[(\mathbf{P}_p\boldsymbol{\mu})^T(\mathbf{P}_p\boldsymbol{\mu})\right] \end{aligned} . \tag{5.52}$$

Or approximately to OSP(p) by

$$\mathrm{OSP}(p) = \bar{\alpha}_p^T\mathbf{S}_p^T\mathbf{S}_p\bar{\alpha}_p \tag{5.53}$$

without involving noise covariance matrix estimation. Theoretically, the value of OSP(p) in (5.52) increases as the value of p increases. For any given error threshold ε, VD can be determined by a stopping rule by ε. The value of p determining the OSP(p) in (5.52) or (5.53) is denoted by $\mathrm{VD}^{\mathrm{OSP}}$. Two criteria are developed to detect the abrupt change of the OSP(p) value. One that is based on the gradient, denoted by "∇," is defined by

$$\mathrm{VD}_{\mathrm{algorithm}}^{\mathrm{OSP},\nabla}(\varepsilon) = \arg\left\{\min_{1\le p\le L}\left|\frac{\mathrm{OSP}(p+1)}{\mathrm{OSP}(p)} - \frac{\mathrm{OSP}(p)}{\mathrm{OSP}(p-1)}\right| < \varepsilon\right\} \tag{5.54}$$

and the other that is based on the difference, denoted by minus "$-$," is defined by

$$\mathrm{VD}_{\mathrm{algorithm}}^{\mathrm{OSP},-}(\varepsilon) = \arg\left\{\min_{1\le p\le L}|\mathrm{OSP}(p+1) - \mathrm{OSP}(p)| < \varepsilon\right\} \tag{5.55}$$

For the difference criterion, the sample mean vector $\boldsymbol{\mu}$ is normalized before orthogonal projection so that the values of threshold ε for these two criteria are comparable for analysis. The threshold ε

in (5.54) and (5.55) is generally selected according to a sudden drop or a clear gap between two consecutive values of p in plots of the gradient in (5.54) and difference (5.55) versus the value of p.

It should be noted that (5.52), (5.54), or (5.55) actually involves two key parameters needed to be addressed. One is the error threshold ε, which is already included in (5.54) and (5.55). The other parameter is the algorithm that is used to produce the p-signal matrix $\mathbf{S}_p$, which is not particularly specified in (5.54) and (5.55) but rather uses the term of "algorithm" in (5.54) and (5.55) for a generic expression. Since the OSP method is derived from the linear spectral unmixing approach, the spectral signal sources $\{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_p\}$ in (5.49) determined by $\mathrm{VD}^{\mathrm{OSP}}$ are referred to as "virtue endmembers" instead of image endmembers commonly used in linear spectral unmixing.

Apparently, a different algorithm will produce a different p-signal matrix $\mathbf{S}_p$. As a result, the value of $\mathrm{VD}^{\mathrm{OSP}}$ will also be different. For example, when the algorithm used to produce $\mathrm{VD}^{\mathrm{OSP}}$ is an endmember extraction algorithm such as pixel purity index (PPI) (Boardman, 1994), N-finder algorithm (N-FINDR) (Winter, 1999a,b), vertex component analysis (VCA) (Nascimento and Bioucas-Dias, 2005), simplex growing algorithm (SGA) (Chang *et al.* 2006), and so on, the $\mathbf{S}_p$ becomes an endmember matrix $\mathbf{E}_p$ formed by p endmembers $\{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_p\}$. In this case, $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{PPI}}$, $\mathrm{VD}^{\mathrm{OSP}}_{\text{N-FINDR}}$, $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{VCA}}$, and $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{SGA}}$ are the number of endmembers estimated by specific endmember extraction algorithms. However, if the algorithm used to produce $\mathrm{VD}^{\mathrm{OSP}}$ is not an endmember extraction algorithm, but rather other algorithms such as an automatic target generation process (ATGP) (Ren and Chang, 2003), the matrix $\mathbf{S}_p$ is then not necessarily an endmember matrix and can be considered as a target signature matrix $\mathbf{M}_p$. In this case, $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{ATGP}}$ is actually the number of target signatures of interest. So, when an algorithm is specified to produce $\mathbf{S}_p$ for $\mathrm{VD}^{\mathrm{OSP}}$ in (5) or (6), a specific algorithm will be included in the notation of $\mathrm{VD}^{\mathrm{OSP}}$ for clarity, such as $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{UNCLS}}(\varepsilon)$ with the unsupervised nonnegativity constrained least squares (UNCLS) (Chang and Heinz, 2000a,b) to produce $\mathbf{S}_p$, $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{UFCLS}}(\varepsilon)$ with the unsupervised fully constrained least squares (UFCLS) (Heinz and Chang, 2001) to produce $\mathbf{S}_p$, and $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{ATGP}}(\varepsilon)$ with ATGP to produce a target signature matrix $\mathbf{M}_p$. However, if no particular algorithm is specified in $\mathrm{VD}^{\mathrm{OSP}}$, the SVD can be used as a generic algorithm to produce $\mathbf{S}_p$ in which case it is $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{SVD}}(\varepsilon)$.

5.4.2 Signal Subspace Estimation (SSE)

When the OSP(p) is derived in (5.52), there is no assumption made on the noise $\bar{\mathbf{n}} = (1/N)\sum_{i=1}^{N} \mathbf{n}_i$. Using the model described by (5.49) with an assumption of Gaussian noise, Bioucas-Dias and Nascimento (2005) derived the so-called SSE to determine the number of spectral signal sources, p. In this case, the noise in (5.49) is Gaussian with zero mean and covariance matrix given by $\mathbf{K}_{\bar{\mathbf{n}}} = (1/N)\sum_{i=1}^{N} E\left[\mathbf{n}_i\mathbf{n}_i^T\right]$. If the signal is expressed as $\mathbf{x}_i = \mathbf{r}_i - \mathbf{n}_i$ from (5.49), mean squared error (MSE), $\mathrm{MSE}(p) = E\left\lfloor \left(\bar{\mathbf{x}} - \hat{\bar{\mathbf{x}}}_p\right)^T \left(\bar{\mathbf{x}} - \hat{\bar{\mathbf{x}}}_p\right) \right\rfloor$, where $\hat{\bar{\mathbf{x}}}_p = \hat{\mathbf{P}}_p \bar{\mathbf{r}}$ and $\bar{\mathbf{x}} = (1/N)\sum_{i=1}^{N} \mathbf{x}_i$, can be calculated. The SSE estimates the number of spectral signal sources based on the dimensionality of its found signal subspace by solving the following optimization problem:

$$\mathrm{VD}^{\mathrm{SSE}} = \arg\left\{ \min\nolimits_{1 \le p \le L} \left(\bar{\mathbf{r}}^T \hat{\mathbf{P}}_p^{\perp} \bar{\mathbf{r}} + 2\,\mathrm{trace}\left[\hat{\mathbf{P}}_p \mathbf{K}_{\bar{\mathbf{n}}}/N\right] \right) \right\} \tag{5.56}$$

where $\hat{\mathbf{P}}_p = \mathbf{S}_p\mathbf{S}_p^T$, $\bar{\mathbf{r}} = (1/N)\sum_{i=1}^{N} \mathbf{r}_i$ is obtained from (1), $\mathbf{K}_{\bar{\mathbf{n}}}$ is the covariance matrix of the noise, and $\mathbf{S}_p$ is composed of p eigenvectors for the covariance matrix of signal $\mathbf{x}_i$, corresponding to p largest magnitudes of eigenvalues.

Recently, an improved version of SSE, called hyperspectral signal subspace identification by minimum error (HySime), was derived by Bioucas-Dias and Nascimento (2008). The difference between SSE and HySime is that the MSE in HySime is calculated in two steps, that is, $\mathrm{MSE}(p|\mathbf{x}_i) = E\left\lfloor \left(\mathbf{x}_i - \hat{\mathbf{x}}_{i,p}\right)^T \left(\mathbf{x}_i - \hat{\mathbf{x}}_{i,p}\right)|\mathbf{x}_i \right\rfloor$ and $\mathrm{MSE}(p) = E[\mathrm{MSE}(p|\mathbf{x}_i)]$, where $\hat{\mathbf{x}}_{i,p} = \hat{\mathbf{P}}_p \mathbf{r}_i$ and HySime projects individual signal vectors but not their means as done by SSE. This leads to the following optimization problem:

$$\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{Hysime}} = \arg\left\{\min_{1 \le p \le L}\left(\mathrm{trace}\left[\hat{\mathbf{P}}_p^{\perp}\mathbf{K}_{\mathbf{r}}\right] + 2\,\mathrm{trace}\left[\hat{\mathbf{P}}_p \mathbf{K}_{\mathbf{n}}\right]\right)\right\} \tag{5.57}$$

where $\mathbf{K}_{\mathbf{r}}$ is the covariance matrix of a sample vector $\mathbf{r}$, $\mathbf{K}_{\mathbf{n}}$ is the covariance matrix of the estimated noise, and $\hat{\mathbf{P}}_p$ is the projection matrix same as that in (5.57). By further simplification, HySime in (5.58) can be rederived as

$$\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{Hysime}} = \arg\left\{\min_{1 \le p \le L}\left\lfloor \sum\nolimits_{j=1}^{p} \left(-v_j + 2\sigma_j^2\right)\right\rfloor\right\} \tag{5.58}$$

where $v_j = \mathbf{e}_j^T \mathbf{K}_{\mathbf{r}} \mathbf{e}_j$ and $\sigma_j^2 = \mathbf{e}_j^T \mathbf{K}_{\mathbf{n}} \mathbf{e}_j$ for $1 \le j \le L$ and $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_L\}$ are the eigenvectors of $\mathbf{K}_{\mathbf{x}}$, the estimated covariance matrix of the signal vector $\mathbf{x}$. Let $\delta_j = -v_j + 2\sigma_j^2$. The minimization in (5.58) is achieved when all δ_j's are negative and thus VD is the number of negative δ_j's. Compared to SSE one theoretical advantage of HySime over SSE is that HySime searches all possible permutations of eigenvectors to find negative δ_j's, while SSE projects data sample vectors onto the subspace linearly spanned by the eigenvectors that are obtained by descending eigenvalues (Bioucas-Dias and Nascimento, 2008).

5.4.3 Discussions on OSP and SSE/HySime

The first technique ever developed to estimate VD is the HFC method. It formulates VD estimation as a binary composite hypothesis testing problem and then designs an NP test to see if a spectral signal source is present in a particular spectral dimension. The beauty of the HFC method is its performance determined by the false alarm probability, P_F. By tuning the value of P_F according to different applications the value of VD also varies. However, there is a disadvantage of using the HFC method where there is no guideline provided to find the spectral signal sources once the value of VD is determined. The two techniques, $\mathrm{VD}^{\mathrm{OSP}}$ and $\mathrm{VD}^{\mathrm{SSE/HySime}}$, presented in the previous section are developed for this purpose in the sense of linear data representation by finding a best number of signatures, which determines the VD, to represent all the data sample vectors in a linear form. The development of $\mathrm{VD}^{\mathrm{OSP}}$ is inspired by this drawback. Its idea is based on LSMA where the entire data can be represented via a linear mixing model formed by a finite set of spectral signal sources that yields the minimum mixing error. In this case, the number of spectral signal sources is VD and the spectral signal sources can be considered as VEs as opposed to commonly used image endmembers in LSMA. It should be noted that these VEs vary with algorithms that are used to find them.

Interestingly, $\mathrm{VD}^{\mathrm{OSP}}$ is not the only LSMA-based technique that can be used for VD estimation. $\mathrm{VD}^{\mathrm{SSE}}$ and $\mathrm{VD}^{\mathrm{HySime}}$ also follow the same approach. However, there are six key differences between $\mathrm{VD}^{\mathrm{OSP}}$ and $\mathrm{VD}^{\mathrm{SSE/HySime}}$ described below.

1. The projection matrix used in the $\mathrm{VD}^{\mathrm{OSP}}$ is $\mathbf{P}_p = \mathbf{S}_p\left(\mathbf{S}_p^T \mathbf{S}_p\right)^{-1}\mathbf{S}_p^T$ as opposed to the projection matrix used in $\mathrm{VD}^{\mathrm{SSE/HySime}}$, $\hat{\mathbf{P}}_p = \mathbf{S}_p \mathbf{S}_p^T$ where the former is an OSP operator, while the latter is not.
2. $\mathrm{VD}^{\mathrm{OSP}}$ is derived from $\mathbf{P}_p \boldsymbol{\mu} = \mathbf{S}_p \bar{\alpha}_p$ in $\langle \mathbf{S}_p \rangle$ compared to the error $\hat{\mathbf{P}}_p \mathbf{K}_{\bar{\mathbf{n}}}/N$ resulting from the noise covariance matrix $\mathbf{K}_{\bar{\mathbf{n}}}$ in $\mathrm{VD}^{\mathrm{SSE/HySime}}$.
3. The noise in $\mathrm{VD}^{\mathrm{SSE/HySime}}$ is assumed to be Gaussian, whereas the noise in $\mathrm{VD}^{\mathrm{OSP}}$ does not make assumption of Gaussianity. As shown in Harsanyi and Chang (1994), the OSP

does not make Gaussian assumption on the noise and Gaussian noise has little effect on its performance. As a result, the well-known Gaussian maximum likelihood estimator used for linear spectral unmixing becomes a special case of the OSP classifier. With the same interpretation, $\mathrm{VD}^{\mathrm{SSE/HySime}}$ also turns out to be not as general as $\mathrm{VD}^{\mathrm{OSP}}$ due to the use of Gaussian noise.

4. $\mathrm{VD}^{\mathrm{SSE/HySime}}$ is developed based on minimization of the error caused by the use of the linear model in (5.49) in which case the model error is assumed to be described by a Gaussian noise, whereas $\mathrm{VD}^{\mathrm{OSP}}$ is derived by minimizing the number of signatures to be used to best represent the data in a linear form, that is, linear regression model. The error is measured by the principle of orthogonality, not Gaussian noise used in the SSE/HySime.
5. The most important difference is that like the HFC method $\mathrm{VD}^{\mathrm{SSE/HySime}}$ does not provide a constructive algorithm to find signals that form the signal subspace. It produces a fixed single value for the image data regardless of applications. On the contrary, $\mathrm{VD}^{\mathrm{OSP}}$ is determined by two key parameters, the error threshold ε and the algorithm to be used to find p VEs $\{\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_p\}$. By adjusting the error threshold ε and the algorithm to be used to find the spectrally distinct signatures $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{algorithm}}(\varepsilon)$ varies. In other words, $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{algorithm}}(\varepsilon)$ can produce different values determined by different applications, such as the number of endmembers, the number of man-made signatures, and the number of anomalies, all of which are supposed to have different values. Compared to $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{algorithm}}(\varepsilon)$ that varies with applications $\mathrm{VD}^{\mathrm{SSE/HySime}}$ is a constant value for all different applications. So, from an exploitation point of view, the SSE/HySime is not a practical criterion.
6. Finally, it is worth noting that the three VD estimation techniques, $\mathrm{VD}^{\mathrm{HFC/NWHFC}}$, $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{algorithm}}(\varepsilon)$, and $\mathrm{VD}^{\mathrm{SSE/HySime}}$, belong to three different categories. The $\mathrm{VD}^{\mathrm{HFC/NWHFC}}$ is indeed $\mathrm{VD}^{\mathrm{HFC/NWHFC}}(P_F)$, which is specified by the error threshold parameter, false alarm probability, P_F. On the other hand, the value of $\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{algorithm}}(\varepsilon)$ is a function of an application that involves two parameters, error threshold ε and the algorithm to be used to find signal sources in $\mathbf{S}_p$ compared to $\mathrm{VD}^{\mathrm{SSE/HySime}}$ whose value is fixed at a constant value regardless of applications.

5.5 Synthetic Image Experiments

The synthetic images used for experiments conducted in this section are the six scenarios (three TI scenarios, TI1, TI2, TI3, and three TE scenarios, TE1, TE2, TE3) designed in Chapter 4. A major advantage of using these synthetic images is to allow users to simulate complete ground truth to evaluate the effectiveness of algorithms to be designed and developed for quantitative study and comparative analysis. In these six simulated scenarios there are 100 pure panel pixels, 20 mixed panel pixels, and 10 subpixel panels simulated by five mineral signatures, alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M) in Figure 1.12(c) along with image background simulated by the sample mean. These 130 panel pixels are either implanted into the image background to simulate the three scenarios TI1, TI2, and TI3, or embedded into the image background to simulate the three scenarios TE1, TE2, and TE3. So, technically speaking, there are only five spectrally distinct pure signatures and one spectrally distinct mixed background signature used to simulate synthetic images.

5.5.1 Data Characterization-Driven Criteria

The data characterization-driven criteria developed in Section 5.3 for VD estimation can be classified into two categories.

1. *Category I*: criteria involving no parameters and producing a single constant value regardless of applications

$$\mathrm{VD}_{\mathrm{RE}}^{\mathrm{FA}}, \mathrm{VD}_{\mathrm{IE}}^{\mathrm{FA}}, \mathrm{VD}_{\mathrm{XE}}^{\mathrm{FA}}, \mathrm{VD}_{\mathrm{EIF}}^{\mathrm{FA}}, \mathrm{VD}_{\mathrm{AIC}}^{\mathrm{ITC}}, \mathrm{VD}_{\mathrm{MDL}}^{\mathrm{ITC}}.$$

2. *Category II*: criteria involving one parameter, either error threshold ε or false alarm probability P_F.

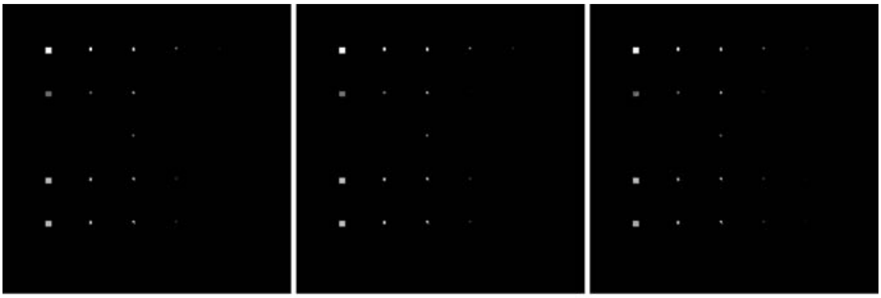
$$\mathrm{VD}_{\lambda}^{\mathrm{eigen}}(a\%), \mathrm{VD}_{\lambda}^{\mathrm{eigen}}(\varepsilon), \mathrm{VD}\hat{\lambda}^{\ \mathrm{eigen}}(\varepsilon), \mathrm{VD}_{\hat{\lambda}-\lambda}^{\mathrm{eigen}}(\varepsilon), \mathrm{VD}_{\lambda}^{\mathrm{SVD}}(\varepsilon), \mathrm{VD}_{\hat{\lambda}}\mathrm{SVD}(\varepsilon), \mathrm{VD}_{\hat{\lambda}-\lambda}^{\mathrm{SVD}}(\varepsilon),$$
$$\mathrm{VD}_{\mathrm{disk}}^{\mathrm{GR}}(\varepsilon), \mathrm{VD}_{\mathrm{diff}}^{\mathrm{GR}}(\varepsilon), \mathrm{VD}_{\mathrm{HFC}}^{\mathrm{NP}}(P_F), \mathrm{VD}_{\mathrm{NWHFC}}^{\mathrm{NP}}(P_F).$$

5.5.1.1 Target Implantation (TI) Scenarios

The 130 target panel pixels in the TI scenarios are implanted into the image background simulated by the sample mean of the real Cuprite image scene in such a way that the background pixels are removed to accommodate these implanted target panel pixels. As a consequence, the 10 subpixel panels implanted in the three scenarios TI1, TI2, and TI3 shown in Figure 5.2 are nearly invisible. Since TI1 is a scenario of clean target panel pixels implanted in a clean image background and is the simplest case with no noise involvement, several criteria that use noise estimation will not be applicable such as AIC and MDL. TI2 is a scenario that has clean target panel pixels implanted in the image. So, technically speaking, TI2 has only five spectrally distinct pure signatures and one spectrally distinct background signature that is a mixed signature in the scene in spite of the fact that it has 100 pure panel pixels, 20 mixed panel pixels, and 10 subpixel panels. On the other hand, TI3 is a scenario with an additive Gaussian noise to be added to scenario TI1. So, TI3 has no pure target panel pixels because all the panel pixels and background signature are corrupted by the added Gaussian noise.

Table 5.1 calculates the values of VD estimated by the criteria in Category I where an N/A indicates a case to which a particular criterion is not applicable since scenario TI1 does not have simulated noise but this criterion requires noise to perform noise estimation.

An interesting observation can be made from Table 5.1. For noise-free cases, that is, scenario TI1, the four FA-based criteria, $\mathrm{VD}_{\mathrm{RE}}^{\mathrm{FA}}$, $\mathrm{VD}_{\mathrm{IE}}^{\mathrm{FA}}$, $\mathrm{VD}_{\mathrm{XE}}^{\mathrm{FA}}$, and $\mathrm{VD}_{\mathrm{EIF}}^{\mathrm{FA}}$, worked very effectively to



(a) Scenario TI1

(b) Scenario TI2

(c) Scenario TI3

Figure 5.2 Three TI scenarios: TI1, TI2 and TI3 for TI.

Table 5.1 VD estimated for six scenarios by criteria in Category I

	$\text{VD}_{\text{RE}}^{\text{FA}}$	$\text{VD}_{\text{IE}}^{\text{FA}}$	$\text{VD}_{\text{XE}}^{\text{FA}}$	$\text{VD}_{\text{EIF}}^{\text{FA}}$	$\text{VD}_{\text{AIC}}^{\text{ITC}}$	$\text{VD}_{\text{MDL}}^{\text{ITC}}$
TI1	5	5	5	5	N/A	N/A
TI2	188	1	188	3	1	1
TI3	188	1	188	3	1	1

produce exactly five pure spectrally distinct mineral signatures in the way for which they are designed. However, when there is noise present in the simulated image data, these four criteria joined with $\text{VD}_{\text{AIC}}^{\text{ITC}}$, $\text{VD}_{\text{MDL}}^{\text{ITC}}$ faild to produce correct values for VD.

Table 5.2 also calculates the values of VD estimated by the criteria in Category II with a parameter specified by either energy percentage $a\%$ or an error threshold ε or false alarm probability P_F.

As shown in Table 5.2, the best criterion was the one produced by the NWHFC method regardless of P_F.

5.5.1.2 Target Embeddedness (TE) Scenarios

Unlike TI scenarios TE scenarios insert all the 130 target panel pixels in the image background as embedded pixels shown in Figure 5.3 by being superimposed atop background pixels. As a result, none of these target panel pixels are pure because they are mixed by background pixels.

Compared to the three TI scenarios in Figure 5.2, all the 130 panel pixels are nearly visible in Figure 5.3 due to the superimposition of panel pixels over background pixels. Since the target panel pixels are added to superimpose over background pixels, the background signature should be considered as the sixth spectrally distinct signature. In this case, in all the three TE scenarios there should have six spectrally distinct signatures, but none of them is pure. This is the key difference between TI scenarios and TE scenarios. Similar to Tables 5.1 and 5.2, Tables 5.3 and 5.4 calculate the values of VD estimated by criteria in Categories I and II.

Like scenario TI1 the four FA-based criteria, $\text{VD}_{\text{RE}}^{\text{FA}}$, $\text{VD}_{\text{IE}}^{\text{FA}}$, $\text{VD}_{\text{XE}}^{\text{FA}}$ and $\text{VD}_{\text{EIF}}^{\text{FA}}$, actually performed effectively for noise-free scenario TE1 in the way they are designed for by producing six spectrally distinct signatures. Once again, the best ones were still NP detection-based criteria, HFC with $\text{VD} = 6$ to represent six spectrally distinct signatures and NWHFC with $\text{VD} = 5$ to indicate that there are five purest signatures in the scene.

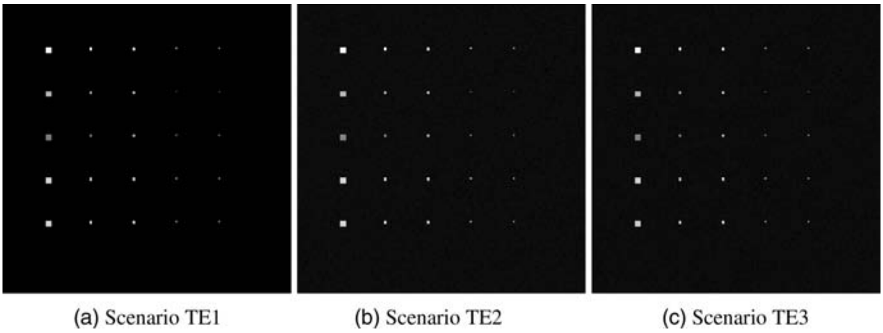

(a) Scenario TE1 (b) Scenario TE2 (c) Scenario TE3

Figure 5.3 Three TE scenarios: TE1, TE2 and TE3.

Table 5.2 VD estimated for three TI scenarios by criteria in Category II

$a\%$	TI scenarios	99	98	97	96	95
$\mathrm{VD}_{\lambda}^{\mathrm{eigen}}(a\%)$	TI1	3	3	2	2	2
	TI2	176	169	164	159	154
	TI3	176	169	164	159	154
ε	TI scenarios	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}
$\mathrm{VD}_{\hat{\lambda}-\lambda}^{\mathrm{eigen}}(\varepsilon)$	TI1	1	1	1	1	1
	TI2	1	1	1	1	1
	TI3	1	1	1	1	1
$\mathrm{VD}_{\hat{\lambda}\text{-}\lambda}^{\mathrm{SVD}}(\varepsilon)$	TI1	1	1	1	1	1
	TI2	1	1	1	1	1
	TI3	1	1	1	1	1
$\mathrm{VD}_{\lambda}^{\mathrm{eigen}}(\varepsilon)$	TI1	3	2	1	1	1
	TI2	3	2	1	1	1
	TI3	3	2	1	1	1
$\mathrm{VD}_{\lambda}^{\mathrm{SVD}}(\varepsilon)$	TI1	6	6	6	2	1
	TI2	4	4	3	2	1
	TI3	4	4	3	2	1
$\mathrm{VD}_{\lambda}^{\mathrm{eigen}}(\varepsilon)$	TI1	6	5	5	5	3
	TI2	22	4	4	3	2
	TI3	189	4	4	3	2
$\mathrm{VD}_{\lambda}^{\mathrm{SVD}}(\varepsilon)$	TI1	5	5	5	5	5
	TI2	22	4	4	3	1
	TI3	189	4	4	3	1
ε	TI scenarios	10^{0}	10^{1}	10^{2}	10^{3}	10^{4}
$\mathrm{VD}_{\mathrm{disk}}^{\mathrm{GR}}(\varepsilon)$	TI1	6	6	6	6	3
	TI2	188	188	188	185	22
	TI3	188	188	188	185	23
$\mathrm{VD}_{\mathrm{diff}}^{\mathrm{GR}}(\varepsilon)$	TI1	6	6	6	6	2
	TI2	188	184	4	3	2
	TI3	188	184	4	3	2
P_F	TI scenarios	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
$\mathrm{VD}_{\mathrm{HFC}}^{\mathrm{NP}}(P_F)$	TI1	99	67	59	47	39
	TI2	7	4	3	3	3
	TI3	7	4	3	3	3
$\mathrm{VD}_{\mathrm{NWHFC}}^{\mathrm{NP}}(P_F)$	TI1	117	83	66	52	48
	TI2	5	5	5	5	5
	TI3	5	5	5	5	5
$\mathrm{VD}_{\mathrm{NSP}}^{\mathrm{NP}}(P_F)$	TI1	30	30	30	30	30
	TI2	78	72	67	62	59
	TI3	78	72	68	62	59

Table 5.3 VD estimated for six scenarios by criteria in Category I

	$\mathrm{VD}_{\mathrm{RE}}^{\mathrm{FA}}$	$\mathrm{VD}_{\mathrm{IE}}^{\mathrm{FA}}$	$\mathrm{VD}_{\mathrm{XE}}^{\mathrm{FA}}$	$\mathrm{VD}_{\mathrm{EIF}}^{\mathrm{FA}}$	$\mathrm{VD}_{\mathrm{AIC}}^{\mathrm{ITC}}$	$\mathrm{VD}_{\mathrm{MDL}}^{\mathrm{ITC}}$
TE1	6	6	6	6	N/A	N/A
TE2	188	1	188	3	1	1
TE3	188	1	188	3	1	1

Table 5.4 VD estimated for three TE scenarios by criteria in Category II

$a\%$	TE scenarios	99	98	97	96	95
$\mathrm{VD}_{\lambda}^{\mathrm{eigen}}(a\%)$	TE1	1	1	1	1	1
	TE2	150	129	111	94	78
	TE3	150	129	111	94	79
ε	TE scenarios	10^{-6}	10^{-5}	10^{-4}	10^{-3}	10^{-2}
$\mathrm{VD}_{\hat{\lambda}-\lambda}^{\mathrm{eigen}}(\varepsilon)$	TE1	1	1	1	1	189
	TE2	1	1	1	1	1
	TE3	1	1	1	1	1
$\mathrm{VD}_{\hat{\lambda}\text{-}\lambda}^{\mathrm{SVD}}(\varepsilon)$	TE1	1	1	1	1	1
	TE2	1	1	1	1	1
	TE3	1	1	1	1	1
$\mathrm{VD}_{\lambda}^{\mathrm{eigen}}(\varepsilon)$	TE1	3	2	1	1	1
	TE2	3	2	1	1	1
	TE3	3	2	1	1	1
$\mathrm{VD}_{\hat{\lambda}\text{-}\lambda}^{\mathrm{SVD}}(\varepsilon)$	TE1	6	6	6	2	1
	TE2	4	4	3	2	1
	TE3	4	4	3	2	1
$\mathrm{VD}_{\lambda}^{\mathrm{eigen}}(\varepsilon)$	TE1	6	6	5	2	1
	TE2	4	4	3	2	1
	TE3	4	4	3	2	1
$\mathrm{VD}_{\lambda}^{\mathrm{SVD}}(\varepsilon)$	TE1	6	6	6	6	3
	TE2	189	4	4	3	1
	TE3	52	4	4	3	1
ε	TE scenarios	10^{0}	10^{1}	10^{2}	10^{3}	10^{4}
$\mathrm{VD}_{\mathrm{disk}}^{\mathrm{GR}}(\varepsilon)$	TE1	6	6	6	6	3
	TE2	188	188	188	185	22
	TE3	188	188	188	185	23
$\mathrm{VD}_{\mathrm{diff}}^{\mathrm{GR}}(\varepsilon)$	TE1	6	6	6	6	2
	TE2	188	179	164	2	2
	TE3	188	179	164	2	2
P_F	TE scenarios	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
$\mathrm{VD}_{\mathrm{HFC}}^{\mathrm{NP}}(P_F)$	TE1	7	6	6	6	5
	TE2	7	5	5	3	2
	TE3	7	5	5	2	2
$\mathrm{VD}_{\mathrm{NWHFC}}^{\mathrm{NP}}(P_F)$	TE1	6	5	5	5	5
	TE2	5	5	5	5	5
	TE3	5	5	5	5	5
$\mathrm{VD}_{\mathrm{NSP}}^{\mathrm{NP}}(P_F)$	TE1	34	34	33	33	33
	TE2	77	71	66	62	57
	TE3	76	71	66	61	58

Table 5.5 VD estimated for synthetic image by SSE and HySime

	SSE	HySime
TI2	2	5
TI3	2	5
TE2	6	5
TE3	6	5

5.5.2 Data Representation-Driven Criteria

Analogous to data characterization-driven criteria, SSE/HySime corresponds to Category I that produces a single constant value for the data regardless of applications, while the OSP-based criteria belong to a third type, Category III, that includes criteria producing the values of VD determined by two parameters, error threshold ε and application-based algorithm, used to find signal sources. When one of two parameters is fixed at a constant value, the criteria in Category III are reduced to criteria in Category II. Similarly, if both parameters are fixed at constant values the criteria in Category III become criteria in Category I. As noted in scenarios TI1 and TE1, there is no simulated noise in the scenes to account for model error in linear data representation specified by (5.49) in which case data representation-driven criteria are not applicable. Therefore, TI1 and TE1 are not considered in this section. Table 5.5 calculates the values of VD estimated by SSE and HySime where the HySime seemed to work better than the SSE by producing VD = 5, which represents five pure signatures in TI scenarios and five most purest signatures in TE scenarios.

Table 5.6 tabulates the values of VD estimated by OSP-based methods with two gradient and difference criteria using various algorithms, ATGP, UNCLS, UFCLS, N-FINDR, SGA, and VCA, where the values in parentheses were the values of error threshold ε used in (5.54) and (5.55).

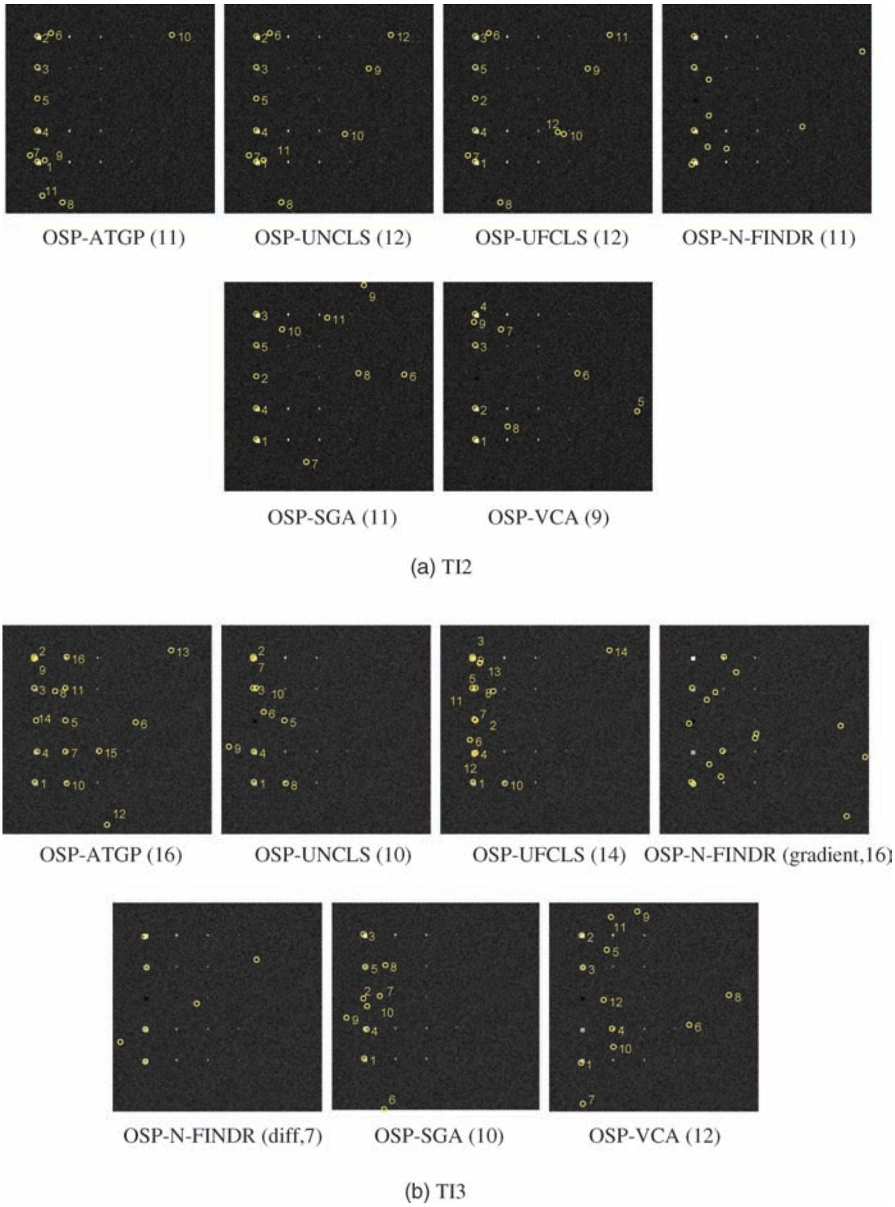
Comparing Table 5.6 to Table 5.5, OSP-based methods seemed to produce higher numbers of signal sources than SSE/HySime do.

As expected, different algorithms used by OSP-based methods generated different values of VD in Table 5.6, which were difficult to be used for performance evaluation due to the lack of information of extracted signal sources by different algorithms. These extracted pixels

Table 5.6 VD estimated for synthetic image by OSP

		OSP- SVD	OSP- ATGP	OSP- UNCLS	OSP- UFCLS	OSP- NFINDR	OSP- SGA	OSP- VCA
TI2	Gradient	4 (−120)	9 (−70)	12 (−60)	12 (−60)	11 (−60)	11 (−55)	8 (−60)
	Diff	4 (−120)	11 (−50)	11 (−50)	12 (−50)	11 (−50)	11 (−50)	9 (−60)
TI3	Gradient	4 (−120)	16 (−70)	10 (−70)	14 (−70)	16 (−50)	8 (−50)	12 (−60)
	Diff	4 (−120)	8 (−70)	6 (−70)	14 (−70)	7 (−50)	10 (−50)	12 (−60)
TE2	Gradient	6 (−120)	8 (−60)	12 (−60)	14 (−60)	8 (−50)	11 (−60)	12 (−60)
	Diff	4 (−120)	8 (−50)	10 (−50)	15 (−60)	10 (−50)	2 (−60)	11 (−60)
TE3	Gradient	4 (−110)	5 (−60)	13 (−70)	19 (−70)	7 (−50)	8 (−60)	6 (−60)
	Diff	4 (−110)	15 (−70)	13 (−70)	8 (−70)	10 (−50)	2 (−60)	10 (−60)

provided crucial information on applications to which these algorithms were applied. In order to address this issue, an application in endmember extraction is considered for illustration. Figure 5.4(a)–(d) lists all the pixels extracted by ATGP, UNCLS, UFCKS, N-FINDR, SGA, and VCA for scenarios TI2, TI3, TE2, and TE3, respectively, where the number used to mark a pixel



OSP-ATGP (11) OSP-UNCLS (12) OSP-UFCLS (12) OSP-N-FINDR (11)

OSP-SGA (11) OSP-VCA (9)

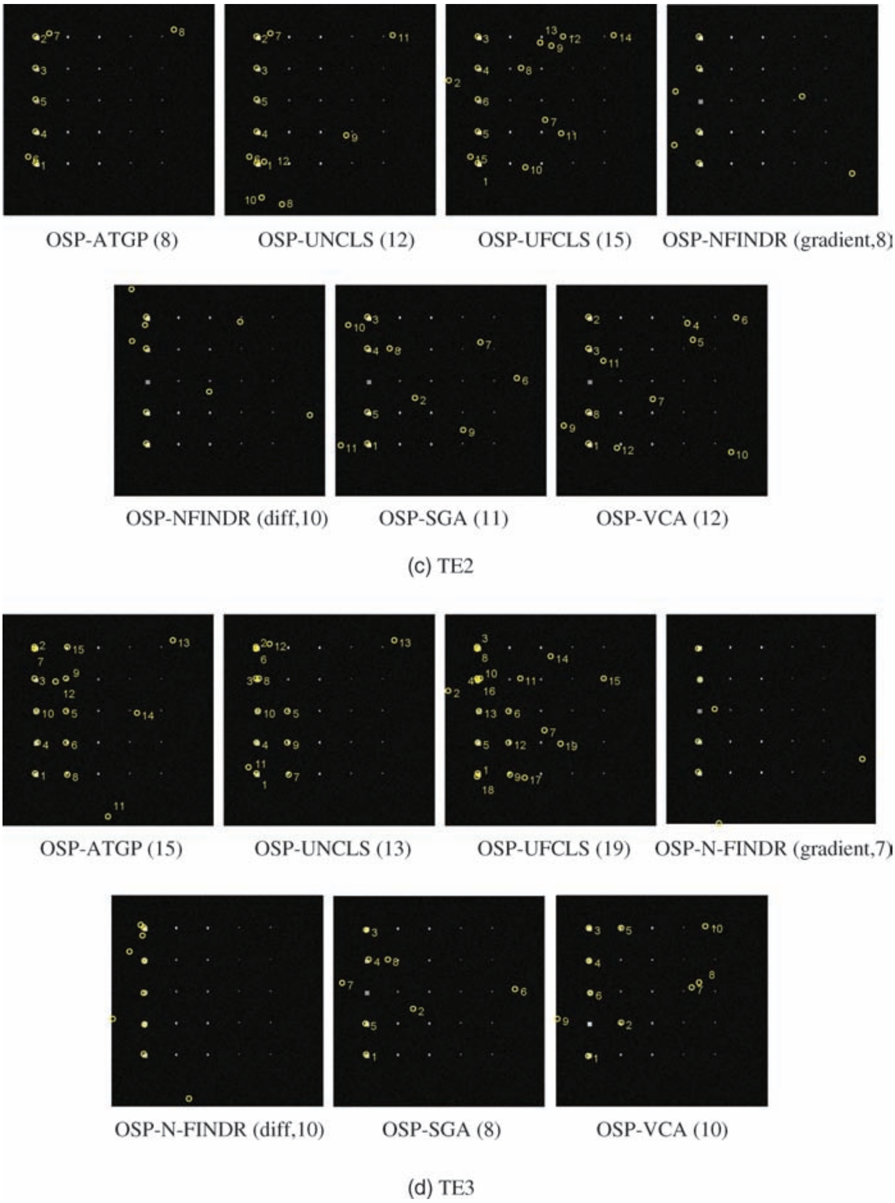
(a) TI2

OSP-ATGP (16) OSP-UNCLS (10) OSP-UFCLS (14) OSP-N-FINDR (gradient,16)

OSP-N-FINDR (diff,7) OSP-SGA (10) OSP-VCA (12)

(b) TI3

Figure 5.4 Pixels extracted by OSP-ATGP, OSP-UNCLS, OSP-UFCLS, OSP-N-FINDR, OSP-SGA, and OSP-VCA.



OSP-ATGP (8) OSP-UNCLS (12) OSP-UFCLS (15) OSP-NFINDR (gradient,8)

OSP-NFINDR (diff,10) OSP-SGA (11) OSP-VCA (12)

(c) TE2

OSP-ATGP (15) OSP-UNCLS (13) OSP-UFCLS (19) OSP-N-FINDR (gradient,7)

OSP-N-FINDR (diff,10) OSP-SGA (8) OSP-VCA (10)

(d) TE3

Figure 5.4 (*Continued*)

was the order in which the particular pixel is extracted. Since N-FINDR extracted all the pixels together at the same time, there were no numbers assigned to extracted pixels. In particular, in TI2 both gradient and difference produced the same number of signatures, 11. So, there is only one figure shown to represent both cases.

Table 5.7 SAM and SID measure of six signatures in Figure 5.5(a)

SAM SID	A	B	C	K	M	Sample mean
A	0	0.1576	0.2038	0.0961	0.1421	0.1822
B	0.0320	0	0.0951	0.1733	0.0968	0.0656
C	0.0496	0.0097	0	0.2114	0.1108	0.0441
K	0.0126	0.0457	0.0600	0	0.1263	0.1923
M	0.0231	0.0115	0.0142	0.0232	0	0.0924
Sample mean	0.0405	0.0046	0.0025	0.0522	0.0103	0

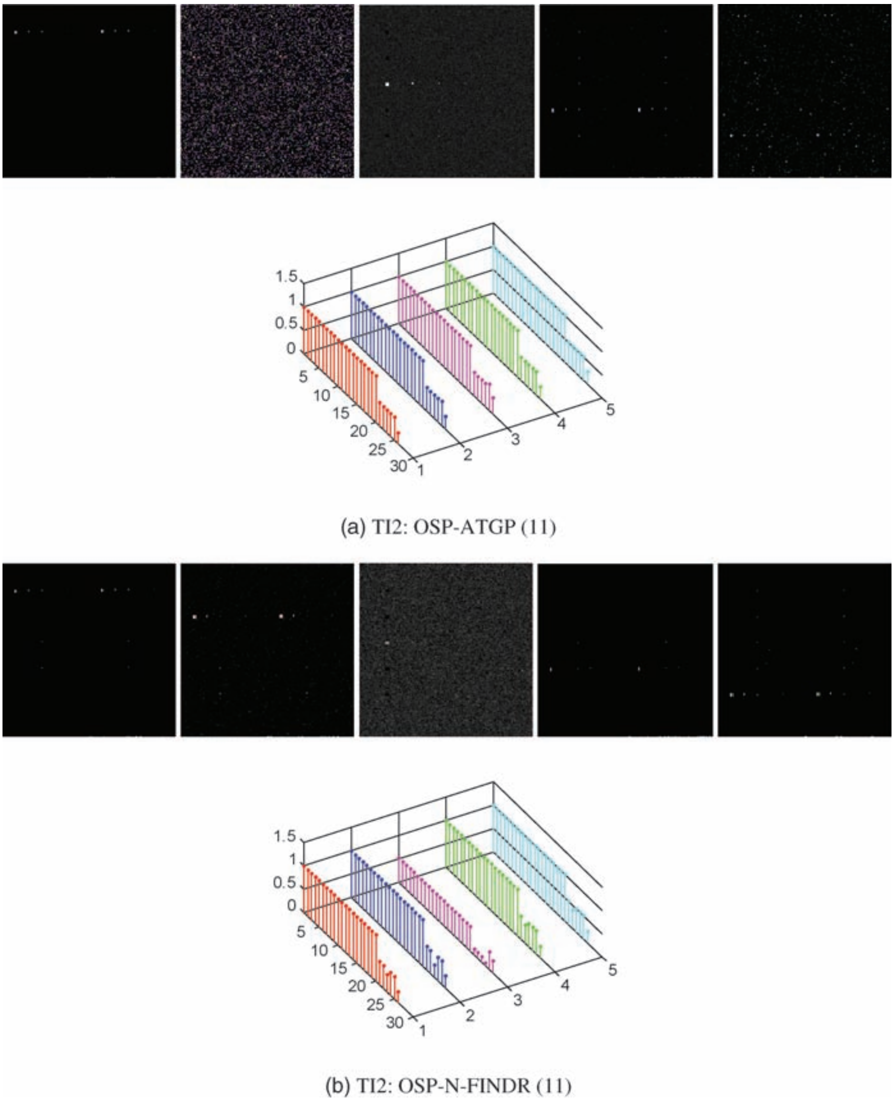
As shown in Figure 5.4, the best OSP-based methods were those using ATGP and LSMA algorithms, UNCLS and UFCLS, in all TI and TE scenarios that could extract panel pixels that corresponded to all five distinct mineral signatures, while some endmember extraction algorithms (EEAs) failed to extract panel pixels in row 3 corresponding to the mineral signature calcite, for example, OSP-N-FINDR, using gradient and difference criteria and OSP-VCA in scenario TI2. The reason for EEAs missing the calcite signature was the fact that the calcite has its spectral signature very close and similar to that of the sample mean used to simulate the image background as shown in Figure 4.12(a) and (b). As a consequence, EEAs considered the calcite as a contaminated sample mean signature. If the background signature was extracted first, the calcite would not be extracted afterward as a distinct endmember. To further confirm this finding, the upper and lower triangular arrays in Table 5.7 tabulate the similarity values among these five mineral signatures and the sample mean calculated by the SAM and SID, respectively, where the calcite and the sample mean have closest signatures with smallest values.

Nevertheless, if there is a panel pixel extracted by an EEA to specify one of the five mineral signatures, it was always extracted earlier than later. This indicated that panel pixels corresponding to pure or purest signatures were more significant than other pixels. Accordingly, they were always among those pixels that were extracted first.

As another application, linear spectral unmixing was considered for experiments. Since ATGP, UNCLS, and UFCLS represent the same class of target detection algorithms compared to N-FINDR, SGA, and VCA that belong to the same class of endmember extraction algorithms, ATGP and N-FINDR were selected as their respective representatives and their extracted pixels were used to form desired signature matrices **M** in (2.75) for supervised linear hyperspectral unmixing. Figure 5.5 shows the FCLS-unmixed results along with plots of their quantification results for scenarios TI2, TI3, TE2, and TE3 using the pixels extracted in Figure 5.4 by OSP-ATGP and OSP-N-FINDR. As noted above, the numbers of pixels extracted by OSP-N-FINDR using gradient and difference criteria were different. In this case, the larger number was selected to extract pixels by N-FINDR to achieve better background suppression, that is, TI2 (11 by both), TI3 (16 by gradient), TE2 (10 by difference), and TE3 (10 by difference).

Several interesting observations can be made by examining the results in Figure 5.5, which have shown great benefits of using synthetic images that cannot be obtained from real image experiments.

1. *TI2:* Although the OSP-N-FINDR missed extraction of the calcite signature, the FCLS-unmixed results using the OSP-N-FINDR extracted pixels demonstrated that the panel pixels in row 3



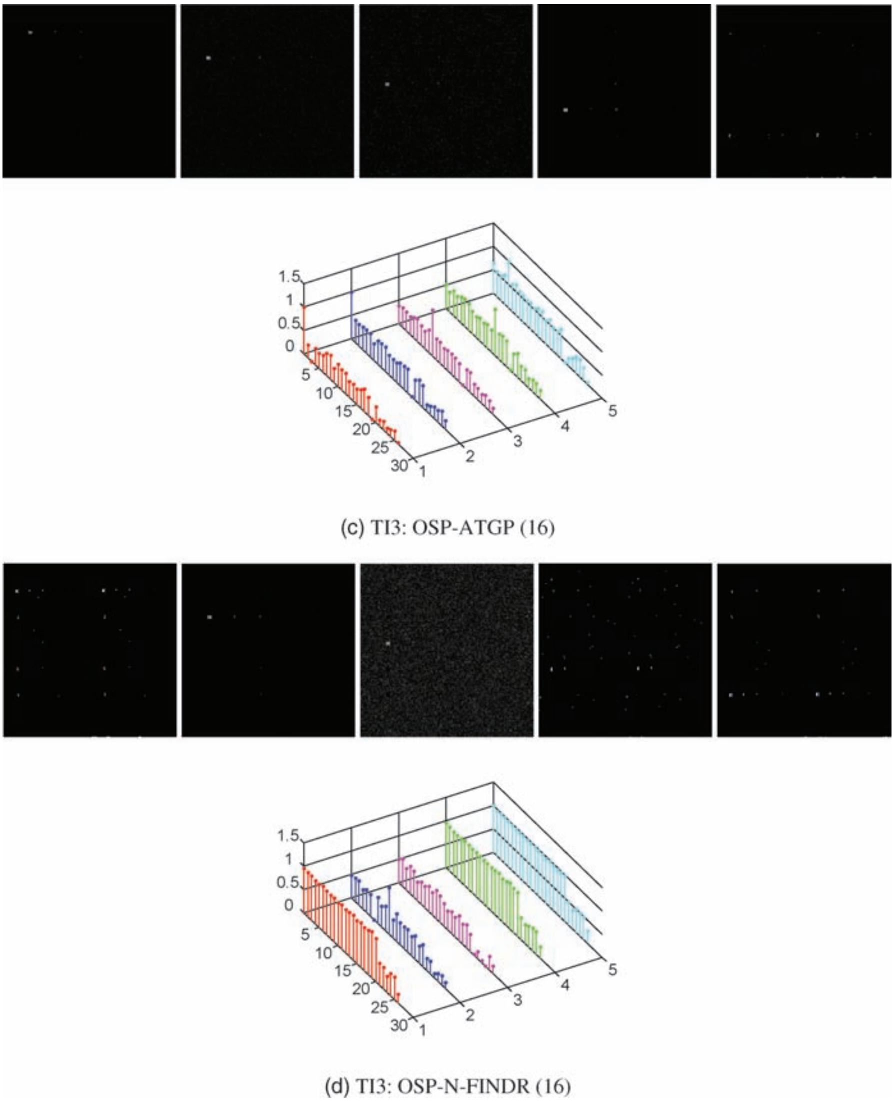
(a) TI2: OSP-ATGP (11)

(b) TI2: OSP-N-FINDR (11)

Figure 5.5 FCLS-unmixed results for panel pixels in five rows using pixels extracted by OSP-ATG and OSP-N-FINDR.

specified by the calcite signature could be still classified effectively even though only half of abundance fractions in 20 pure panel pixels in the first two columns and small amounts of abundance fractions in mixed and subpixel panels were detected.

2. *TI3:* This is the TI1 scenario corrupted by a Gaussian noise. Despite that both OSP-ATGP and OSP-N-FINDR extracted the same number of signatures, 16, interestingly, most of their extracted pixels were quite different. As a consequence, their FCLS-unmixed results were also very different.

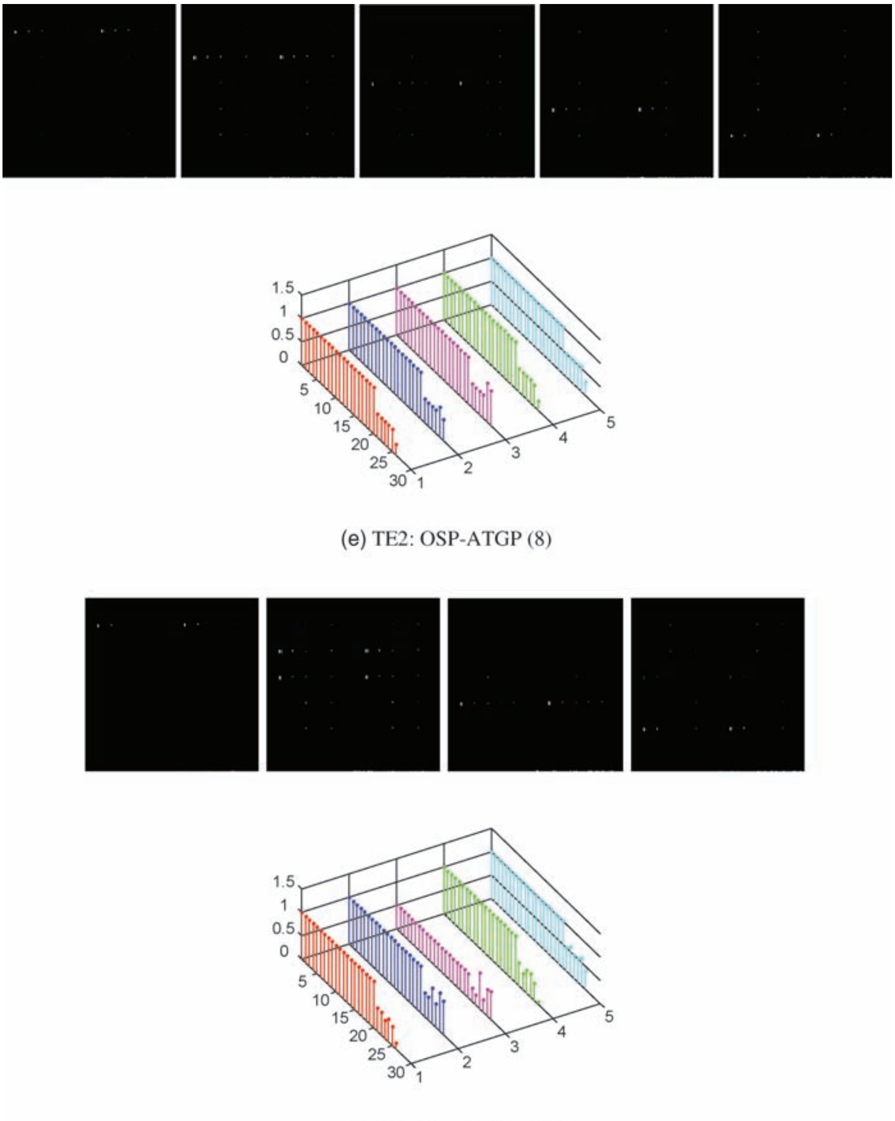


(c) TI3: OSP-ATGP (16)

(d) TI3: OSP-N-FINDR (16)

Figure 5.5 (*Continued*)

3. *TE2:* This scenario does not satisfy ASC constraint. As expected, the quantification of FCLS-unmixed results would not be correct. However, the results in Figure 5.5(e) show a surprising result that the FCLS using the 8 pixels extracted by OSP-ATGP produced nearly correct quantification results except subpanel pixels. On the other hand, FCLS using 10 pixels extracted by OSP-N-FINDR produced worse results in Figure 5(f) than those in Figure 5(e), which only used 8 OSP-ATGP-generated pixels. These experiments demonstrated that using more pixel information did not necessarily improve results.
4. *TE3:* The results obtained from this scenario were completely opposite to those of TE2 scenario with one conclusion in common. That is, more pixels did not provide more useful information but rather conflicting information that may further deteriorate performance.



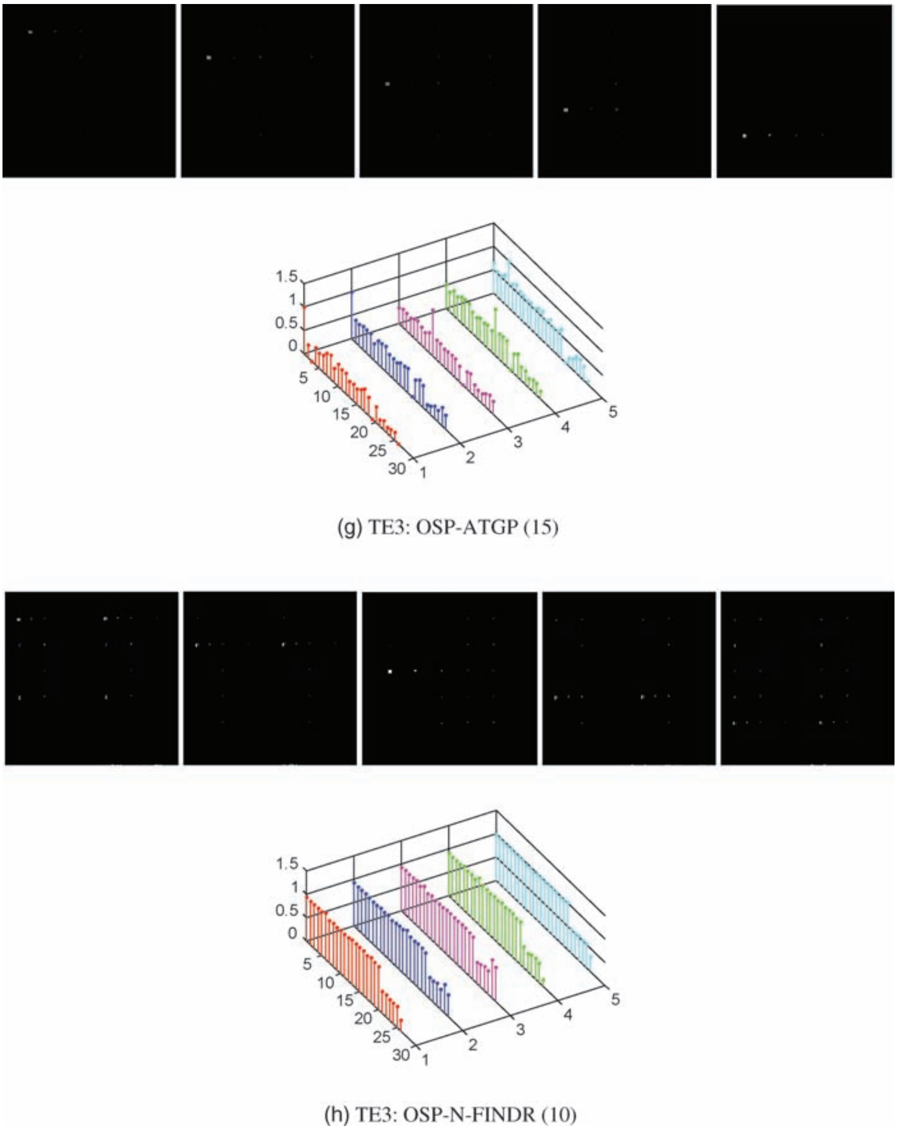
(e) TE2: OSP-ATGP (8)

(f) TE2: OSP-N-FINDR (10)

Figure 5.5 (*Continued*)

5.6 VD Estimated for Real Hyperspectral Images

To conclude this chapter, this section presents experiments based on real image scenes described in Section 1.7 of Chapter 1. According to the experimental results for synthetic images in Section 5.5, only the HFC and NWHFC methods from data characterization-driven criteria have been shown to be effective. Therefore, Table 5.8 tabulates the values of VD estimated by the HFC and NWHFC



(g) TE3: OSP-ATGP (15)

(h) TE3: OSP-N-FINDR (10)

Figure 5.5 (*Continued*)

methods for three AVIRIS data sets: Cuprite data (reflectance and radiance data), Purdue data with/out background (BKG), LCVF data, plus a 15-panel HYDICE data.

As for data representation-driven criteria Table 5.9 tabulates the values of VD estimated by SSE and HySime for Cuprite data (reflectance and radiance data), Purdue data with/out background (BKG), LCVF data, and 15-panel HYDICE data.

Comparing VD estimated by HFC/NWHFC in Table 5.8 to the VD estimated by SSE/HySime in Table 5.9, the SSE/HySime-estimated values were found to be within the ranges of

Table 5.8 VD estimated for real images by HFC and NWHFC

	P_F	10^{-1}	10^{-2}	10^{-3}	10^{-4}
HYDICE	HFC	14	11	9	9
	NWHFC	20	14	13	13
Cuprite Reflectance	HFC	34	30	24	22
	NWHFC	32	29	25	23
Cuprite Radiance	HFC	29	18	17	15
	NWHFC	23	20	20	19
Purdue with BKG	HFC	72	42	34	29
	NWHFC	19	18	18	18
Purdue without BKG	HFC	55	33	30	24
	NWHFC	18	18	18	15
LCVF	HFC	8	6	4	4
	NWHFC	20	13	11	11

HFC/NWHFC-estimated VD values except the Purdue data with P_F set to around 10^{-3}. It is known that the samples in Purdue data are heavily mixed. It makes sense that various VD estimation techniques produce a wide range of VD. Furthermore, Table 5.10 tabulates the values of VD estimated by OSP-based methods with two different criteria for real images using various algorithms where the chosen thresholds using dB unit are provided in the parentheses. For example, ATGP, UNCLS, and UFCLS were used to find VD, OSP-ATGP, OSP-UNCLS, and OSP-UFCLS provided estimates of the number of spectrally distinct signal sources for data representation. On the other hand, when N-FINDR, SGA, and VCA were used to find VD, OSP-N-FINDR, OSP-SGA, and OSP-VCA were used to estimate the number of endmembers in the data.

Comparing the results in Table 5.10 to the results in Tables 5.8 and 5.9 VD-estimated values by OSP-based methods were generally found to be higher than those by HFC/NWHFC and SSE/HySime due to the fact that the former were specifically designed to find the number of spectrally distinct signatures for LSMA in which case signal sources considered to be VEs were generally more endmembers than required for endmember extraction, while the latter were designed to estimate VD without considering any specific application.

Unlike the synthetic images considered in Section 5.5 that have complete knowledge of the ground truth about the signatures real image data sets generally have only partial knowledge

Table 5.9 VD estimated for real images by SSE and HySime

	SSE	HySime
HYDICE	10	20
Cuprite reflectance	27	16
Cuprite radiance	18	17
Purdue with BKG	8	13
Purdue without BKG	7	13
LCVF	12	11

Table 5.10 VD estimated for real images by OSP

		OSP-SVD	OSP-ATGP	OSP-UNCLS	OSP-UFCLS	OSP-NFINDR	OSP-SGA	OSP-VCA
HYDICE	gradient	16 (−110)	23 (−70)	15 (−70)	11 (−70)	10 (−50)	9 (−60)	19 (−60)
	diff	16 (−110)	11 (−70)	15 (−70)	15 (−70)	10 (−50)	9 (−60)	14 (−60)
Cuprite Reflectance	gradient	22 (−110)	16 (−70)	26 (−70)	31 (−70)	13 (−50)	19 (−60)	21 (−60)
	diff	21 (−110)	17 (−70)	19 (−70)	13 (−70)	10 (−50)	17 (−70)	14 (−60)
Cuprite Radiance	gradient	11 (−110)	18 (−70)	21 (−70)	14 (−70)	13 (−50)	13 (−60)	18 (−60)
	diff	15 (−110)	13 (−70)	17 (−70)	14 (−70)	11 (−50)	6 (−70)	13 (−60)
Purdue with BKG	gradient	11 (−110)	18 (−70)	26 (−70)	18 (−70)	13 (−60)	13 (−60)	16 (−60)
	diff	8 (−110)	24 (−70)	19 (−70)	11 (−70)	15 (−60)	12 (−60)	16 (−60)
Purdue without BKG	gradient	9 (−110)	15 (−70)	10 (−70)	13 (−70)	16 (−60)	8 (−60)	15 (−60)
	diff	12 (−110)	13 (−70)	13 (−70)	20 (−70)	14 (−60)	8 (−60)	14 (−60)
LCVF	gradient	14 (−110)	7 (−60)	10 (−70)	11 (−60)	5 (−50)	15 (−60)	5 (−60)
	diff	5 (−110)	14 (−70)	14 (−70)	16 (−70)	10 (−50)	4 (−60)	8 (−60)

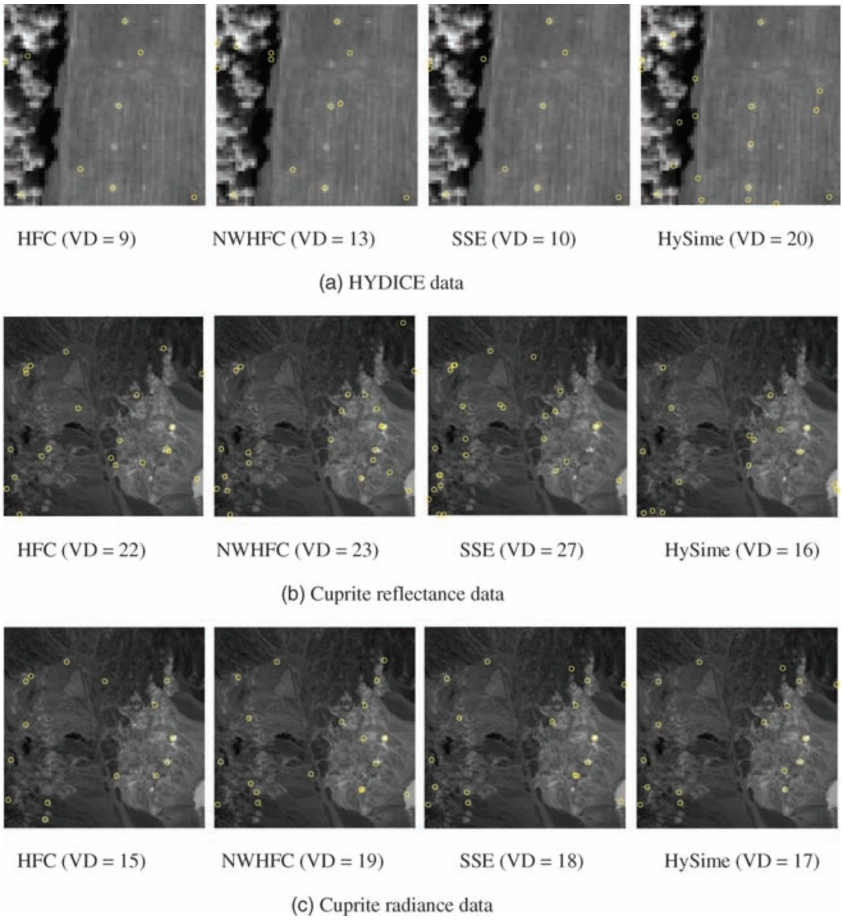
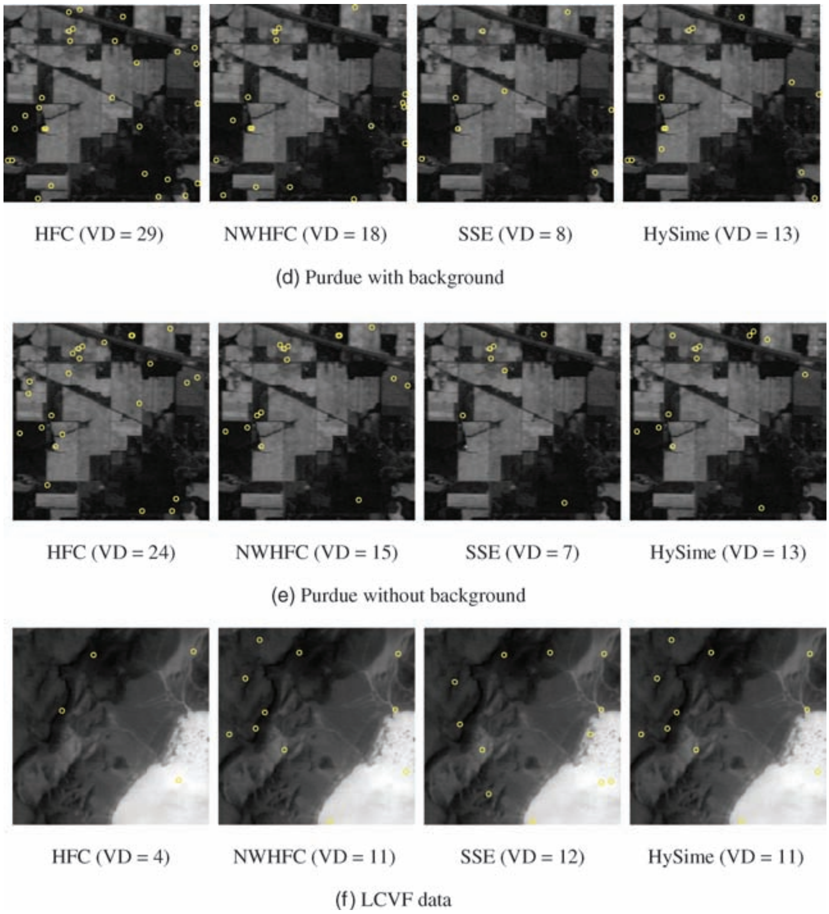


Figure 5.6 Pixels extracted by N-FINDR for the four image scenes, HYDICE, cuprite, Purdue, and LCVF.



HFC (VD = 29) NWHFC (VD = 18) SSE (VD = 8) HySime (VD = 13)

(d) Purdue with background

HFC (VD = 24) NWHFC (VD = 15) SSE (VD = 7) HySime (VD = 13)

(e) Purdue without background

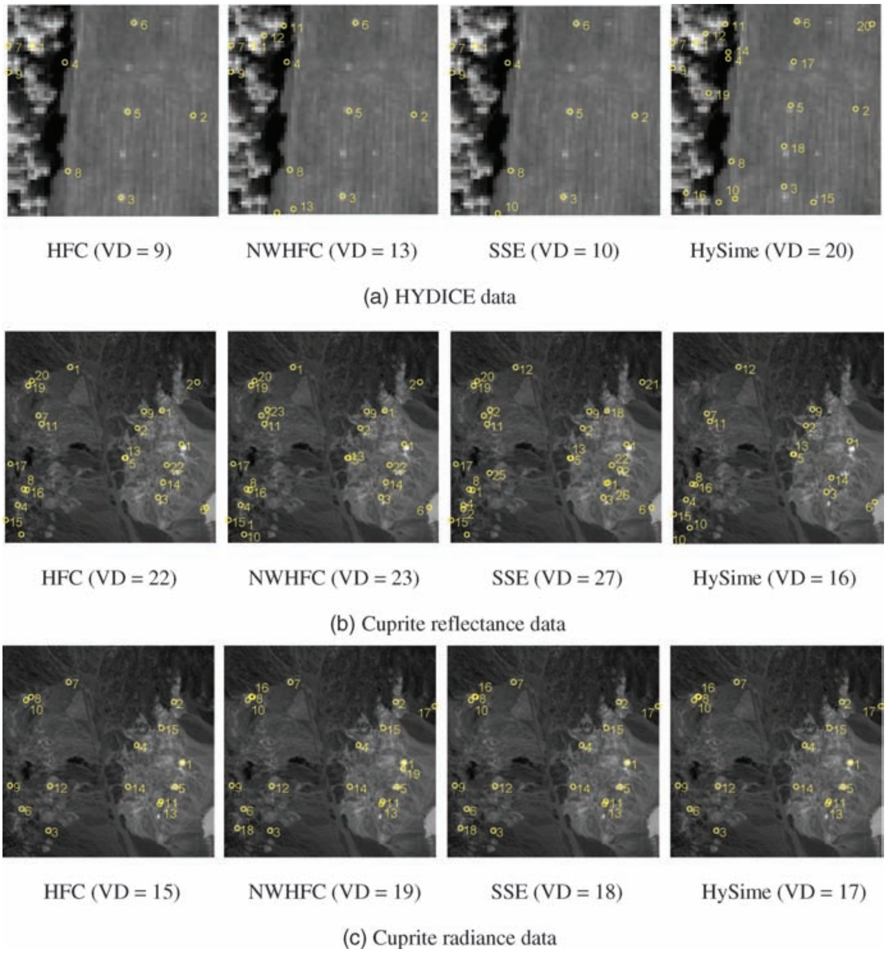
HFC (VD = 4) NWHFC (VD = 11) SSE (VD = 12) HySime (VD = 11)

(f) LCVF data

Figure 5.6 (*Continued*)

about signatures of interest from the ground truth. In this case, the true VD is never known. Consequently, it is difficult to conduct fair comparative analysis among various VD estimation techniques. Nevertheless, since the techniques are mainly developed from a perspective of data representation in a linear form, LSMA seems an appropriate application to be used for performance evaluation in which case FCLS is used to perform spectral unmixing and the least-squares error (LSE) is used as a criterion for quantitative analysis. Moreover, HFC/NWHFC and SSE/HySime do not provide any algorithm to find VEs. In this case, N-FINDR considered in Chapter 7 and ATGP are once again used to extract desired pixels to be used for FCLS. Figures 5.6 and 5.7 show pixels extracted by N-FINDR and ATGP where the P_F for HFC and NWHFC is chosen to be $P_F \leq 10^{-4}$.

Figure 5.8 also shows pixels extracted by OSP methods using various algorithms to find desired signatures to form a linear mixing model for FCLS because the values of VD estimated by gradient and difference were generally different; a larger value was selected as the number of VEs generated in Figure 5.7.



HFC (VD = 9) NWHFC (VD = 13) SSE (VD = 10) HySime (VD = 20)

(a) HYDICE data

HFC (VD = 22) NWHFC (VD = 23) SSE (VD = 27) HySime (VD = 16)

(b) Cuprite reflectance data

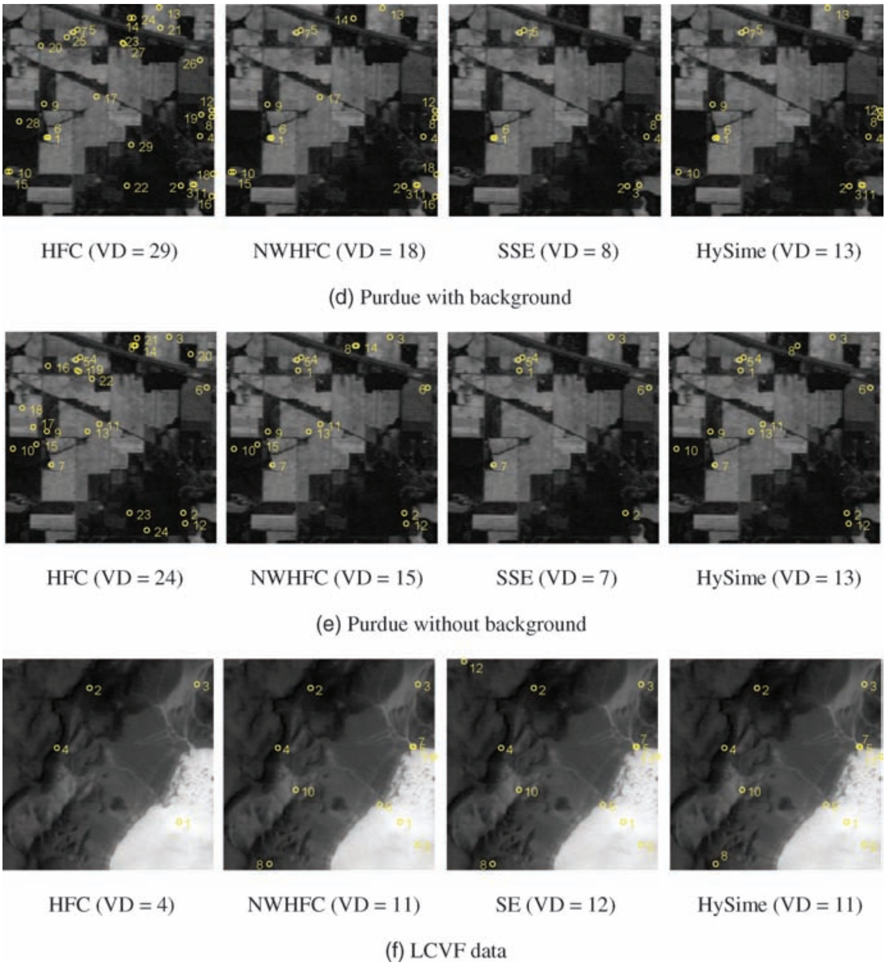
HFC (VD = 15) NWHFC (VD = 19) SSE (VD = 18) HySime (VD = 17)

(c) Cuprite radiance data

Figure 5.7 Pixels extracted by ATGP for the four image scenes, HYDICE, cuprite, Purdue, and LCVF.

Using the pixels found in Figures 5.6–5.8 to form a signature matrix for FCLS to perform spectral unmixing, Tables 5.11 and 5.12 present the LSE values between FCLS-unmixed abundance fractions and the real values of data sample vectors, where the numbers in parentheses are the numbers of signatures used by FCLS to form signature matrices for linear unmixing.

According to the results in Table 5.11, it seemed that NWHFC-ATGP and HySime-ATGP were the best in most cases. Table 5.12 shows that OSP-UFCLS showed the best performance among all OSP-based methods except the only case where OSP-ATGP produced a smaller MSE for the HYDICE (64 × 64) data than that by OSP-UFCLS. However, in this particular case, OSP-ATGP required 23 signatures to outperform OSP-UFCLS, which only needed 15 signatures. If we further



HFC (VD = 29) NWHFC (VD = 18) SSE (VD = 8) HySime (VD = 13)

(d) Purdue with background

HFC (VD = 24) NWHFC (VD = 15) SSE (VD = 7) HySime (VD = 13)

(e) Purdue without background

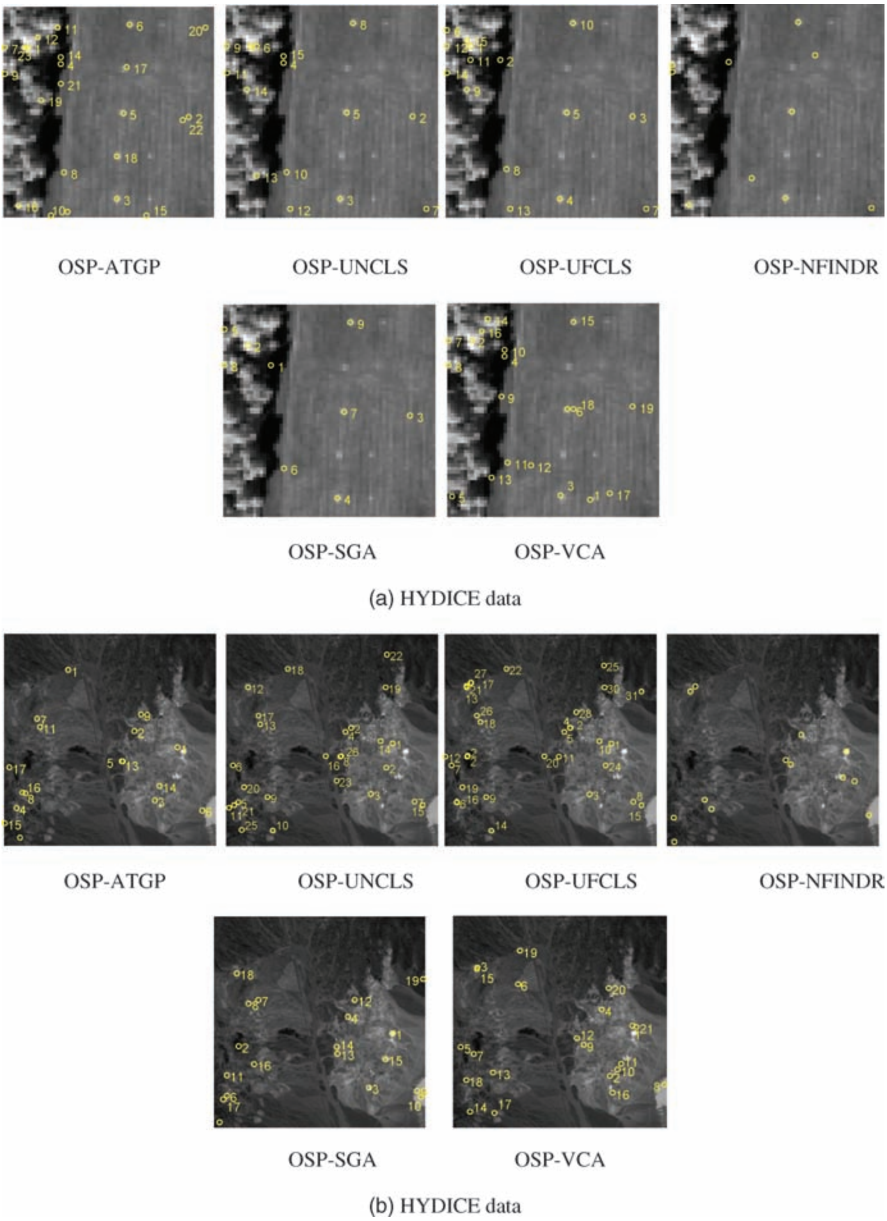
HFC (VD = 4) NWHFC (VD = 11) SE (VD = 12) HySime (VD = 11)

(f) LCVF data

Figure 5.7 (*Continued*)

compare OSP-UFCLS to NWHFC-ATGP and HySime-ATGP, the former performed better than the latter with only one case that HySime-ATGP produced a smaller MSE for the HYDICE (64 × 64) data than that by OSP-UFCLS. But once again the HySime-ATGP required 20 signatures to outperform OSP-UFCLS using 15 signatures.

As a concluding remark, the experimental results in this section for the four real image scenes suggested that the best method to estimate VD for LSMA would be the OSP-UFCLS. This conclusion makes sense because UFCLS is an unsupervised abundance fully constrained method particularly designed for LSMA.



OSP-ATGP OSP-UNCLS OSP-UFCLS OSP-NFINDR

OSP-SGA OSP-VCA

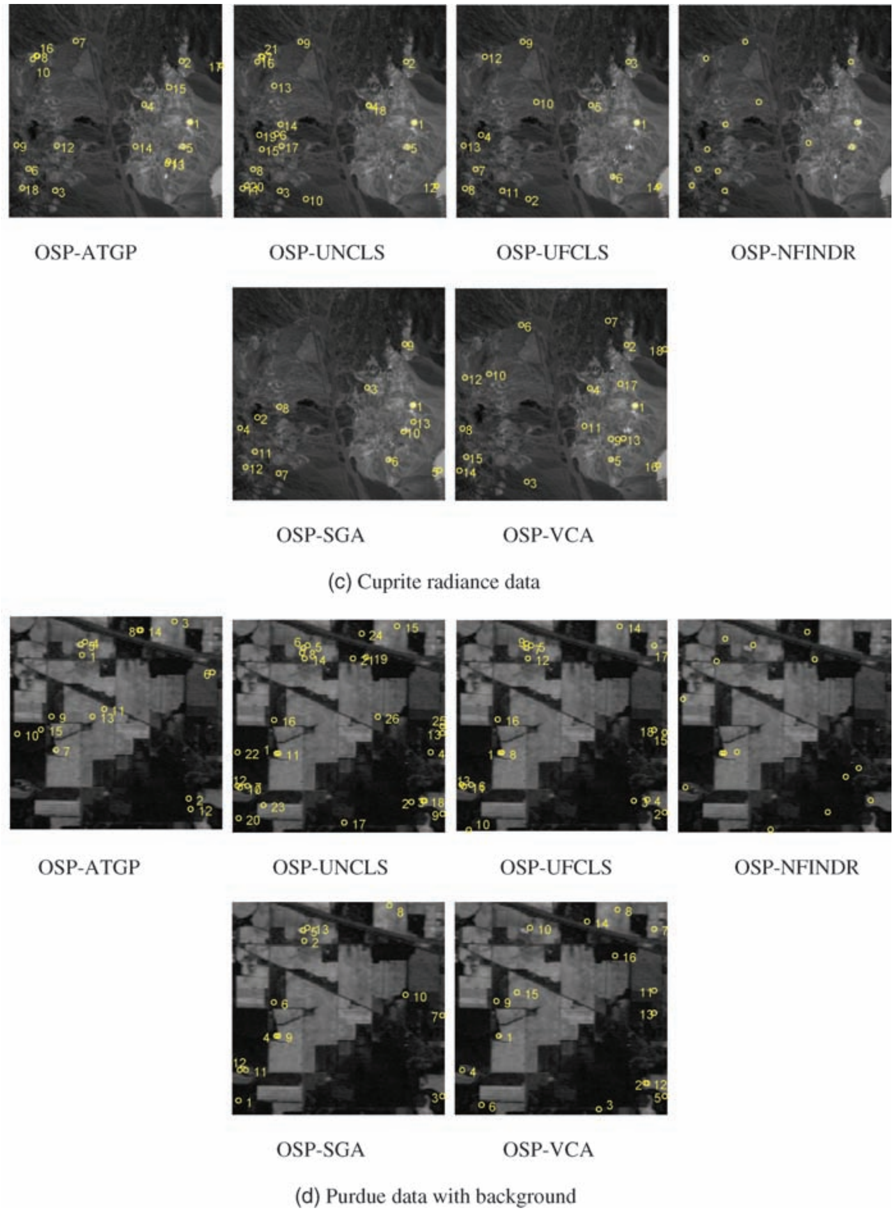
(a) HYDICE data

OSP-ATGP OSP-UNCLS OSP-UFCLS OSP-NFINDR

OSP-SGA OSP-VCA

(b) HYDICE data

Figure 5.8 VEs obtained by OSP-ATGP, OSP-UNCLS, OSP-UFCLS, OSP-N-FINDR, OSP-SGA, and OSP-VCA: (a) HYDICE data, (b) cuprite reflectance data, (c) cuprite radiance data, (d) Pudure data with background, (e) Purdue data without background, (f) LCVF data.



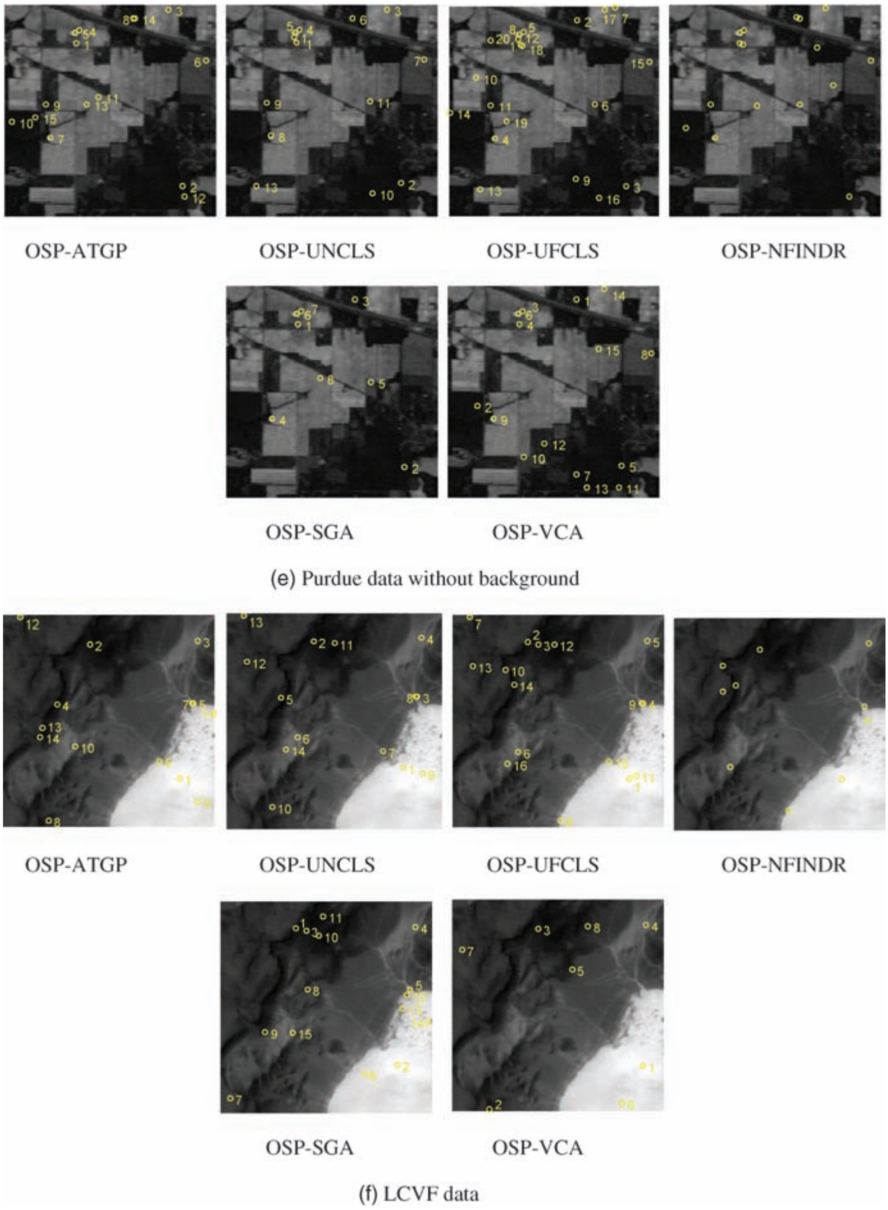
(c) Cuprite radiance data

(d) Purdue data with background

Figure 5.8 (*Continued*)

5.7 Conclusions

Estimating the number of signal sources/signatures is a crucial preprocessing step in hyperspectral data exploitation. Its accuracy has significant impact on image analysis and interpretation. Recently, the concept of VD proposed in Chang (2003a) and Chang and Du (2004) has shown success in estimating the number of spectrally distinct signatures present in hyperspectral imagery.



(e) Purdue data without background

(f) LCVF data

Figure 5.8 (*Continued*)

The exploration of VD was also investigated for many applications (Chang, 2006). This chapter revisits the concept of VD and redefines it as a general terminology to deal with signal sources of interest in various applications where the value of VD should vary with applications of interest and should not be fixed at a single constant value that fits all applications. In order to address this need, two types of criteria, data characterization-driven criteria and data representation-driven criteria, are used to develop VD estimation techniques where the former is focused on data characterization

Table 5.11 LSE of VEs estimated for the four image scenes by IN-FINDR and ATGP

	HFC-INFINDR	HFC-ATGP	NWHFC-IN-FINDR	NWHFC-ATGP	SSE-IN-FINDR	SSE-ATGP	HySime-IN-FINDR	HySime-ATGP
HYDICE	2325.1 (9)	2330.2 (9)	2148.1 (13)	1450.5 (13)	2395.2 (10)	1942.2 (10)	709.56 (20)	863.80 (20)
Cuprite reflectance	6627.1 (22)	18311 (22)	5790.8 (23)	17352 (23)	8682.5 (27)	16634 (27)	8976.1 (16)	31084 (16)
Cuprite radiance	1184.1 (15)	49254 (15)	860.53 (19)	33026 (19)	1579.3 (18)	33039 (18)	25890 (17)	33071 (17)
Purdue with BKG	1532.4 (29)	1541.4 (29)	2343.0 (18)	2601.1 (18)	11330 (8)	6498.7 (8)	5906.8 (13)	3402.4 (13)
Purdue without BKG	1480.3 (24)	929.84 (24)	1627.7 (15)	1072.7 (15)	2936.7 (7)	10205 (7)	1750.8 (13)	1125.8 (13)
LCVF	79.37 (4)	81.49 (4)	66.25 (11)	57.93 (11)	69.54 (12)	57.48 (12)	66.25 (11)	57.93 (11)

Table 5.12 LSE of VEs estimated for real image by OSP-ATGP, OSP-UNCLS, OSP-UFCLS, OSP-NFINDR, OSP-SGA, and OSP-VCA

	OSP-ATGP	OSP-UNCLS	OSP-UFCLS	OSP-NFINDR	OSP-SGA	OSP-VCA
HYDICE (64×64)	792.90 (23)	1089.9 (15)	935.34 (15)	2596.0 (10)	1793.0 (9)	992.11 (19)
Cuprite reflectance	30920 (17)	19671 (26)	2453.7 (31)	12777 (13)	6399.3 (19)	37851 (21)
Cuprite radiance	33039 (18)	638.07 (21)	1260.5 (14)	21666 (13)	2090.7 (13)	1207.6 (18)
Purdue with BKG	1784.6 (24)	870.10 (26)	1198.3 (18)	1962.0 (15)	3074.1 (13)	1651.5 (16)
Purdue without BKG	1072.7 (15)	1315.0 (13)	861.69 (20)	1150.8 (16)	3121.6 (8)	1340.9 (15)
LCVF	53.03 (14)	49.15 (14)	49.20 (16)	59.24 (10)	51.68 (15)	81.28 (8)

in terms of spectral statistics regardless of applications as opposed to the latter that finds an optimal number of signal sources to best represent data via a linear model. Tables 5.13 and 5.14 categorize these two types of criteria according to their design rationales. Category I includes factor analysis-based criteria, called real error (RE). Imbedded error (IE), extracted error (XE), and empirical indicator function (EIF) proposed by Malinowski (1997a, 1977b) and two criteria arising in passive array processing, an information theoretic criterion (AIC), minimum description length (MDL), signal subspace estimation (SSE), and hyperspectral signal subspace identification by minimum error (HySime) all produce single constant values for VD and have nothing to do with any application or algorithm. Category II comprises criteria involving one parameter that can be adapted to determine various values of VD, which include error threshold (ε)-based methods (commonly used eigenvalue-based energy methods and Gershgorin radius-based methods) and false alarm probability (P_F)-based methods. Category III is made up of OSP-based criteria that involve two parameters, an error threshold ε and an algorithm to be specified by a particular application to determine the value of VD.

Table 5.13 Data characterization-driven criteria

VD estimation	Criteria	Design rationale	VD category
Eigenvalue distribution (Eigen)	Energy percentage	Variance	II (ε)
Singular value decomposition (SVD)	Singular value percentage	Singular values	II (ε)
Malinowski's error theory	RE XE IE EIF	Factor analysis	I
Information theoretic criteria (ITC)	AIC MDL	Likelihood function	I
Gershgorin radius (GR)	Gershgorin disks Gershgorin radii	Gershgorin circle theorem	II (ε)
Neyman-Pearson (NP) detection	HFC NWHFC NSP	Binary composite hypothesis testing	II (P_F)

Tables 5.15–5.17 further summarize all criteria in Category I, Category II, and Category III, respectively, developed in this chapter based on assumptions, models, and parameters, where an asterisk (*) indicates the best technique in a specified category.

For example, SSE/HySime is considered the best criterion in Category I while the NP detection-based techniques are the best in Category II. Since there is only one technique in Category III and the performance of VD actually depends upon the application to be specified, no asterisk (*) is assigned to the best performance. Nevertheless, according to all experiments discussed in this chapter, the techniques in Category III are generally better than the techniques in Categories I and II, with some occasions where the NP detection-based techniques in Category II can do better.

Table 5.14 Data representation characterization-driven criteria

VD estimation	Criteria	Design rationale	VD category
Orthogonal subspace projection (OSP)	SVD ATGP UNCLS UFCLS N-FINDR VCA SGA	Linear model with a model error using orthogonal subspace projection (OSP)	III (ε, algorithm)
SSE	Mean squared error	Linear model with additive Gaussian noise	I
HySime	Mean squared error	Linear model with additive Gaussian noise	I

Table 5.15 VD criteria in Category I

Criteria	Noise assumption	Noise estimation	Parameters	Criterion	Model required
$\mathrm{VD}^{\mathrm{FA}}_{\mathrm{RE}}$	i.i.d.	No	No	RE	FA
$\mathrm{VD}^{\mathrm{FA}}_{\mathrm{XE}}$	i.i.d.	No	No	RE	FA
$\mathrm{VD}^{\mathrm{FA}}_{\mathrm{IE}}$	i.i.d.	No	No	IE	FA
$\mathrm{VD}^{\mathrm{FA}}_{\mathrm{EIF}}$	i.i.d.	No	No	EIF	FA
$\mathrm{VD}^{\mathrm{ITC}}_{\mathrm{AIC}}$	i.i..d. Gaussian	No	No	AIC	No
$\mathrm{VD}^{\mathrm{ITC}}_{\mathrm{MDL}}$	i.i..d. Gaussian	No	No	MDL	No
$\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{SSE}}$	Gaussian	Yes	No	LSE	LMM

Table 5.16 VD criteria in Category II

Methods for VD	Noise assumption	Noise estimation	Error threshold	Criterion	Model require
$\mathrm{VD}^{\mathrm{eigen}}_{\lambda}(\varepsilon)$, $\mathrm{VD}^{\mathrm{SVD}}_{\lambda}(\varepsilon)$	No	No	Yes (ε)	Covariance eigenvalue	No
$\mathrm{VD}^{\mathrm{eigen}}_{\hat{\lambda}}(\varepsilon)$, $\mathrm{VD}^{\mathrm{SVD}}_{\hat{\lambda}}(\varepsilon)$	No	No	Yes (ε)	Correlation eigenvalue	No
$\mathrm{VD}^{\mathrm{eigen}}_{\hat{\lambda}\text{-}\lambda}(\varepsilon)$, $\mathrm{VD}^{\mathrm{SVD}}_{\hat{\lambda}\text{-}\lambda}(\varepsilon)$	No	No	Yes (ε)	Difference eigenvalue	No
$\mathrm{VD}^{\mathrm{eigen}}_{\lambda}(a\%)$	No	No	Yes ($a\%$)	Covariance eigenvalue	No
$\mathrm{VD}^{\mathrm{GR}}_{\mathrm{disk}}(\varepsilon)$	i.i.d	No	Yes (ε)	Gershgorin disk	No
$\mathrm{VD}^{\mathrm{GR}}_{\mathrm{diff}}(\varepsilon)$	i.i.d	No	Yes (ε)	Difference in Gershgorin radius	No
$^{*}\mathrm{VD}^{\mathrm{NP}}_{\mathrm{HFC}}(P_F)$	No	No	Yes (P_F)	NP detection theory	No
$^{*}\mathrm{VD}^{\mathrm{NP}}_{\mathrm{NWHFC}}(P_F)$	No	Yes	Yes (P_F)	NP detection theory	No
$\mathrm{VD}^{\mathrm{NP}}_{\mathrm{NSP}}(P_F)$	No	Yes	Yes (P_F)	NP detection theory	No

Table 5.17 VD criteria in Category III

Methods for VD	Noise assumption	Error threshold	Specified algorithm	Criterion	Model require
$\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{ATGP}}(\varepsilon)$	No	Yes	ATGP	SNR	LMM
$\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{UFCLS}}(\varepsilon)$	No	Yes	UFCLS	SNR	LMM
$\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{VCA}}(\varepsilon)$	No	Yes	VCA	SNR	LMM
$\mathrm{VD}^{\mathrm{OSP}}_{\mathrm{SGA}}(\varepsilon)$	No	Yes	SGA	SNR	LMM

6

Data Dimensionality Reduction

Dimensionality reduction (DR) has been used in hyperspectral data exploitation for various purposes. In particular, it has been used as a preprocessing technique to reduce a very high-dimensional data space to a manageable low-dimensional space in which data analysis can be performed more effectively. Two common approaches are widely used for DR, here referred to as DR by transform (DRT) and DR by band selection (DRBS). While the former utilizes a transform to compact data in some optimal sense, the latter finds an appropriate band subset to represent data via a certain optima criterion. Two types of transforms, components analysis (CA), and feature extraction (FE), are developed for DRT. A CA transform is generally considered as a transformation that uses statistics as a criterion to de-correlate and convert data into a set of uncorrelated data components for analysis. The transforms of this type include two commonly used second-order statistics component transforms, data variance-based principal components analysis (PCA) and signal-to-noise ratio (SNR)-based maximum noise fraction (MNF) transform as well as high-order statistics-based CA transforms with criteria such as a third-order statistics-based skewness, a fourth-order statistics-based kurtosis, plus statistical independence-based independent component analysis (ICA) that requires infinite order of statistics. An FE transform uses a feature extraction-based criterion to produce a set of feature vectors so that data can be represented by these feature vectors. The transforms of this type include Fisher's ratio-based linear discriminant analysis (FLDA) and linear mixture model-based orthogonal subspace projection (OSP). All these transformation techniques can be very useful in processing of hyperspectral imagery and will be used in later chapters in this book. A second approach to DR is DRBS. Unlike DRT that produces transformed data DRBS seeks a band subset to represent the original data so that the data information of interest can be preserved in the selected band subset. Interestingly, as will be seen, most of the criteria designed for DRT are also applicable to DRBS.

6.1 Introduction

Hyperspectral images are collected by hundreds of contiguous spectral channels, and thus the data volume to be processed is considered to be huge. With such high spectral resolution, spectral correlation among bands is expected to be very high and the band-to-band spectral information may be overlapped or shared in some aspects. To address this issue, two general techniques are commonly used. One is DRT and the other is DRBS.

As for DRT, a general approach is to use component analysis (CA) transforms. It is known that one of most frequently used CA transforms is the principal components analysis (PCA) that makes

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

use of eigenvalues to determine the significance of principal components (PCs) generated by PCA so that DR is accomplished by selecting PCs in accordance with the magnitudes of their associated eigenvalues. Unfortunately, such PCA-based DR (PCA-DR) may not be effective or appropriate for hyperspectral image analysis as demonstrated in Wang and Chang (2006a). A similar approach, called maximum noise fraction (MNF) (Green et al., 1988) or noise-adjusted principal components (NAPC) transform (Lee et al., 1990), which was developed based on signal-to-noise ratio (SNR), also suffers from the same drawbacks as PCA does. One major issue for both PCA and MNF is that many subtle material substances that are uncovered by hyperspectral imaging sensors with very high spectral resolution cannot be characterized by second-order statistics. This may be due to the fact that sample pools of such substances are relatively small and their contribution to second-order statistics is very little. Consequently, these substances may not be captured by the second-order statistics-based PCA and MNF in their PCs. In order to address this issue, we must rely on high-order statistics-based CA transforms that include third-order statistics-based skewness transform (skewness-DR), fourth-order statistics-based kurtosis transform (kurtosis-DR), and statistical independence-based independent component analysis (ICA) transform (ICA-DR). Interestingly, using a criterion higher than variance or SNR to measure the significance of each transformed component to perform DR has not been explored in the past until a recent work (Wang and Chang, 2006a; Ren et al., 2006), which showed that DR can be greatly benefited by using statistics of orders higher than 2. For example, when CA transforms use criteria such as third-order statistics of skewness and fourth-order statistics of kurtosis to perform DR, they are referred to as third-order statistics-based skewness transform and fourth-order statistics-based kurtosis transform, respectively. Specifically, for the case that the mutual information is used to measure statistical independence, it becomes ICA transform. With this interpretation, the significance of a transformed component measured by a DR criterion is referred to as priority score and this particular component is then further prioritized and ranked by its associated priority score. As an example, the criterion used in PCA-DR is data variance and the priority score of a PC is calculated by the magnitude of its corresponding eigenvalue. So, all PCA-generated PCs are ranked by their associated priority scores, data variances that is, magnitudes of eigenvalues. Unfortunately, when it comes to ICA, the advantage of using component prioritization vanishes. This is because there is no specific criterion to be used to measure the significance of each of the independent components (ICs) ICA generates, and thus ICA cannot prioritize its generated ICs. For example, most algorithms designed to implement ICA, such as FastICA (Hyvarinen and Oja, 1997), make use of projection vectors randomly generated to initialize algorithms. As a result, ICs generated earlier are not necessarily more significant than those generated later. Therefore, in order for ICA to be used for DR, a criterion must be included in ICA algorithms to prioritize ICs in a similar way as PCA, MNF, skewness, and kurtosis do. Three different approaches for IC prioritization are derived in Wang and Chang (2006a). One is called statistics-prioritized ICA-DR (SPICA-DR) that is similar to those taken by PCA, MNF, skewness, and kurtosis. It utilizes criteria that can characterize statistics of any order. Another is considered as random ICA-DR (RICA-DR) that takes advantage of the nature in randomness resulting from the use of random initial projection vectors in ICA and implements an ICA algorithm in different runs to find their common intersection of ICs that will be used for DR. A third approach is referred to as initialization-driven ICA-DR (IDICA-DR) that employs a custom-designed initialization algorithm to produce an appropriate set of initial projection vectors that will be used to prioritize ICs.

The purpose of DRT is to compact data through a transformation so that the transformed data can preserve as much as data information measured by a certain criterion. An alternative way to accomplish the same goal as DRT does is DRBS. However, unlike DRT that uses a specific transformation to perform data compaction, DRBS does not perform data transformation but rather preserves original data information by selecting a subset of bands that are most likely to represent the data in some optimal sense. In other words, with very high spectral resolution hyperspectral data provide vast amount of data information that may be redundant or overlapped among adjacent bands, which can be removed without significant loss of information. DRBS takes advantage of such high inter-band correlation by selecting only representative bands to avoid band redundancy so as to achieve DR. The bands to be selected are determined by the significance of data information contained in these bands that are measured by a criterion for optimality. Despite the fact that DRT and DRBS are considered to be different approaches, the criteria used for one are indeed applicable to the other. In order to facilitate and simplify discussions in this chapter, we let q denote either the number of dimensions to be retained DRT or the number of bands to be selected by DRBS.

6.2 Dimensionality Reduction by Second-Order Statistics-Based Component Analysis Transforms

A CA transform generally transforms the image data into a set of data components so that the correlation among the transformed data components is uncorrelated according to a criterion. More specifically, a component transform represents a data space by a set of its generated data components. Two second-order component transforms have been widely used in remote sensing image processing, which are variance-based PCA transforms and SNR-based transforms and discussed as follows.

6.2.1 Eigen Component Analysis Transforms

The simplest eigen-CA transforms are those based on data variance. PCA represents this type of data variance-based CA transforms.

6.2.1.1 Principal Components Analysis

The principal components analysis (PCA), also known as Hotelling transform (Gonalez and Woods, 2002) as well as principal components transformation (PCT) (Richards and Jia, 1999; Schowengerdt, 1997), is an optimal transform to represent data in the sense of data variance. It can be considered as a discrete time version of the Karhunen–Loeve transform (KLT) in signal processing and communications (Poor, 1994) that is an optimal transform using eigenfunctions as basis functions to represent and de-correlate a function in the sense of mean-squared error. It is generally referred to as Karhunen–Loeve expansion that represents a function as a series in terms of eigenfunctions where these eigenfunctions are continuous-time functions. When they are sampled at discrete time instants, eigenfunctions become eigenvectors in a discrete case, in which case KLT is reduced to PCA. So, technically speaking, KLT used in hyperspectral data compression is indeed the principal components analysis (PCA) not what it was originally developed in statistical signal processing and communications in at least two major key aspects. The first and foremost is the used criterion. While KLT is a mean-squared error (MSE)-based transform that assumes the availability of data probability distribution to perform “mean” in terms of statistical expectation, PCA makes use of the sample covariance matrix without assuming the data probability distribution in which case PCA should be considered as a “least squares error (LSE)”-based transform, not an MSE-based transform where LSE is actually the sample variance. Secondly, KLT is generally referred to as KL expansion in statistical signal processing where a signal can be decomposed as a series of orthogonal functions, to be called eigenfunctions. For example, Fourier transform is a special case of KLT where the used sinusoidal functions are basically eigenfunctions. Therefore, in general, KLT is a continuous time transform function. By contrast, PCA is a matrix transform used to de-correlate data sample vectors into linear combinations of using eigenvectors as basis vectors for their data representations with eigenvalues as their corresponding coefficients. In light of this interpretation PCA is indeed a discrete time of KL expansion. Unfortunately, KLT has been widely abused in image processing where the image data are represented by matrices in which case KLT should be implemented as its discrete-time version, PCA. It seems that such key differences have been overlooked in hyperspectral data compression. The idea of PCA can be briefly described as follows.

Assume that $\mathrm{S} = \{\mathbf{r}_i\}_{i=1}^N$ is a set of L-dimensional image pixel vectors and $\boldsymbol{\mu}$ is the mean of the sample pool S obtained by $\boldsymbol{\mu} = (1/N)\sum_{i=1}^N \mathbf{r}_i$. Let $\mathbf{X}$ be the sample data matrix formed by $\mathbf{X} = [\mathbf{r}_1\mathbf{r}_2 \cdots \mathbf{r}_N]$. Then the sample covariance matrix of the S is obtained by $\mathbf{K} = (1/N)\left[\mathbf{X}\mathbf{X}^T\right] = (1/N)\left[\sum_{i=1}^N (\mathbf{r}_i - \boldsymbol{\mu})(\mathbf{r}_i - \boldsymbol{\mu})^T\right]$. If we further assume that $\{\lambda_l\}_{l=1}^L$ is the set of eigenvalues obtained from the covariance matrix $\mathbf{K}$ and $\{\mathbf{v}_l\}_{l=1}^L$ are their corresponding unit eigenvectors, that is, $||\mathbf{v}_l|| = 1$, we can define a diagonal matrix $\mathbf{D}_\sigma$ with variances $\left\{\sigma_l^2\right\}_{l=1}^L$ along the diagonal line as

$$\mathbf{D}_\sigma = \begin{bmatrix} \sigma_1^2 & 0 & \mathbf{0} \\ 0 & \ddots & 0 \\ \mathbf{0} & 0 & \sigma_L^2 \end{bmatrix} \tag{6.1}$$

and an eigenvector matrix $\boldsymbol{\Lambda}$ specified by $\{\mathbf{v}_l\}_{l=1}^L$ as

$$\boldsymbol{\Lambda} = [\mathbf{v}_1\mathbf{v}_2 \cdots \mathbf{v}_L] \tag{6.2}$$

such that

$$\mathbf{D}_\sigma = \boldsymbol{\Lambda}^T\mathbf{K}\boldsymbol{\Lambda} \tag{6.3}$$

Using the eigenvector matrix $\boldsymbol{\Lambda}$ a linear transform $\xi_{\boldsymbol{\Lambda}}$ defined by

$$\xi_{\boldsymbol{\Lambda}}(\mathbf{r}) = \boldsymbol{\Lambda}^T\mathbf{r} \tag{6.4}$$

transforms every data sample $\mathbf{r}_i$ to a new data sample, $\tilde{\mathbf{r}}_i$ by

$$\tilde{\mathbf{r}}_i \equiv \boldsymbol{\Lambda}^T\mathbf{r}_i \tag{6.5}$$

As a result, the mean of new $\xi_{\boldsymbol{\Lambda}}$-transferred data samples $\{\tilde{\mathbf{r}}_i\}_{i=1}^N$ becomes $\tilde{\boldsymbol{\mu}} = (1/N)\sum_{i=1}^N \tilde{\mathbf{r}}_i$ and its resulting covariance matrix is reduced to a diagonal matrix given by

$$\tilde{\mathbf{K}} = (1/N)\sum_{i=1}^N (\tilde{\mathbf{r}}_i - \tilde{\boldsymbol{\mu}})(\tilde{\mathbf{r}}_i - \tilde{\boldsymbol{\mu}})^T = \boldsymbol{\Lambda}^T\mathbf{K}\boldsymbol{\Lambda} = \mathbf{D}_\sigma \tag{6.6}$$

Equation (6.6) implies that the $\xi_{\boldsymbol{\Lambda}}$-transferred data matrix $\tilde{\mathbf{X}} = [\tilde{\mathbf{r}}_1\tilde{\mathbf{r}}_2 \cdots \tilde{\mathbf{r}}_N]$ has been de-correlated or whitened by the matrix $\boldsymbol{\Lambda}$ that is referred to as a whitening matrix (Poor, 1994). The transform $\xi_{\boldsymbol{\Lambda}}$ defined by (6.5) is generally called principal component transform and the lth component of $\hat{\mathbf{X}}$ is formed by

$$\xi_{\mathbf{v}_l}(\mathbf{X}) = \mathbf{v}_l^T\mathbf{X} \tag{6.7}$$

and is called the lth principal component (PC) that consists of $\left\{\mathbf{v}_l^T\mathbf{r}_i\right\}_{i=1}^N$ that are $\xi_{\mathbf{v}_l}$-transferred data samples corresponding the lth eigenvalue λ_l. PCA is a process that implements the transform

$\xi_{\boldsymbol{\Lambda}}$ defined by (6.4) to obtain a set of principal components (PCs) via (6.5) or (6.7) with all $1 \leq l \leq L$. In order to achieve DR, only the PCs specified by eigenvectors that correspond to first q largest eigenvalues will be retained, while the PCs specified by eigenvectors corresponding to the remaining (L–q) smaller eigenvalues will be discarded. The same process can be accomplished by the singular value decomposition (SVD) to be described in Section 6.2.1.6.

6.2.1.2 Standardized Principal Components Analysis

In PCA, its focus is placed on the variances of the image pixel vectors $\{\mathbf{r}_i\}_{i=1}^N$. It has been shown by Singh and Harison (1985) that in some applications in remote sensing, it may be more effective to deal with data co-variances rather than data variances. Such co-variance-based PCA is called standardized principal components analysis (SPCA).

Assume that the covariance matrix $\mathbf{K}$ is given by

$$\mathbf{K} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \sigma_{13} & \cdots & \sigma_{1L} \\ \sigma_{21} & \sigma_2^2 & \sigma_{23} & \cdots & \sigma_{2L} \\ \rho_{31} & \sigma_{32} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \sigma_{L-1}^2 & \sigma_{(L-1)L} \\ \sigma_{L1} & \sigma_{L2} & \cdots & \sigma_{L(L-1)} & \sigma_L^2 \end{bmatrix} \tag{6.8}$$

with the lth variance and (i,j)-covariance denoted by σ_l^2 and σ_{ij}, respectively. Now we define a standard deviation matrix of $\mathbf{K}$ via (6.78) as $\mathbf{D}_\sigma^{1/2}$ that is the diagonal matrix of the form

$$\mathbf{D}_\sigma^{1/2} = \begin{bmatrix} \sigma_1 & 0 & \mathbf{0} \\ 0 & \ddots & 0 \\ \mathbf{0} & 0 & \sigma_L \end{bmatrix}. \tag{6.9}$$

Then $\bar{\mathbf{r}}_i = \mathbf{D}_\sigma^{-1/2}\mathbf{r}_i$ is called a standardized sample of $\mathbf{r}_i$ and $\mathbf{K}$ can be expressed as

$$\mathbf{K} = \mathbf{D}_\sigma^{1/2}\mathbf{R}_\mathbf{K}\mathbf{D}_\sigma^{1/2} \text{ or } \mathbf{R}_\mathbf{K} = \mathbf{D}_\sigma^{-1/2}\mathbf{K}\mathbf{D}_\sigma^{-1/2} \tag{6.10}$$

where $\mathbf{R}_\mathbf{K}$ is called the correlation coefficient matrix defined by

$$\mathbf{R}_\mathbf{K} = \begin{bmatrix} 1 & \kappa_{12} & \kappa_{13} & \cdots & \kappa_{1L} \\ \kappa_{21} & 1 & \kappa_{23} & \cdots & \kappa_{2L} \\ \kappa_{31} & \kappa_{32} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 & \kappa_{(L-1)L} \\ \kappa_{L1} & \kappa_{L2} & \cdots & \kappa_{L(L-1)} & 1 \end{bmatrix} \tag{6.11}$$

with $\kappa_{ij} = \sigma_{ij}/\sqrt{\sigma_i\sigma_j}$ and $i \neq j$. It should be noted that the $\mathbf{R}_\mathbf{K}$ in (6.11) is not the sample correlation matrix $\mathbf{R}$. The κ_{ij} in $\mathbf{R}_\mathbf{K}$ is generally called the (i,j)th correlation coefficient of $\mathbf{K}$.

Using (6.2) $\boldsymbol{\Lambda} = [\mathbf{v}_1\mathbf{v}_2 \cdots \mathbf{v}_L]$ is the eigenvector matrix of $\mathbf{K}$ formed by its unit eigenvectors $\{\mathbf{v}_l\}_{l=1}^L$. Through (6.10) we can obtain

$$\mathbf{I} = \boldsymbol{\Lambda}^T\mathbf{R}_\mathbf{K}\boldsymbol{\Lambda} = \boldsymbol{\Lambda}^T\mathbf{D}_\sigma^{-1/2}\mathbf{K}\mathbf{D}_\sigma^{-1/2}\boldsymbol{\Lambda} \tag{6.12}$$

that is the $L \times L$ identity matrix. Combining the eigenvector matrix $\mathbf{\Lambda}$ in (6.2) and the diagonal matrix $\mathbf{D}_\sigma^{-1}$ obtained by (6.9) we can define a linear transform $\xi_{\mathbf{D}_\sigma^{-1}\mathbf{\Lambda}}$ by

$$\xi_{\mathbf{D}_\sigma^{-1}\mathbf{\Lambda}}(\mathbf{r}_i) = \left(\mathbf{D}_\sigma^{-1/2}\mathbf{\Lambda}\right)^T \mathbf{r}_i \tag{6.13}$$

that is called standardized PCA (SPCA) and denoted by

$$\mathbf{r}_i^{\text{SPCA}} \equiv \xi_{\mathbf{D}_\sigma^{-1}\mathbf{\Lambda}}(\mathbf{r}_i) = \left(\mathbf{D}_\sigma^{-1/2}\mathbf{\Lambda}\right)^T \mathbf{r}_i \tag{6.14}$$

Using (6.12) and (6.14), the covariance matrix of the new data samples $\left\{\mathbf{r}_i^{\text{SPCA}}\right\}_{i=1}^N$ that are obtained from $\{\mathbf{r}_i\}_{i=1}^N$ by the SPCA in (6.14) becomes an identity matrix.

Similarly, in analogy with the decomposition of $\mathbf{K}$, its inverse matrix $\mathbf{K}^{-1}$ can be also decomposed as

$$\mathbf{K}^{-1} = \mathbf{D}_\varsigma^{1/2}\mathbf{R}_{\mathbf{K}^{-1}}\mathbf{D}_\varsigma^{1/2} \tag{6.15}$$

where

$$\mathbf{D}_\varsigma^{1/2} = \begin{bmatrix} \zeta_1 & 0 & \mathbf{0} \\ 0 & \ddots & 0 \\ \mathbf{0} & 0 & \zeta_L \end{bmatrix} \tag{6.16}$$

and $\left\{\zeta_l^2\right\}_{l=1}^L$ are variances of $\mathbf{K}^{-1}$ and

$$\mathbf{R}_{\mathbf{K}^{-1}} = \begin{bmatrix} 1 & \eta_{12} & \eta_{13} & \cdots & \eta_{1L} \\ \eta_{21} & 1 & \eta_{23} & \cdots & \eta_{2L} \\ \eta_{31} & \eta_{32} & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 & \eta_{(L-1)L} \\ \eta_{L1} & \eta_{L2} & \cdots & \eta_{L(L-1)} & 1 \end{bmatrix} \tag{6.17}$$

with η_{ij} being the (i,j)th correlation coefficient of $\mathbf{K}^{-1}$ and $i \neq j$. It turns out that the ς_l in (6.16) can be related to the σ_l in (6.9) by the following formula:

$$\zeta_l = \sigma_l^{-1}\left(1 - \upsilon_l^2\right)^{-1/2} \tag{6.18}$$

where υ_l^2 is a multiple correlation coefficient of the data in the lth dimension on all other $L - 1$ dimensions obtained by using the multiple regression theory. So, υ_l^2 is the reciprocal of a good noise variance estimate for the lth-dimensional data space. It should be noted that the $\mathbf{D}_\zeta$ in (6.16) is not an inverse of the $\mathbf{D}_\sigma$ in (6.9), nor is $\mathbf{R}_{\mathbf{K}^{-1}}$ in (6.17). The major advantage of using ζ_l over σ_l is that as shown in (6.18), ζ_l removes its correlation on other ζ_l's for $l \neq k$, while σ_l does not. Like PCA, SPCA achieves DR by only retaining standard PCs corresponding to eigenvectors that are associated with first q largest eigenvalues.

6.2.1.3 Singular Value Decomposition

Another eigen-CA transform is the singular value decomposition (SVD). Unlike PCA that is primarily designed to de-correlate the covariance matrix, SVD is one of most widely used techniques in systems, communications, and signal processing to resolve issues caused by ill-conditioned systems, such as underdetermined or overdetermined least squares system. It provides a matrix factorization of an arbitrary matrix into a product of two unitary matrices and a diagonal matrix. More specifically, let $\mathbf{A}_{m\times n}$ be an $m \times n$ real matrix. Define two matrices $\mathbf{B}_{m\times m} = \mathbf{A}_{m\times n}(\mathbf{A}_{m\times n})^T$ and $\mathbf{C}_{n\times n} = \mathbf{A}_{m\times n}^T\mathbf{A}_{m\times n}$ that can, respectively, be referred to as outer product matrix and inner product matrix where both matrices $\mathbf{B}_{m\times m}$ and $\mathbf{C}_{n\times n}$ are symmetric, semidefinite with non-negative real eigenvalues, and have the same rank. In particular, when $\mathbf{A}_{m\times n}$ is an m-dimensional column vector $\mathbf{x}$, the outer product matrix is an $m \times m$ matrix, $\mathbf{x}\mathbf{x}^T$, and the inner product matrix $\mathbf{x}^T\mathbf{x}$ is a scalar, both of which have rank 1. In this case, the only nonzero eigenvalue of the outer product matrix of $\mathbf{x}\mathbf{x}^T$is specified by its inner product matrix $\mathbf{x}^T\mathbf{x}$.

Assume that the eigenvalues of $\mathbf{B}_{m\times m}$ and $\mathbf{C}_{n\times n}$ are $\{\lambda_i\}_{i=1}^n$ and $\{\bar{\lambda}_i\}_{i=1}^n$ that can be arranged in descending order in magnitude as follows:

$$\begin{array}{c} \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_q > 0 = \lambda_{q+1} \geq \cdots \geq \lambda_r \geq \cdots \geq \lambda_m \\ \bar{\lambda}_1 \geq \bar{\lambda}_2 \geq \cdots \geq \bar{\lambda}_{\bar{q}} > 0 = \bar{\lambda}_{\bar{q}+1} \geq \cdots \geq \bar{\lambda}_r \geq \cdots \geq \bar{\lambda}_n \end{array} \tag{6.19}$$

where $r = \min\{m, n\}$. Since both matrices $\mathbf{B}_{n\times n}$ and $\mathbf{C}_{m\times m}$ have the same rank and also identical nonzero eigenvalues, $q = \bar{q}$ and $\lambda_j = \bar{\lambda}_j$ for all $1 \leq j \leq q$. Then the set of square root of eigenvalues $\{\lambda_i\}_{i=1}^r$ in (6.19)

$$\sqrt{\lambda_1} \geq \sqrt{\lambda_2} \geq \cdots \geq \sqrt{\lambda_q} > 0 = \sqrt{\lambda_{q+1}} \geq \sqrt{\lambda_{q+2}} \geq \cdots \geq \sqrt{\lambda_r} \tag{6.20}$$

is called the singular values of the matrix $\mathbf{A}_{m\times n}$ (Chen, 1999). Now we can further decompose the matrix $\mathbf{A}_{m\times n}$ into the following factorization form:

$$\mathbf{A}_{m\times n} = \mathbf{F}_{m\times m}\mathbf{D}_{m\times n}\mathbf{G}_{n\times n}^T \tag{6.21}$$

where $\mathbf{F}_{m\times m}$ is an $m \times m$ unitary matrix with its column vectors being orthonormalized eigenvectors of the $m \times m$ matrix $\mathbf{B}_{m\times m} = \mathbf{A}_{m\times n}(\mathbf{A}_{m\times n})^T$ so that $\mathbf{F}_{m\times m}^{-1} = \mathbf{F}_{m\times m}^T$, $\mathbf{G}_{n\times n}$ is an $n \times n$ unitary matrix with its column vectors being orthonormalized eigenvectors of the $n \times n$ matrix $\mathbf{C}_{n\times n} = \mathbf{A}_{m\times n}^T\mathbf{A}_{m\times n}$ so that $\mathbf{G}_{n\times n}^{-1} = \mathbf{G}_{n\times n}^T$, and $\mathbf{D}_{m\times n}$ is an $m \times n$ diagonal matrix $\mathbf{D}_{m\times n} = \mathrm{diag}\{\sqrt{\lambda_1}, \sqrt{\lambda_2}, \ldots, \sqrt{\lambda_r}\}$ with its diagonal entries specified by the singular values of $\mathbf{A}_{m\times n}$ and arranged in descending order in magnitude, $\sqrt{\lambda_1} \geq \sqrt{\lambda_2} \geq \cdots \geq \sqrt{\lambda_r}$. Specifically, if the rank of $\mathbf{A}_{m\times n}$ is r, then $\mathbf{D}_{m\times n}$ is a square matrix of size $r \times r$ with $q = r$.

In hyperspectral data exploitation, assume that $\{\mathbf{r}_i\}_{i=1}^N$ is a set of entire image pixel vectors or a set of training samples in a hyperspectral image. $\mathbf{A}_{m\times n}$ in (6.21) can be considered by either a data matrix formed by data samples/image pixel vectors with the subscript m and n denoting the total number of spectral bands and the number of image pixel vectors (such as total number of image pixel vectors or training samples), respectively, or a sample correlation/covariance matrix $\mathbf{K}_{L\times L}/\mathbf{R}_{L\times L}$ formed by the total number of data samples/image pixel vectors with $m = n = L$ being the total number of spectral bands, and q being the number of dimensions to be retained, respectively. In the former case, the matrix $\mathbf{A}_{m\times n}$ in (6.21) is formed by data sample vectors, $\{\mathbf{r}_i\}_{i=1}^N$ with the ith column vector specified by the ith image pixel vector $\mathbf{r}_i$. So, the resulting matrix is represented by $\mathbf{A}_{L\times N} = [\mathbf{r}_1\mathbf{r}_2\cdots\mathbf{r}_N]$ with $q \leq r \leq \min\{L, N\}$. In (6.21) $\mathbf{F}_{L\times L}$ and $\mathbf{G}_{N\times N}$ are called left and right singular vector matrices of $\mathbf{A}_{L\times N}$ and they are generally different. The singular values of $\mathbf{A}_{L\times N}$ are simply square root of non-negative eigenvalues of $N\mathbf{R}_{L\times L}$, $\sqrt{\lambda_1} \geq \sqrt{\lambda_2} \geq \cdots \geq \sqrt{\lambda_q}$. In other words, if we interpret eigenvalues as variances, the singular values are simply their standard deviations. As for the latter case, the matrix $\mathbf{A}_{m\times n}$ in (6.21) is formed by the data sample correlation matrix, $\mathbf{R}_{L\times L} = \frac{1}{N}\mathbf{A}_{L\times N}(\mathbf{A}_{L\times N})^T$, that becomes the outer product matrix of $\mathbf{A}_{L\times N}$, $\mathbf{A}_{L\times N}(\mathbf{A}_{L\times N})^T$ scaled by a constant $(1/N)$. The singular values of $\mathbf{A}_{L\times N}(\mathbf{A}_{L\times N})^T$ are exactly the same non-negative eigenvalues of $\mathbf{R}_{L\times L}$, and the left and right singular vector matrices of $\mathbf{A}_{L\times N}(\mathbf{A}_{L\times N})^T$, $\mathbf{F}_{L\times L}$, and $\mathbf{G}_{L\times L}$ turn out to be the same as the eigenvector matrix $\mathbf{\Lambda}$ described by (6.2); (6.21) is reduced to (6.3), in which case SVD becomes PCA described in Section 6.2.1.1.

In order to further explore insights into the relationship between the SVD and PCA, let $\{\lambda_j\}_{j=1}^L$ and $\left\{\hat{\lambda}_j\right\}_{j=1}^L$ be eigenvalues of the sample covariance matrix $\mathbf{K}_{L\times L}$ and the sample correlation matrix $\mathbf{R}_{L\times L}$ with their corresponding eigenvectors $\{\mathbf{v}_j\}_{j=1}^L$ and $\{\hat{\mathbf{v}}_j\}_{j=1}^L$, respectively. Also let $\{s_{\text{data},j}\}_{j=1}^L$, $\{s_j\}_{j=1}^L$ and $\{\hat{s}_j\}_{j=1}^L$ be the singular values of $\mathbf{A}_{L\times N} = [\mathbf{r}_1\mathbf{r}_2\ldots\mathbf{r}_N]$, $\mathbf{K}_{L\times L}$ and $\mathbf{R}_{L\times L}$ with their corresponding singular vectors $\{\mathbf{u}_{\text{data},j}\}_{j=1}^L$, $\{\mathbf{u}_j\}_{j=1}^L$ and $\{\hat{\mathbf{u}}_j\}_{j=1}^L$, respectively. The following relationships can be derived and summarized as follows.

1. $s_j = \lambda_j$
2. $\frac{s_{\text{data},j}}{\sqrt{N}} = \sqrt{\hat{s}_j} = \sqrt{\hat{\lambda}_j}$ where $(1/\sqrt{N})s_{\text{data},j}$ is the squared root of the eigenvalue λ_j resulting from the fact that the sample correlation/covariance matrix **R/K** is the outer product of the data matrix with/out mean removed.
3. $\mathbf{u}_j = \pm\mathbf{v}_j$ that implies that for each j $\mathbf{u}_j$ and $\mathbf{v}_j$ are either identical or differ by a sign. In the latter case, $\mathbf{u}_j$ and $\mathbf{v}_j$ point to complete opposite directions. However, it is the sign difference that distinguishes the SVD from PCA and makes both PCA and SVD two different transformations that also yield different performances. In order for SVD to avoid such a sign issue of which one, $\mathbf{u}_j$ or $\mathbf{v}_j = -\mathbf{u}_j$ should be selected as the desired singular vector; we can map all data sample vectors on the singular vector $\mathbf{u}_j$ and sum all their projections by calculating their inner products via $\langle\mathbf{r}_i, \mathbf{u}_j\rangle = \mathbf{r}_i^T\mathbf{u}_j$. If the total sum of the projections is non-negative, that is, $\sum_{i=1}^N \mathbf{r}_i^T\mathbf{u}_j \geq 0$ is non-negative, the desired singular vector is set to $\mathbf{u}_j$ and $\mathbf{v}_j = -\mathbf{u}_j$otherwise, that is, $\mathbf{v}_j = -\mathbf{u}_j$ if $\sum_{i=1}^N \mathbf{r}_i^T\mathbf{u}_j < 0$.
4. $\mathbf{u}_{\text{data},j} = \pm\hat{\mathbf{u}}_j$ and $\hat{\mathbf{u}}_j \neq \mathbf{u}_j$ for $1 \leq j \leq L$.
5. $\hat{\mathbf{u}}_j = \pm\hat{\mathbf{v}}_j$, which implies that for each j, $\hat{\mathbf{u}}_j$ and $\hat{\mathbf{v}}_j$ are either identical or differ by a sign. In the latter case, $\hat{\mathbf{u}}_j$ and $\hat{\mathbf{v}}_j$ point to complete opposite directions.

Finally, as an alternative, we can also find the singular values of the inner product matrix of the matrix $\mathbf{A}_{L\times N}$, $(\mathbf{A}_{L\times N})^T\mathbf{A}_{L\times N}$ with size of $N \times N$. It turns out that both inner product matrix of $\mathbf{A}_{L\times N}$, $(\mathbf{A}_{L\times N})^T\mathbf{A}_{L\times N}$ and the outer product matrix of $\mathbf{A}_{L\times N}$, $\mathbf{A}_{L\times N}(\mathbf{A}_{L\times N})^T$ have the same identical non zero singular values with only difference in the number of zero singular values. This implies that to perform DR for any matrix $\mathbf{A}_{m\times n}$, either inner product matrix $(\mathbf{A}_{L\times N})^T\mathbf{A}_{L\times N}$ or outer product matrix $\mathbf{A}_{L\times N}(\mathbf{A}_{L\times N})^T$ can be used for SVD. Apparently, in hyperspectral imaging the data sample correlation matrix $\mathbf{R}_{L\times L} = \frac{1}{N}\mathbf{A}_{L\times N}(\mathbf{A}_{L\times N})^T$that is a $(1/N)$-scaled outer product matrix of a data matrix $\mathbf{A}_{L\times N}$ is the most intuitive and logical way to be chosen for DR.

There are also other factorization forms similar to (6.6) that can be used in place of SVD, for example, Cholesky decomposition, QR decomposition, and Householder transformation (Golub and Van Loan, 1989), that can be used for real-time implementation (see Chapter 33 and Chang (2013)).

6.2.2 Signal-to-Noise Ratio-Based Components Analysis Transforms

The PCA discussed in Section 6.2.1 is developed to arrange PCs in descending order of data variance. However, data variance does not mean image quality. In other words, PCA-ordered PCs are not necessarily ordered by image quality as shown by Green *et al.* (1988). In order to address this issue, Green *et al.* (1998) used an approach similar to PCA, called maximum noise fraction (MNF), that was based on a different criterion, signal-to-noise (SNR), to measure image quality. It was later shown by Lee *et al.* (1990) that MNF actually performed two stage processes, noise whitening with unit variance followed by PCA. Because of that MNF was also referred to as noise-adjusted principal component (NAPC) transform.

6.2.2.1 Maximum Noise Fraction Transform

The idea of MNF can be briefly described as follows. Assume that $\{\mathbf{r}_i\}_{i=1}^N$ is a set of entire image pixel vectors in a hyperspectral image with size $N = n_r n_c$ when n_r and n_c denote the number of rows and columns in the image, respectively. Let each image pixel vector also be denoted by an L-dimensional column vector $\mathbf{r}_i = (r_{i1}, r_{i2}, \ldots, r_{iL})^T$. Suppose that the lth band image can be represented by an N-dimensional column vector, $\mathbf{b}_l = (r_{l1}, r_{l2}, \ldots, r_{lN})^T$. It assumes that an observation model

$$\mathbf{b}_l = \mathbf{s}_l + \mathbf{n}_l \tag{6.22}$$

where $\mathbf{b}_l$ is an observation vector, $\mathbf{s}_l$ is an N-dimensional signal column vector, and $\mathbf{n}_l$ is an N-dimensional column vector uncorrelated with $\mathbf{s}_l$.

Let $\sigma_{n_l}^2$ and $\sigma_{s_l}^2$ denote the noise variance and signal variance of $\mathbf{b}_l$, respectively. We define the noise fraction (NF) of the lth band image vector $\mathbf{b}_l$ to be the ratio of the noise variance, $\sigma_{\mathbf{n}_l}^2$ in the lth band image to the total variance, $\sigma_{\mathbf{b}_l}^2$ in the lth band image given by

$$\mathrm{NF}_l = \sigma_{\mathbf{n}_l}^2 / \sigma_{\mathbf{b}_l}^2 \tag{6.23}$$

where $\sigma_{\mathbf{b}_l}^2 = (1/N)\sum_{i=1}^N (r_{il} - \mu_l)^2$ and $\mu_l = (1/N)\sum_{i=1}^N r_{il}$.

Assume that $\mathbf{w}_l$ is an L-dimensional column vector that will be used to linearly transform the lth band image vector $\mathbf{b}_l = (r_{l1}, r_{l2}, \ldots, r_{lN})^T$ to a new lth band image described by $\tilde{\mathbf{b}}_l = (\tilde{r}_{l1}, \tilde{r}_{l2}, \ldots, \tilde{r}_{lN})^T$ via

$$\tilde{r}_{li} = \mathbf{w}_l^T \mathbf{r}_i = \sum\nolimits_{k=1}^L w_{lk} r_{ik} \tag{6.24}$$

It is worth noting that the ith component of the lth band image vector $\tilde{\mathbf{b}}_l$, $\tilde{r}_{li}$ in (6.24) is obtained by weighted sum over image pixels in all the L bands of the ith image pixel vector $\mathbf{r}_i$. So, MNF is to find a transform specified by $\mathbf{w}_l^{\mathrm{MNF}}$to maximize the NF defined by

$$\max_{\mathbf{w}_l}\left\{\sigma_{\tilde{\mathbf{n}}_l}^2 / \sigma_{\tilde{\mathbf{b}}_l}^2\right\} = \max_{\mathbf{w}_l}\left\{\frac{\mathbf{w}_l^T\left[\sigma_{\mathbf{n}_l}^2\right]\mathbf{w}_l}{\mathbf{w}_l^T\left[\sigma_{\mathbf{b}_l}^2\right]\mathbf{w}_l}\right\} = \frac{\left(\mathbf{w}_l^{\mathrm{MNF}}\right)^T\left[\sigma_{\mathbf{n}_l}^2\right]\mathbf{w}_l^{\mathrm{MNF}}}{\left(\mathbf{w}_l^{\mathrm{MNF}}\right)^T\left[\sigma_{\mathbf{b}_l}^2\right]\mathbf{w}_l^{\mathrm{MNF}}} \tag{6.25}$$

Let $\mathbf{W}^{\mathrm{MNF}} = \left[\mathbf{w}_1^{\mathrm{MNF}}\mathbf{w}_2^{\mathrm{MNF}} \ldots \mathbf{w}_L^{\mathrm{MNF}}\right]$ be an MNF transform matrix such that $\tilde{\mathbf{X}} = \left(\mathbf{W}^{\mathrm{MNF}}\right)^T\mathbf{X}$ where $\mathbf{X} = [\mathbf{r}_1\mathbf{r}_2 \cdots \mathbf{r}_N]$ and $\tilde{\mathbf{X}} = [\tilde{\mathbf{r}}_1\tilde{\mathbf{r}}_2 \cdots \tilde{\mathbf{r}}_N]$. Then we can obtain the lth MNF-transformed band image by $\tilde{\mathbf{b}}_l = (\tilde{r}_{l1}, \tilde{r}_{l2}, \ldots, \tilde{r}_{lN})^T$ via $\tilde{r}_{li} = \left(\mathbf{w}_l^{\mathrm{MNF}}\right)^T \mathbf{r}_i = \sum_{k=1}^{L} w_{lk}^{\mathrm{MNF}} r_{ik}$ specified by (6.24). Since the criterion of NF given by (6.23) can be re-expressed as

$$\mathrm{NF}_l = \frac{\sigma^2_{\mathbf{n}_l}}{\sigma^2_{\mathbf{b}_l}} = \frac{\sigma^2_{\mathbf{n}_l}}{\sigma^2_{\mathbf{s}_l} + \sigma^2_{\mathbf{n}_l}} = \frac{1}{\left(\sigma^2_{\mathbf{s}_l}/\sigma^2_{\mathbf{n}_l}\right) + 1} = \frac{1}{\mathrm{SNR}_l + 1} \tag{6.26}$$

where $\mathrm{SNR}_l = \sigma^2_{\mathbf{s}_l}/\sigma^2_{\mathbf{n}_l}$ is signal-to-noise ratio defined in (6.25). As a consequence, maximizing the NF_l specified by (6.23) is equivalent to minimizing SNR_l specified by (6.26). The Green *et al.* developed MNF is to find a set of $\left\{\mathbf{w}_l^{\mathrm{MNF}}\right\}_{l=1}^{L}$ to maximize the noise fraction in each of bands and then arrange MNF-transformed bands in descending order of maximum noise fractions according to (6.23) or in ascending order of SNR according to (6.26).

6.2.2.2 Noise-Adjusted Principal Component Transform

Recently, Lee *et al.* (1990) re-interpreted MNF transform and showed that MNF transform was nothing more than a two-stage process that first whitened noise variances of each band image to unit variance, then performed PCA transform on the noise-whitened band images. As a result, PCA-generated principal components can be arranged in the descending order of SNR that is the reverse order arranged by MNF transform. With this new interpretation, MNF is further referred to as noise-adjusted principal component (NAPC) transform. In other words, we can reinterpret MNF transform that maximizes the NF_l in (6.23) to minimize its reciprocal defined by

$$\min_{\mathbf{w}_l}\left\{\sigma^2_{\tilde{\mathbf{b}}_l}/\sigma^2_{\tilde{\mathbf{n}}_l}\right\} = \min_{\mathbf{w}_l}\left\{\frac{\mathbf{w}_l^T\left[\sigma^2_{\mathbf{b}_l}\right]\mathbf{w}_l}{\mathbf{w}_l^T\left[\sigma^2_{\mathbf{n}_l}\right]\mathbf{w}_l}\right\} \tag{6.27}$$

or maximize the SNR over the reciprocal of (6.26) defined by

$$\begin{aligned} \max_{\mathbf{w}_l}\{\mathrm{SNR}_l\} &= \max_{\mathbf{w}_l}\left\{\frac{\sigma^2_{\mathbf{b}_l}}{\sigma^2_{\mathbf{n}_l}}\right\} = \max_{\mathbf{w}_l}\left\{\frac{\sigma^2_{\mathbf{s}_l} + \sigma^2_{\mathbf{n}_l}}{\sigma^2_{\mathbf{n}_l}}\right\} \\ &= \max_{\mathbf{w}_l}\left\{\left(\sigma^2_{\mathbf{s}_l}/\sigma^2_{\mathbf{n}_l}\right) + 1\right\} = \max_{\mathbf{w}_l}\{\mathrm{SNR}_l + 1\} \end{aligned} \tag{6.28}$$

As a result of (6.27) or (6.28), the obtained transform vectors arrange band images in ascending order of noise fractions or descending order of SNR. Interestingly, MNF used in the popular ENVI software is actually minimum noise fraction specified by (6.27).

The argument outlined above by (6.27) and (6.28) was based on Green *et al.*'s approach for each band image $\mathbf{b}_l$, not an entire hyperspectral image cube. As noted, the lth MNF-transformed band image vector $\tilde{\mathbf{b}}_l$ is obtained by (6.24) whose ith component $\tilde{r}_{li}$ is actually calculated by a weighted band correlation among the L bands within the ith image pixel vector via a particular weight vector $\mathbf{w}_l$. It may not be conceptually clear and easy to be understood from a hyperspectral image viewpoint as an image cube. However, the connection between Green *et al.*'s MNF transform and Lee *et al.*'s NAPC can be better understood if a hyperspectral image is presented as a data matrix as follows. Following the same notations used in the MNF transform, assume that an L-band hyperspectral image has N image pixels denoted by$\{\mathbf{r}_i\}_{i=1}^{N}$ with $N = n_r n_c$ where n_r and n_c denote the number of rows and columns in the image, respectively. Also, let $\mathbf{b}_l =$

$(b_{l1}, b_{l2}, \ldots, b_{lN})^T$ be an N-dimensional column vector that represents the lth band image of the hyperspectral image. Then the relationship between L-dimensional image pixel vectors $\{\mathbf{r}_i\}_{i=1}^{N}\mathbf{r}_i$ and L band images $\{\mathbf{b}_l\}_{l=1}^{L}$can be related by the following data matrix $\mathbf{X} = [\mathbf{r}_1\mathbf{r}_2 \cdots \mathbf{r}_N]$:

$$\mathbf{X} = [\mathbf{r}_1\mathbf{r}_2 \ldots \mathbf{r}_N] = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1(N-1)} & r_{1N} \\ r_{11} & r_{11} & \ddots & \ddots & r_{2N} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ r_{(L-1)1} & \ddots & \ddots & r_{(L-1)(N-1)} & r_{(L-1)N} \\ r_{L1} & r_{L1} & \cdots & r_{L(N-1)} & r_{LN} \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1^T \\ \mathbf{b}_2^T \\ \vdots \\ \mathbf{b}_{L-1}^T \\ \mathbf{b}_L^T \end{bmatrix} \tag{6.29}$$

and

$$b_{lk} = r_{lk} \text{ for } 1 \leq l \leq L \text{ and } 1 \leq k \leq N \tag{6.30}$$

According to (6.29) and (6.30), Green *et al.*'s MNF transform performs on the left-hand side of (6.30) band-by-band images in a similar fashion as a remotely sensed image is stored by the Band SeQuential (BSQ) (Schowengerdt, 1997, p. 25). On the other hand, Lee *et al.*'s NAPC transform processes a hyperspectral image as the data matrix, that is, $\mathbf{X} = [\mathbf{r}_1\mathbf{r}_2 \cdots \mathbf{r}_N]$ on the left-hand side of (6.29) in the same way as a remotely sensed image is stored by the band-interleaved-by-pixel (BIP) (Schowengerdt, 1997, p. 26). Therefore, in the NAPC transform, the sample data covariance matrix is obtained by $\mathbf{K} = \left\{(1/N)\left[\mathbf{X}\mathbf{X}^T\right]\right\} - \boldsymbol{\mu}\boldsymbol{\mu}^T = (1/N)\left[\sum_{i=1}^{N}(\mathbf{r}_i - \boldsymbol{\mu})(\mathbf{r}_i - \boldsymbol{\mu})^T\right]$ and the noise covariance matrix, $\mathbf{K_n}$ is estimated from the data matrix $\mathbf{X}$ (Lee *et al.* 1990). A fast algorithm derived by Roger (1994) to implement NAPC transform is summarized as follows.

Algorithm for NAPC Transform

1. Find a whitening matrix $\mathbf{F}$ to orthonormalize $\mathbf{K_n}$ such that

$$\mathbf{F}^T\mathbf{K_n}\mathbf{F} = \mathbf{I} \text{ and } \mathbf{F}^T\mathbf{F} = \mathbf{D}_\mathbf{n}^{-1} \tag{6.31}$$

   where $\mathbf{D_n}$ is the diagonal variance matrix of $\mathbf{K_n}$.
2. Find the resulting noise-adjusted data covariance matrix given by $\mathbf{K}_{\text{adj}} = \mathbf{F}^T\mathbf{K}\mathbf{F}$.
3. Find an eigenvector matrix resulting from PCA operating on $\mathbf{K}_{\text{adj}}$, denoted by $\mathbf{H}$ such that

$$\mathbf{H}^T\mathbf{K}_{\text{adj}}\mathbf{H} = \mathbf{D}_{\text{adj}} \text{ and } \mathbf{H}^T\mathbf{H} = \mathbf{I} \tag{6.32}$$

   where $\mathbf{D}_{\text{adj}}$ is the diagonal variance matrix of $\mathbf{K}_{\text{adj}}$.
4. Finally, the desired NAPC transform can be derived by

$$\boldsymbol{\Lambda}^{\text{NAPC}} = \mathbf{HF} \tag{6.33}$$

   Now, let $\left\{\mathbf{w}_l^{\text{NAPC}}\right\}_{l=1}^{L}$ be the NAPC transform vectors obtained from $\boldsymbol{\Lambda}^{\text{NAPC}}$ in (6.33), that is, $\boldsymbol{\Lambda}^{\text{NAPC}} = \left[\mathbf{w}_1^{\text{NAPC}}\mathbf{w}_2^{\text{NAPC}} \ldots \mathbf{w}_L^{\text{NAPC}}\right]$ that is similar to (6.2). Then $\left\{\mathbf{w}_l^{\text{NAPC}}\right\}_{l=1}^{L}$ arrange band images in accordance with descending order of SNR.

According to (6.25) and (6.33), MNF transform and NAPC transform achieve DR by only retaining first q projection vectors $\left\{\mathbf{w}_l^{\text{MNF}}\right\}_{l=1}^{q}$ and $\left\{\mathbf{w}_l^{\text{NAPC}}\right\}_{l=1}^{q}$ that correspond to the q largest SNRs

One major disadvantage of implementing MNF or NAPC transform is estimation of noise covariance matrix. Since it is based on the criterion of SNR, reliable noise estimation must be guaranteed. For details of estimation of noise covariance matrix we refer to Section 17.3 in Chang (2003a).

6.3 Dimensionality Reduction by High-Order Statistics-Based Components Analysis Transforms

Recall that $\{\mathbf{r}_i\}_{i=1}^N$ is the set of image pixel vectors in a hyperspectral image where $\mathbf{X} = [\mathbf{r}_1\mathbf{r}_2 \ldots \mathbf{r}_N]$ is a data matrix that represents an image cube as an $L \times N$ data matrix formed by $\{\mathbf{r}_i\}_{i=1}^N$ with $N = n_r n_c$. Let $\mathbf{w}$ be an L-dimensional column vector and assumed to be a desired projection vector. Then $\mathbf{z} = \mathbf{w}^T\mathbf{X} = (z_1, z_2, \ldots, z_N)^T$ is an $N \times 1$ column vector that represents the projection of the entire hyperspectral image pixel vectors $\{\mathbf{r}_i\}_{i=1}^N$ being mapped along the direction of $\mathbf{w}$. Now, assume that $F(\cdot)$ is a function to be explored and defined on the projection space $\mathbf{z} = \mathbf{w}^T\mathbf{X}$. The selection of the function F depends on various applications. For example, in order to detect small targets in a large unknown background, skewness and kurtosis are generally used as criteria to measure asymmetry and flatness of a distribution, respectively. In this case, $F(.)$ can be defined by skewness (κ_3) as follows:

$$\begin{aligned} F(z_i) = \kappa_3(z_i) &= \left(E\left[(z_i - \mu)^3\right]\right)/\sigma^3 \\ &= \left(E\left[(\mathbf{w}^T\mathbf{r}_i - \mu)^3\right]\right)/\sigma^3 \end{aligned} \quad \text{for each } i = 1, 2, \ldots, N \tag{6.34}$$

that is the normalized third central moment or kurtosis (κ_4)

$$\begin{aligned} F(z_i) = \kappa_4(z_i) &= \left(E\left[(z_i - \mu)^4\right]\right)/\sigma^4 \\ &= \quad \left(E\left[(\mathbf{w}^T\mathbf{r}_i - \mu)^4\right]\right)/\sigma^4 \end{aligned} \quad \text{for each } i = 1, 2, \ldots, N \tag{6.35}$$

that is the normalized fourth central moment. The μ and σ in (6.34) and (6.35) are the mean and standard deviation of random variable z_i, respectively. Since small targets can be characterized by those pixels that cause maximal magnitude of asymmetry and ripples of a distribution, finding a projection vector $\mathbf{w}$ that maximizes (6.34) and (6.35) is equivalent to finding a direction which these pixels are most likely aligned with. By projecting all data samples $\{\mathbf{r}_i\}_{i=1}^N$ on the projection vector $\mathbf{w}$, the desired small targets can be detected by those pixels that yield the largest projection along the direction of $\mathbf{w}$.

If we assume that most of image background can be described by second-order statistics and the statistical behaviors of targets of interest go beyond second-order statistics, a logical preprocessing for detecting such targets will remove the image background prior to target detection. In doing so, we first remove the sample mean and de-correlate the data matrix $\mathbf{X}$ by the sphering method described as follows.

6.3.1 Sphering

The idea of sphering is to centralize the mean of the data samples $\{\mathbf{r}_i\}_{i=1}^N$ at the origin while normalizing the data variances to one. This allows us to completely remove the first-order statistics and all the second-order statistics and focus on statistics orders higher than 2. Here, a remark on sphering is noteworthy. Despite that many choose "sphering" to equate "whitening," a concept widely used in communications and signal processing, and consider both concepts to be the same. Technically speaking, they are not equivalent. A whitening process or filter operates an input process that can be any random process and outputs a white process (see (6.56)) that is a zero-mean random process with all samples being uncorrelated (Therrien, 1992). In other words, a white process is a random process whose mean is zero and power spectral density function (power density spectrum) is a constant that implies that all samples resulting from a white process are completely uncorrelated. With this interpretation, we consider "*sphering*" to be different from "*whitening*" in the sense that the former make statistics of second order go away by normalizing variances to one as opposed to the latter that only de-correlates the data by nullifying co-variances while still retaining variances. Consequently, the covariance matrix of the sphered data is an identity matrix compared to a whitened data whose covariance matrix is a diagonal matrix, but not necessarily an identity matrix.

As a result of sphering the data, two sets of data samples can be categorized. One set is made up of all data samples lying on the surface of a sphere centered at the origin with unit radius. The set of these data samples represents uninteresting data samples that may include most image background pixels. The second set of data samples contains all data samples that are not on the sphere, that is, either inside or outside the sphere. Only these data samples are of major interest and can be further explored by orders of statistics higher than variance. So, working only on this set of data samples may exclude most of image background samples.

In order to perform sphering, we first remove the sample mean of data set by $\tilde{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu} \cdot \mathbf{1}^T = [\mathbf{r}_1 - \boldsymbol{\mu}, \mathbf{r}_2 - \boldsymbol{\mu}, \ldots, \mathbf{r}_N - \boldsymbol{\mu}]$, where $\boldsymbol{\mu} = (1/N)\sum_{i=1}^{N} \mathbf{r}_i$ is the sample mean vector and $\mathbf{1} = [\underbrace{11\cdots 1}_{N}]^T$ is column vector with all ones in the components. Next step we will de-correlate the zero-mean data sample matrix $\tilde{\mathbf{X}}$.

Assume that $\{\lambda_l\}_{l=1}^{L}$ are the eigenvalues of the sample covariance matrix $\mathbf{K}_{\tilde{\mathbf{X}}} = (1/N)\tilde{\mathbf{X}}\tilde{\mathbf{X}}^T$ formed by $\tilde{\mathbf{X}}$ and $\{\mathbf{v}_l\}_{l=1}^{L}$ be their corresponding eigenvectors with $||\mathbf{v}_l|| = 1$. Suppose that $\boldsymbol{\Lambda} = [\mathbf{v}_1\mathbf{v}_2 \ldots \mathbf{v}_L]$ is the eigenvector matrix formed by $\{\mathbf{v}_l\}_{l=1}^{L}$. The covariance matrix can be decomposed into

$$\boldsymbol{\Lambda}^T\mathbf{K}_{\hat{\mathbf{X}}}\boldsymbol{\Lambda} = \mathbf{D}_\lambda \tag{6.36}$$

Let $\mathbf{D}_\lambda = \begin{bmatrix} \lambda_1 & 0 & \mathbf{0} \\ 0 & \ddots & 0 \\ \mathbf{0} & 0 & \lambda_L \end{bmatrix}$. Multiplying both sides of (6.36) by $\mathbf{D}_\lambda$ results in

$$(\mathbf{D}_\lambda)^{1/2}\boldsymbol{\Lambda}^T\mathbf{K}_{\tilde{\mathbf{X}}}\boldsymbol{\Lambda}(\mathbf{D}_\lambda)^{-1/2} = \mathbf{I} \tag{6.37}$$

From (6.27), we obtain the desired sphering matrix $\mathbf{A}$, given by

$$\mathbf{A} = \boldsymbol{\Lambda}(\mathbf{D}_\lambda)^{-1/2} \tag{6.38}$$

so that $\mathbf{A}^T\mathbf{K}_{\tilde{\mathbf{X}}}\mathbf{A} = \mathbf{I}$. The data set resulting from applying the sphering matrix $\mathbf{A}$ to the original data set, $\{\mathbf{r}_i\}_{i=1}^{N}$, is denoted by $\{\tilde{\mathbf{r}}_i\}_{i=1}^{N}$ and the process of using (6.37) and (6.38) is called sphering that is also known as a whitening process of $\mathbf{X}$. In this case, the data matrix $\mathbf{Y}$ has zero mean and an identity matrix as its covariance matrix.

If we replace the original data samples $\{\mathbf{r}_i\}_{i=1}^{N}$ with sphered data samples $\{\tilde{\mathbf{r}}_i\}_{i=1}^{N}$, then $\tilde{\mathbf{z}} = \mathbf{w}^T\tilde{\mathbf{X}} = \mathbf{w}^T[\tilde{\mathbf{r}}_1\tilde{\mathbf{r}}_2 \cdots \tilde{\mathbf{r}}_N] = (\mathbf{w}^T\tilde{\mathbf{r}}_1, \mathbf{w}^T\tilde{\mathbf{r}}_2, \ldots, \mathbf{w}^T\tilde{\mathbf{r}}_N)^T = (\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_N)^T$. Both the skewness in

(6.34) and the kurtosis in (6.35) are reduced to

$$\kappa_3(\tilde{z}_i) = E\left[\left(\mathbf{w}^T\tilde{\mathbf{r}}_i\right)^3\right] \text{ for each } i = 1, 2, \ldots, N \tag{6.39}$$

and

$$\kappa_4(\tilde{z}_i) = E\left[\left(\mathbf{w}^T\tilde{\mathbf{r}}_i\right)^4\right] \text{ for each } i = 1, 2, \ldots, N \tag{6.40}$$

respectively where $\{\tilde{\mathbf{r}}_i\}_{i=1}^N$ are considered as random vectors.

6.3.2 Third-Order Statistics-Based Skewness

After the data are sphered, the next task is to search for a projection vector that is optimal in some sense. If the skewness is used as a criterion, the projection vector should be the one that points to the direction where the projected data has the most asymmetric histogram. If the kurtosis is used as criterion, the projected data will yield the most heavy-tailed histogram. To find the projector that yields the maximal skewness, we impose a constrained problem as follows:

$$\begin{aligned}&\max_{\mathbf{w}}\left\{(1/N)\textstyle\sum_{i=1}^N z_i^3\right\}\\ &= \max_{\mathbf{w}}\left\{(1/N)\textstyle\sum_{i=1}^N \mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\right\}\end{aligned} \quad \text{subject to } \mathbf{w}^T\mathbf{w} = 1 \tag{6.41}$$

where z_i is the projection resulting from the sphered data sample $\mathbf{y}_i$ via the projection vector $\mathbf{w}$. The constraint $\mathbf{w}^T\mathbf{w} = 1$ is used for normalization such that the skewness of the resulting data after projection will not be affected by the magnitude of $\mathbf{w}$. Using the Lagrange multiplier method, an objective function is obtained by

$$J(\mathbf{w}) = E\left[\mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\right] - \lambda(\mathbf{w}^T\mathbf{w} - 1) \tag{6.42}$$

where the expectation of $E\left[\mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\right]$ in (6.42) replaces the sample average $(1/N)\sum_{i=1}^N \mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i$ to indicate $\{\tilde{\mathbf{r}}_i\}_{i=1}^N$ are random vectors.

Differentiating (6.42) with respective $\mathbf{w}$ results in

$$\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}} = 3E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\tilde{\mathbf{r}}_i^T\right]\mathbf{w} - 2\lambda\mathbf{w} = 0 \tag{6.43}$$

Setting $\lambda' = (2/3)\lambda$ yields

$$\left(E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\tilde{\mathbf{r}}_i^T\right] - \lambda'\mathbf{I}\right)\mathbf{w} = 0 \tag{6.44}$$

Solving (6.44) is equivalent to finding the eigenvalue λ' of the matrix $E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\tilde{\mathbf{r}}_i^T\right]$ and its corresponding eigenvector $\mathbf{w}^*$.

6.3.3 Fourth-Order Statistics-Based Kurtosis

When the kurtosis is used as an optimal criterion, the constrained problem specified by (6.41) becomes

$$\begin{array}{l}\max_{\mathbf{w}}\left\{(1/N)\sum_{i=1}^{N} z_i^4\right\} \\ = \max_{\mathbf{w}}\left\{(1/N)\sum_{i=1}^{N} \mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\right\}\end{array} \quad \text{subject to } \mathbf{w}^T\mathbf{w} = 1 \tag{6.45}$$

and the Lagrangian is given by

$$J(\mathbf{w}) = E\left[\mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\right] - \lambda(\mathbf{w}^T\mathbf{w} - 1) \tag{6.46}$$

Differentiating (6.46) with respect to $\mathbf{w}$ and setting $\lambda' = (1/2)\lambda$ results in

$$\left(E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\right] - \lambda'\mathbf{I}\right)\mathbf{w} = 0 \tag{6.47}$$

that once again is to solve the eigenvalue λ' and its associated eigenvector $\mathbf{w}^*$ of the matrix $E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\right]$. The obtained $\mathbf{w}^*$ is a desired projector that yields the maximum kurtosis.

6.3.4 High-Order Statistics

In Sections 6.3.2 and 6.3.3 we discussed DR transforms using skewness and kurtosis as optimal criteria to find projection vectors. In this section, we extend their ideas to statistics of order higher than 4. Using the fifth normalized central moment (refer to as fifth moment) as a criterion and following the same derivation in Sections 6.3.2 and 6.3.3, (6.47) can be modified as

$$\left(E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\tilde{\mathbf{r}}_i^T\right] - \lambda'\mathbf{I}\right)\mathbf{w} = 0 \tag{6.48}$$

Similarly, with the sixth normalized central moment (refer to as sixth moment) as a criterion, (6.44) becomes

$$\left(E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\right] - \lambda'\mathbf{I}\right)\mathbf{w} = 0 \tag{6.49}$$

Since $\tilde{\mathbf{r}}_i^T\mathbf{w} = \mathbf{w}^T\tilde{\mathbf{r}}_i$ and it is a scalar, (6.48) and (6.49) can be expressed, respectively, as follows:

$$\left(E\left[\tilde{\mathbf{r}}_i\left(\tilde{\mathbf{r}}_i^T\mathbf{w}\right)^3\tilde{\mathbf{r}}_i^T\right] - \lambda'\mathbf{I}\right)\mathbf{w} = 0 \tag{6.50}$$

and

$$\left(E\left[\tilde{\mathbf{r}}_i\left(\tilde{\mathbf{r}}_i^T\mathbf{w}\right)^4\tilde{\mathbf{r}}_i^T\right] - \lambda'\mathbf{I}\right)\mathbf{w} = 0 \tag{6.51}$$

Therefore, using the kth normalized central moment as an optimal criterion for optimal projector is equivalent to solving the following eigenproblem

$$\left(E\left[\tilde{\mathbf{r}}_i\left(\tilde{\mathbf{r}}_i^T\mathbf{w}\right)^{k-2}\tilde{\mathbf{r}}_i^T\right] - \lambda'\mathbf{I}\right)\mathbf{w} = 0 \tag{6.52}$$

Let $\mathbf{w}^*$ denote the eigenvector of $E\left[\tilde{\mathbf{r}}_i\left(\tilde{\mathbf{r}}_i^T\mathbf{w}\right)^{k-2}\tilde{\mathbf{r}}_i^T\right]$. Using the property of eigendecomposition, (6.52) is reduced to

$$\mathbf{w}^T\left(E\left[\tilde{\mathbf{r}}_i\left(\tilde{\mathbf{r}}_i^T\mathbf{w}\right)^{k-2}\tilde{\mathbf{r}}_i^T\right]\right)\mathbf{w} = \lambda' \tag{6.53}$$

because of $\mathbf{w}^T\mathbf{w} = 1$. Simplifying the left-hand side of (6.53) yields

$$E\left[\mathbf{w}^T\tilde{\mathbf{r}}_i\left(\tilde{\mathbf{r}}_i^T\mathbf{w}\right)^{k-2}\tilde{\mathbf{r}}_i^T\mathbf{w}\right] = E\left[\left(\tilde{\mathbf{r}}_i^T\mathbf{w}\right)^k\right] = E\left[\tilde{z}_i^k\right] \tag{6.54}$$

So the corresponding eigenvalue λ' in (6.53) is the kth central moment of $\tilde{\mathbf{z}} = (\mathbf{w}^*)^T\tilde{\mathbf{X}}$. If $k = 2$, (6.54) is reduced to $E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\right]$, that is, the sample covariance matrix $\mathbf{K}_{\tilde{\mathbf{X}}}$ The $\mathbf{w}^*$ is the eigenvector of $\mathbf{K}_{\tilde{\mathbf{X}}}$ and the λ' is the variance of $\tilde{\mathbf{z}} = (\mathbf{w}^*)^T\tilde{\mathbf{X}}$, which is the standard principal components analysis (SPCA) in Section 6.2.1.2.

6.3.5 *Algorithm for Finding Projection Vectors*

It should be noted that a single projection vector $\mathbf{w}^*$ that solves (6.44) for skewness, (6.47) for kurtosis, or (6.52) for the kth moment represents only one component. In order to continuously generate new components, a sequence of projections must be performed. In this case, when a projector vector $\mathbf{w}^*$ is found, the de-correlated data $\tilde{\mathbf{X}}$ is then mapped into the linear subspace $\langle\mathbf{w}^*\rangle^\perp$ orthogonal to $\langle\mathbf{w}^*\rangle$ that is the space spanned by $\mathbf{w}^*$. The next projection vector $\mathbf{w}^*$ is then found by solving (6.44), (6.47), or (6.52) in the space $\langle\mathbf{w}^*\rangle^\perp$. The same procedure is then continued on until a stop criterion is satisfied such as predetermined number of projections required to be generated. An algorithm for finding a sequence of projection vectors is called projection vector generation algorithm (PVGA) and can be described as follows.

Projection vector generation algorithm

1. Initially, sphere the original data set $\mathbf{X}$ via (6.36)–(6.38). The resulting data set is denoted by $\tilde{\mathbf{X}}$.
2. Find the first projection vector $\mathbf{w}_1^*$ by solving (6.44), (6.47), or (6.52) depending on which criterion is used, skewness or kurtosis or the kth moment.
3. Use the found $\mathbf{w}_1^*$ to generate the first projection image $\tilde{\mathbf{Z}}^1 = \left(\mathbf{w}_1^*\right)^T\tilde{\mathbf{X}} = \left\{\tilde{\mathbf{z}}_i^1|\tilde{\mathbf{z}}_i^1 = \left(\mathbf{w}_1^*\right)^T\tilde{\mathbf{r}}_i\right\}$.
4. Apply the orthogonal subspace projector (OSP) specified by $P_{\mathbf{w}_1^*}^\perp = \mathbf{I} - \mathbf{w}_1^*\left(\left(\mathbf{w}_1^*\right)^T\mathbf{w}_1\right)^{-1}\left(\mathbf{w}_1^*\right)^T$ to the data set $\tilde{\mathbf{X}}$ to produce the first OSP-projected data set, denoted by $\tilde{\mathbf{X}}^1$, $\tilde{\mathbf{X}}^1 = P_{\mathbf{w}_1^*}^\perp\tilde{\mathbf{X}}$.
5. Use the data set $\tilde{\mathbf{X}}^1$ and find the second projection vector $\mathbf{w}_2^*$ by solving (6.44), (6.47), or (6.52) depending upon which criterion is used, skewness or kurtosis or the kth moment.
6. Use the found $\mathbf{w}_2^*$ to generate the second projection image $\tilde{\mathbf{Z}}^2 = \left(\mathbf{w}_2^*\right)^T\tilde{\mathbf{X}}^1 = \left\{\tilde{\mathbf{z}}_i^2|\tilde{\mathbf{z}}_i^2 = \left(\mathbf{w}_2^*\right)^T\tilde{\mathbf{r}}_i\right\}$.
7. Apply $P_{\mathbf{w}_2^*}^\perp = \mathbf{I} - \mathbf{w}_2^*\left(\left(\mathbf{w}_2^*\right)^T\mathbf{w}_2\right)^{-1}\left(\mathbf{w}_2^*\right)^T$ to the data set $\tilde{\mathbf{X}}^1$ to produce the second OSP-projected data set, denoted by $\tilde{\mathbf{X}}^2$, $\tilde{\mathbf{X}}^2 = P_{\mathbf{w}_2^*}^\perp\tilde{\mathbf{X}}^1$ that can be used to produce the third projection vector $\mathbf{w}_3^*$ by solving (6.44), (6.47), or (6.52). Or equivalently, we define a matrix projection matrix $\mathbf{W}^{*2} = \left[\mathbf{w}_1^*\mathbf{w}_2^*\right]$and apply $P_{\mathbf{W}^{*2}}^\perp = \mathbf{I} - \mathbf{W}^{*2}\left(\left(\mathbf{W}^{*2}\right)^T\mathbf{W}^{*2}\right)^{-1}(\mathbf{W}^{*2})^T$ to the original sphered data set $\tilde{\mathbf{X}}$ to obtain $\tilde{\mathbf{X}}^2 = P_{\mathbf{W}^{*2}}^\perp\tilde{\mathbf{X}}$.

8. Repeat the procedure of steps 5–7 over and over again to produce $\mathbf{w}_3^*, \ldots, \mathbf{w}_k^*$ until a stopping criterion is met. It should be noted that a stopping criterion can be either a predetermined number of projection vectors required to be generated or a predetermined threshold for the difference between two consecutive projection vectors.

It should be noted that the implementation of steps 2 and 6 in the PVGA is not trivial. In order to solve (6.44), (6.47), or (6.52) for the optimal projection vector $\mathbf{w}_1^*$, the following iterative procedure is proposed to execute the steps 2 and 6 where only the criterion (6.44) for skewness is used for illustration. Similarly, the same implementation can be used for kurtosis and the kth moment with (6.44) replaced with (6.47) and (6.52), respectively.

Implementation of steps 2 and 6 to execute (6.44) for skewness:

a. Initialize a random projector $\mathbf{w}_1^{(0)}$ and set $k = 0$.
b. Calculate the matrix $E\left[\tilde{\mathbf{r}}_i \tilde{\mathbf{r}}_i^T \mathbf{w}_1^{(k)} \tilde{\mathbf{r}}_i^T\right]$ and find an eigenvector $\mathbf{v}_1^{(k)}$ corresponding to the largest magnitude of eigenvalues of the matrix $E\left[\tilde{\mathbf{r}}_i \tilde{\mathbf{r}}_i^T \mathbf{w}_1^{(k)} \tilde{\mathbf{r}}_i^T\right]$.
c. If the Euclidean distance $||\mathbf{w}_1^{(k)} - \mathbf{v}_1^{(k)}|| > \varepsilon$ and $||\mathbf{w}_1^{(k)} + \mathbf{v}_1^{(k)}|| > \varepsilon$, then let $\mathbf{w}_1^{(k+1)} = \mathbf{v}_1^{(k)}$ and $k \leftarrow k+1$; go to step (b). Otherwise, $\mathbf{w}_1^{(k)}$ is the desired projector $\mathbf{w}_1^*$. Let $\mathbf{w}_1^* = \mathbf{w}_1^{(k)}$ and return to step 3 in the PVGA.

In order for a high-order statistics-based transform to achieve DR, we use PVGA to produce the first q projection vectors $\mathbf{w}_1^*, \ldots, \mathbf{w}_q^*$. For example, the skewness transform achieves DR by using PVGA to find only the first q projection vectors that repeatedly solve (6.41) using (6.44) via successive orthogonal subspace projections, the kurtosis transform achieves DR by using the PVGA to find only the first q projection vectors that repeatedly solve (6.45) using (6.47) via successive orthogonal subspace projections and the kth central moment transform achieves DR by using the PVGA to find only the first q projection vectors $\mathbf{w}^*$ that repeatedly solve (6.53) for λ' via successive orthogonal subspace projections.

6.4 Dimensionality Reduction by Infinite-Order Statistics-Based Components Analysis Transforms

In Section 6.3, transforms using the kth order of statistics with any $k \geq 2$ as an optimal criterion were presented. When the k becomes infinite, that is, statistics of infinite order, the approaches using (6.44), (6.47), and (6.52) in Section 6.3 are no longer applicable for $k = \infty$. To address this issue, two approaches have been investigated. One is Projection Pursuit (PP) discussed in Chapter 16 in Chang (2003a) that uses a projection index as a criterion to find an optimal projection vector. When a projection index is specified by a criterion of the kth order of statistics for $k \geq 2$, the PP is then reduced to transforms in Sections 6.2 and 6.3, particularly, PCA for $k = 2$, skewness for $k = 3$, and kurtosis for $k = 6$. The other is independent component analysis (ICA) that uses mutual information to de-correlate statistical dependency. Since a probability distribution can be fully described by its moment-generating function with infinite number of moments, theoretically ICA can be viewed as a transform using an infinite-order logical extension of any kth high-order component transforms and will be considered in this section.

ICA has received considerable interest in recent years because of its versatile applications ranging from blind source separation, channel equalization to speech recognition, and functional magnetic resonance imaging (Hyvarinen et al., 2001). The key idea of ICA assumes that data are linearly mixed by a set of separate independent signal sources and can then be used to demix these signal sources according to their statistical independency measured by mutual information. In order to validate its approach, an underlying assumption is that at most one source in the mixture model can be allowed to be a Gaussian source. This is due to the fact that a linear mixture of Gaussian sources is still a Gaussian source. More precisely, let **x** be an L-dimensional mixed signal source vector expressed by

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{6.55}$$

where **A** is an $L \times p$ mixing matrix and **s** is a p-dimensional signal source vector made up of p signal sources. The purpose of ICA is to find a demixing matrix **W** that separates the signal source vector **s** into a set of p sources that are statistically independent. Several different criteria have been proposed to measure source independency (Hyvarinen and Oja, 2001). Nevertheless, they all originated from the concept of mutual information that is a criterion to measure the discrepancy between two random sources (Cover and Thomas, 1991).

As a special case of (6.55), suppose that both **x** and **y** are zero-mean p-dimensional column random signal source vectors with covariance matrices $\boldsymbol{\Sigma}_{\mathbf{x}} = (1/p)[\mathbf{x}\mathbf{x}^T]$ and $\boldsymbol{\Sigma}_{\mathbf{y}} = (1/p)[\mathbf{y}\mathbf{y}^T]$, respectively. In order to de-correlate **x** in a similar fashion as the **x** is demixed in (6.55), a whitening matrix $\boldsymbol{\Sigma}_{\mathbf{x}}^{-1/2}$ defined by the inverse of the square root of the covariance matrix, $\boldsymbol{\Sigma}_{\mathbf{x}}$ can be used to whiten the signal source vector **x**. As a consequence, $\boldsymbol{\Sigma}_{\mathbf{y}} = \left(\boldsymbol{\Sigma}_{\mathbf{x}}^{-1/2}\right)\boldsymbol{\Sigma}_{\mathbf{x}}\left(\boldsymbol{\Sigma}_{\mathbf{x}}^{-1/2}\right)^T = \mathbf{I}$ and the resulting source vector **y** becomes an un-correlated signal source vector in analogy with the signal source vector **s** in (6.55) to become a statistical independent source vector resulting from a demixing matrix **W** found by ICA via (6.55). In light of this interpretation, (6.55) is reduced to

$$\mathbf{x} = \boldsymbol{\Sigma}_{\mathbf{x}}^{1/2}\mathbf{y} \Rightarrow \mathbf{y} = \boldsymbol{\Sigma}_{\mathbf{x}}^{-1/2}\mathbf{x} \tag{6.56}$$

where the mixing matrix **A** and the signal source vector **s** in (6.55) are replaced with the square-root of the covariance matrix, $\boldsymbol{\Sigma}_{\mathbf{x}}^{1/2}$and an uncorrelated random source vector **y,** respectively. By virtue of (6.56), the statistical independency measured by ICA is reduced to the second-order statistics de-correlation by PCA. Accordingly, ICA actually performs PCA on the p-dimensional correlated signal source vector **x** via a whitening matrix $\boldsymbol{\Sigma}_{\mathbf{x}}^{-1/2}$ to produce p uncorrelated PCs represented by the uncorrelated signal source vector **y** in terms of second-order statistics. The process of de-correlating the second-order statistics source vector **x** into an uncorrelated signal source vector **y** using (6.56) is generally referred to as whitening in signal processing and communications (Poor, 1994), with $\boldsymbol{\Sigma}_{\mathbf{x}}^{-1/2}$ being used as a whitened matrix similar to (6.38) in Section 6.3.1. The only difference between (6.55) and (6.56) is that the mixing matrix **A** in (6.55) is unknown as opposed to the covariance matrix, $\boldsymbol{\Sigma}_{\mathbf{x}}^{1/2}$ in (6.56) that can be calculated directly from the observed signal source vector **x**.

More interestingly, if we further interpret the mixed signal source vector **x**, the mixing matrix **A**, and **s** in (6.55) as a hyperspectral image pixel vector **r**, an image endmember matrix **M** and an abundance vector $\boldsymbol{\alpha}$, respectively, (6.55) becomes

$$\mathbf{r} = \mathbf{M}\boldsymbol{\alpha} \tag{6.57}$$

that is exactly the linear mixture model used in hyperspectral image analysis with no noise term due to the fact that the noise source is considered as one of sources to be separated. More details can be found in Chapter 15 of Chang (2003a).

Over the past years, DR is generally performed by PCA via (6.56) or (6.38). Interestingly, little work of applying ICA to DR has been reported in the literature except Wang and Chang (2006a). One possible reason is that ICA was not originally developed for the purpose of DR. A second reason may be that the similarity and relationship among the three equations, (6.55)–(6.57), have not been recognized. A third one is that the mixing matrices, $\mathbf{\Sigma}_{\mathbf{x}}^{1/2}$ in (6.56) and $\mathbf{M}$ in (6.57) are assumed to be known or can be generated directly from the data compared to the mixing matrix $\mathbf{A}$ in (6.55) that is totally unknown. Finally, unlike PCA that prioritizes its generated principal components according to the magnitude of eigenvalues, there is no specific criterion to rank components produced by ICA. Since ICA is a well-established technique, we will only focus on the above-described issues that arise in DR.

In order to implement ICA, the algorithm of FastICA developed by Hyvarinen and Oja (2001) is used to find ICs where the deflation approach was applied to generate ICs one by one sequentially and each of ICs is produced by maximizing the negentropy measured by kurtosis. Typically, non-Gaussianity can be measured by the absolute value of kurtosis. But, the kurtosis also has some drawbacks. It is very sensitive to outliers (Hyvarinen and Oja, 2001, p. 182) where a single sample can make the kurtosis very large. To alleviate this problem, some other criteria such as negentropy are introduced as a measure for non-Gaussianity. There are several approximations to negentropy using various other nonlinear functions, such as skewness, tanh, etc. According to our applications in hyperspectral target detection, the ICs of major interest are generally super-Gaussian that is usually caused by outliers. In this case, the kurtosis seems to be an appropriate criterion to be used to generate ICs. There is another symmetric approach (Hyvarinen and Oja, 2001, p. 194) that can be used to find ICs. However, this approach does not offer any advantage over the deflation approach in our applications. Therefore, it is not considered in this chapter.

For each spectral band image, it was converted to a vector. More specifically, assume that a hyperspectral image cube has size of $M \times N \times L$ where L is the number of spectral bands and MN is the size of each spectral band image. The hyperspectral image cube can then be represented by a data matrix $\mathbf{X}$ of size $L \times MN$ with L rows and MN columns. In other words, each row in the data matrix $\mathbf{X}$ is specified by a particular spectral band image. As a result, a total of L ICs can be generated by FastICA. However, there are some issues in implementing FastICA. First of all, FastICA-generated ICs are not necessarily in order of information significance as the way PCs are generated by PCA or MNF in accordance with decreasing magnitude of eigenvalues or SNRs. Another is that ICs generated by FastICA in different runs do not necessarily appear in the same order. These issues are primarily due to the nature that the initial projection unit vectors used to produce ICs via FastICA are randomly generated. Therefore, an IC generated earlier by FastICA is not necessarily more significant than the one generated later.

In order to resolve the issue in the use of the random initial projection unit vectors by FastICA, three approaches are developed. One is statistics-prioritized ICA-DR (SPICA-DR) that considers each generated IC as a random variable. Using this interpretation, we assume that the ith IC$_i$ can be described by a random variable ζ_i with values taken by the gray-level value of the nth pixel in the IC$_i$, denoted by z_n^i. In this case, FastICA-generated ICs can be ranked and prioritized by statistics-based criteria. Another is referred to as Random ICA-DR (RICA-DR) that considers FastICA as a random algorithm by taking advantage of the nature of randomness caused by initial unit projection vectors used by FastICA. Its idea is similar to that used for random endmember extraction algorithms (REEAs) discussed in Chapter 10 and is to run FastICA a number of times to produce sample average of all ICs where the ICs common in all runs will be considered as significant ICs and used for DR. A third approach is called Initialization-Driven ICA-DR (IDICA-DR) that is an approach complete opposite to RICA-DR. This idea is also similar to initialization-driven EEAs developed in Chapter 9. It removes the random nature caused by the initial projection unit vectors used by FastICA by using a custom-designed initialization algorithm to produce an appropriate set of initial projection vectors to initialize FastICA. Consequently, the ICs are prioritized by the projection vectors generated by the initialization algorithm.

6.4.1 Statistics-Prioritized ICA-DR (SPICA-DR)

Assume that the ith IC$_i$ can be described by a random variable ζ_i with values taken by the gray-level value of the nth pixel in the IC$_i$, denoted by z_n^i.

SPICA-DR algorithm

1. Assume that the number of dimensions required to be retained is q.
2. Use FastICA to find $2q$ independent components, $\{\text{IC}_i\}_{i=1}^{2q}$. It should be noted that for each IC FastICA randomly generates a unit vector as an initial projection vector to produce the final desired projection vector for that particular component.
3. Specify a statistics-based criterion J and calculate the Priority Score (PS), $\text{PS}_J(\text{IC}_i)$ for each of $\{\text{IC}_i\}_{i=1}^{2q}$.
4. Prioritize the $\{\text{IC}_i\}_{i=1}^{2q}$ in accordance with $\text{PS}_J(\text{IC}_i)$.
5. Select those ICs with the first p largest $\text{PS}_J(\text{IC}_i)$ to perform DR.

The statistics-based criterion used in step 3 of SPICA-DR can be chosen from one of the following measures to produce priority scores (PS). Since the data has been sphered, the second-order statistics are not included.

1. Sample mean of third-order statistics: skewness for ζ_i.

$$\text{PS}_{\text{skewbess}}(\text{IC}_i) = \left[\kappa_i^3\right]^2 \tag{6.58}$$

where $\kappa_i^3 = E\left[\zeta_i^3\right] = (1/MN)\sum_{n=1}^{MN}\left(z_n^i\right)^3$ is the sample mean of the third-order statistics in the IC$_i$.

2. Sample mean of fourth-order statistics: kurtosis for ζ_i.

$$\text{PS}_{\text{kurtosis}}(\text{IC}_i) = \left[\kappa_i^4\right]^2 \tag{6.59}$$

where $\kappa_i^4 = E\left[\zeta_i^4\right] = (1/MN)\sum_{n=1}^{MN}\left(z_n^i\right)^4$ is the sample mean of the fourth-order statistics in the IC$_i$.

3. Combination of third- and fourth-order statistics for ζ_i:

$$\text{PS}_J(\text{IC}_i) = (1/12)\left[\kappa_i^3\right]^2 + (1/48)\left[\kappa_i^4 - 3\right]^2 \tag{6.60}$$

It should be note that (6.60) is taken from (6.35) in Hyvarinen and Oja (2001, p. 115), which is used to measure the negentropy by high-order statistics.

4. Sample mean of kth-order statistics: kth moments for ζ_i:

$$\text{PS}_{k-\text{thmoment}}(\text{IC}_i) = \left[\kappa_i^k\right]^2 \tag{6.61}$$

where $\kappa_i^k = E\left[\zeta_i^k\right] = (1/MN)\sum_{n=1}^{MN}\left(z_n^i\right)^k$ is the sample mean of the kth moment of statistics in the IC$_i$.

5. Entropy

$$\text{PS}_{\text{entropy}}(\text{IC}_i) = -\sum_{j=1}^{MN} p_{ij}\log p_{ij} \tag{6.62}$$

where $p_i = (p_{i1}, p_{i2}, \ldots, p_{iG})^T$ is the probability distribution derived from the image histogram of IC$_i$ and G is the total number of image gray scales.

6. Information divergence (ID)

$$\mathrm{PS}_{\mathrm{ID}}(\mathrm{IC}_i) = \sum\nolimits_{j=1}^{MN} p_{ij} \log\left(p_{ij}/q_j\right) \tag{6.63}$$

where $\mathbf{p}_i = (p_{i1}, p_{i2}, \ldots, p_{iMN})^T$ is the probability distribution derived from the image histogram of IC_i and $\mathbf{q}_i = (q_{i1}, q_{i2}, \ldots, q_{iMN})^T$ is the probability distribution derived from the image histogram of the sample mean image obtained by $(1/L)\sum_{i=1}^{L} \mathrm{IC}_i$.

It is worth noting that SPICA-DR is supposed to run and prioritize all the ICs, then select the first p prioritized ICs to achieve DR. However, in practice this is not necessary. Our experiments shows that, $2q$ seems to provide a good upper bound for the number of ICs needed to be generated by SPICA-DR.

6.4.2 Random ICA-DR

The idea of RICA-DR is to run FastICA a number of times where a run is defined as one implementation of running FastICA. Since the initial projection vectors randomly generated by FastICA for each run are different, the orders of the generated ICs will be also different. Nevertheless, if the information contained in an IC is significant, such information will always preserved in each run. With this assumption, if FastICA is run in number of times, the ICs containing the common information produced by all runs should be very close within a tolerance. In this case, the process is terminated and the ICs that are common in all runs are the desired ICs for DR. The detailed implementation of RICA-DR is summarized as follows.

RICA-DR algorithm

1. Initialization: Set $n = 0$ and the number of dimensions to be retained to q.
2. At each n, run FastICA to find $2q$ independent components, $\left\{\mathrm{IC}_i^{(n)}\right\}_{i=1}^{2q}$, where each independent component, $\mathrm{IC}_i^{(n)}$ can be formed as a vector, denoted by $\mathbf{v}_i^{(n)}$. It should be noted that FastICA randomly generates a unit vector as an initial projection vector.
3. If $n < 1$, $n \leftarrow n + 1$ and go to step 2. Otherwise, continue.
4. Find common ICs for all runs up to nth run. Two independent components for different runs, $\mathrm{IC}_i^{(\bar{n})}$ and $\mathrm{IC}_j^{(\tilde{n})}$, are considered to be distinct if the spectral angle mapper (SAM) between their corresponding vectors, $\mathrm{v}_i^{\bar{n}}$ and $\mathrm{v}_j^{\tilde{n}}$, is greater than a prescribed threshold ε. Let $\cap_{m=0}^{n}\left\{\mathrm{IC}_k^{(m)}\right\}_{k=1}^{2q}$ denote the common ICs obtained for all runs, $0 \leq m \leq n$.
5. If $\cap_{m=0}^{n-1}\left\{\mathrm{IC}_k^{(m)}\right\}_{k=1}^{2q} \neq \cap_{m=0}^{n}\left\{\mathrm{IC}_k^{(m)}\right\}_{k=1}^{2q}$, go to step 2. Otherwise, the algorithm is terminated and $\cap_{m=0}^{n}\left\{\mathrm{IC}_k^{(m)}\right\}_{k=1}^{2q}$ is the desired set of ICs for DR.

It is worth noting that RICA-DR automatically determine the value of q for DR. The use of the q can be used to limit the number of ICs that FastICA should generate for each run to save time so that FastICA does not have to run through all the ICs to find the common ICs. Like SPICA-DR, empirically, a good upper bound for RICA-DR seems $2q$.

6.4.3 Initialization Driven ICA-DR

In SPICA-DR and RICA-DR, the initial projection unit vectors used by FastICA to produce each of ICs are generated randomly. Therefore, the ICs produced by FastICA in different runs generally appear in different orders. When it comes to DR, this becomes a serious issue because an IC that appears earlier is not necessarily more important or significant than an IC produced later. In order to resolve this issue, an initialization-driven ICA-DR (IDICA-DR) is developed where the initial projection unit vector used to produce each of ICs are selected in a specific manner so that all the ICs will always appear in a fixed order rather than a random order as produced by SPICA-DR or RICA-DR. As a consequence, there is no need of using statistics-based criteria to prioritize ICs as SPICA-DR does or it requires FastICA to run a number of times with different random orders as the RICA-DR does to find the common ICs. A major advantage of using IDICA-DR is that all ICs always appear in the same order regardless of how many runs FastICA is implemented. The algorithm proposed to be used in IDICA-DR to generate a set of initial projector unit vectors is called automatic target generation process (ATGP) that is derived from the automatic target detection and classification algorithm (Ren and Chang, 2003, Chang, 2003a).

IDICA-DR algorithm

1. Assume that the number of dimensions required to be retained is q.
2. Perform sphering on the data matrix $\mathbf{X}$ and let the resulting sphered data matrix be denoted by $\hat{\mathbf{X}}$.
3. Apply ATGP to $\hat{\mathbf{X}}$ to find p target pixel vector $\{\mathbf{t}_i\}_i^q$.
4. Use FastICA to find q independent components, $\{\mathrm{IC}_i\}_{i=1}^q$ where the ith IC_i is generated by FastICA with the ith target pixel vector chosen to be the initial projection vector instead of being generated randomly.

Two comments on IDICA-DR are noteworthy.

1. There is a good reason to choose ATGP to produce initial projection unit vectors for each of ICs. This is because ATGP generates target pixels by a sequence of orthogonal subspace projection (OSP) that is also used by FastICA to produce a sequence of ICs. Therefore, ATGP-generated target pixels, $\{\mathbf{t}_i\}_i^q$, are orthogonal to each other. This implies that one target pixel used as an initial projection vector to generate an IC will not be used again as an initial projection vector to generate other ICs.
2. It may seem intuitive to use eigenvectors as initial projection vectors to generate ICs. Unfortunately, our experiments show that this approach does not always work and its results are not consistent. On the other hand, the use of ATGP in IDICA-DR works better than the use of eigenvectors. This is mainly due to the fact that ATGP-generated vectors are always real target vectors from the data, while the eigenvectors are not but rather projection vectors used to specify components. Therefore, ATGP is chosen for IDICA-DR.

As a concluding remark, it is worthwhile discussing the three approaches developed in this section. First of all, the algorithms derived from these three approaches do not need human intervention. In particular, there is no try-out by users. These three approaches are also based on completely different design rationales and each of them deserves its own merit. SPICA-DR prioritizes ICs according to a statistics-based criterion. On the other hand, RICA-DR algorithm does not rely on any criterion. Instead, it runs itself like a random algorithm with an arbitrary set of random initial projection vectors in a given run. It is then terminated when different runs of FastICA produce common ICs. Due to such

Table 6.1 Comparison among three algorithms, SPICA-DR, RICA-DR, and IDICA-DR

	SPICA-DR	RICA-DR	IDICA-DR
Number of ICs required to generate	$2q$	$2q$	q
Number of ICs required to select	q	Automatic	q
Initial condition	Random	Random	ATGP
Criterion for IC prioritization	High-order statistics	No	ATGP
Criterion for IC selection	Priority	Automatic	ATGP
Computation	Low	Moderate-high	low
Algorithm used to generate ICA	FastICA	FastICA	FastICA

a random nature, RICA-DR requires FastICA to run a number of times with different sets of randomly generated projection vectors. In this case, the value of q is only used to estimate an upper bound on the number of ICs that FastICA must generate in each run. Compared to SPICA-DR and RICA-DR that use randomly generated vectors as initial projection vectors, IDICA-DR takes a completely opposite approach. It makes use of an initialization algorithm called ATGP to produce an appropriate set of initial projection vectors to be used for FastICA. Since the targets generated by ATGP are spectrally distinct in terms of orthogonal subspace projection (OSP), each of these ATGP-generated targets represents one type of signal sources. FastICA then uses these ATGP-generated targets as initial projection vectors to make sure that these ATGP-generated target sources are separated in individual ICs so no two or more ATGP-generated target sources are present in one single IC. Additionally, all the ICs will be also prioritized in a simple way by the same order as the ATGP-generated targets are generated. One salient difference among the three ICA-DR algorithms is the number of components required by FastICA to generate. SPICA-DR and RICA-DR are generally required to generate all the ICs before IC selection. However, it is found empirically that twice the value of q provides a good upper bound on the number of ICs required to be generated. Compared to SPICA-DR and RICA-DR, IDICA-DR only has to generate q ICs without performing IC selection as do SPICA-DR and RICA-DR. So, from a computational complexity point of view, IDICA-DR has least computing time and RICA-DR may experience highest computation since it must run FastICA in a number of runs before the algorithm is terminated. Moreover, SPICA-DR produces and then selects the first q ICs prioritized by a statistics-based criterion. Like SPICA-DR, RICA-DR also produces $2q$ ICs. The difference is that RICA-DR repeatedly runs FastICA with different sets of random initial projection vectors and, in the mean time, also finds their common set of ICs in all runs until the two consecutive runs produce the same common set of ICs. So, there is no need for RICA-DR to select or prioritize ICs. On the contrary, IDICA-DR generates q ICs by FastICA using a specific set of the q targets generated by the ATGP as its initial projection vectors. In this case, the ICs are prioritized and selected by the order the q ATGP-generated targets appear. As expected, the running time required for IDICA-DR is generally less than that for SPICA-DR and RICA-DR because the latter must generate $2q$ ICs, while the former always takes advantage of generating ICs one by one in sequence. Finally, we summarize comparison among these three algorithms in Table 6.1, where q is the number of dimensions to be retained after DR.

6.5 Dimensionality Reduction by Projection Pursuit-Based Components Analysis Transforms

The component analysis transforms discussed in Sections 6.3 and 6.4 are actually special cases of projection pursuit (PP)-based component analysis transforms to be presented in this section where projection vectors specified by various orders of statistics can be interpreted by a more general concept called projection index (PI). This section investigates the idea of representing high-order statistics-based component analysis transforms in the context of PP and further develops three approaches similar to the three ICA-based component analysis transforms in discussed in Sections 6.6.1–6.6.3 to implement the PI-based PP. The first one is to use PI as a criterion to produce components, referred to as projection index components (PIC). In light of this interpretation the PI specified by data variance makes PP become PCA with PICs reduced to PCs. On the other hand, if the PI is specified by mutual information to measure statistical independence, the resulting PP turns out to be ICA in which case PICs become independent components (ICs). While using random initial conditions is the classical PP approach, it actually results in inconsistent PICs. So, the second and third approaches are developed to mitigate this dilemma by eliminating the inconsistency caused by the use of random initial condition, which are random projection index-based projection pursuit (RPI-PI) that is similar to RICA-DR in discussed in Section 6.6.2 and component prioritization-based PP. As for the latter approach two ways are designed to prioritize PICs: PI-PRioritized PP (PI-PRPP) that uses PI as a criterion to prioritize PICs produced by an arbitrary PI-based PP, and initialization-driven PP (ID-PP) that makes use of a custom-designed initialization algorithm to produce a specific set of initial conditions for PP. It should be noted that it requires two PIs used to implement PI-PP for DR with one used to generate PICs and the other used to prioritize PICs. In general, these two PIs are not necessarily the same.

6.5.1 Projection Index-Based Projection Pursuit

This section presents a projection index (PI)-based dimensionality reduction technique known as project pursuit (PP) that uses a PI as a criterion to find directions of interestingness of data to be processed and then represents the data in the data space specified by these new interesting directions. With the context of PI PCA and ICA can be considered as special cases of PP in the sense that PCA uses data variance as a PI to produce eigenvectors while ICA uses mutual information as a PI to produce statistically independent projection vectors.

The term of "projection pursuit (PP)" was first coined by Friedman and Tukey (1974) that was used as a technique for exploratory analysis of multivariate data. The idea is to project a high-dimensional data set into a low-dimensional data space while retaining the information of interest. It designs a PI to explore projections of interestingness. Assume that there are N data points $\{\mathbf{x}_n\}_{n=1}^{N}$ each with dimensionality K; $\mathbf{X} = [\mathbf{r}_1\mathbf{r}_2 \ldots \mathbf{r}_N]$ is a $K \times N$ data matrix and $\mathbf{a}$ is a K-dimensional column vector that serves as a desired projection. Then $\mathbf{a}^T\mathbf{X}$ represents an N-dimensional row vector that is the orthogonal projections of all sample data points mapped onto the direction $\mathbf{a}$. Now if we let $H(\cdot)$ be a function measuring the degree of the interestingness of the projection $\mathbf{a}^T\mathbf{X}$ for a fixed data matrix $\mathbf{X}$, a projection index (PI) is a real-valued function of $\mathbf{a}$, $I(\mathbf{a}) : R^K \to R$ defined by

$$I(\mathbf{a}) = H(\mathbf{a}^T\mathbf{X}) \tag{6.64}$$

The PI can be easily extended to multiple directions $\{\mathbf{a}_j\}_{j=1}^{J}$. In this case, $\mathbf{A} = [\mathbf{a}_1\mathbf{a}_2 \ldots \mathbf{a}_J]$ is a $K \times J$ projection direction matrix and the corresponding projection index is also a real-valued function, $I(\mathbf{A}) : R^{K\times J} \to R$ is given by

$$I(\mathbf{A}) = H(\mathbf{A}^T\mathbf{X}) \tag{6.65}$$

The choice of the $H(\cdot)$ in (6.64) and (6.65) is application dependent. Its purpose is to reveal interesting structures within data sets, such as clustering. However, finding an optimal projection matrix $\mathbf{A}$ in (6.65) is not a simple matter (Chapter 16, Chang, 2003a). In this section, we focus on PIs that are specified by statistics of high orders such as skewness, kurtosis, etc. (Ren et al., 2006).

Assume that the ith projection index-projected component can be described by a random variable ζ_i with values taken by the gray level value of the nth pixel denoted by z_n^i. A general form for finding projection vectors specified by the kth-order statistics: kth moment solves the eigenproblem specified by (6.52). An algorithm that is similar to the one described in Section 6.6.5 for finding a sequence of projection vectors to solve (6.52) can be also derived for PI-based PP (PIPP) as follows.

Projection-index projection pursuit algorithm (PIPP)

1. Initially, assume that $\mathbf{X} = [\mathbf{r}_1\mathbf{r}_2 \ldots \mathbf{r}_N]$ is data matrix and a PI is specified.
2. Find the first projection vector $\mathbf{w}_1^*$ by maximizing the PI.
3. Use the found $\mathbf{w}_1^*$to generate the first projection image $\mathbf{Z}^1 = \left(\mathbf{w}_1^*\right)^T\mathbf{X} = \left\{\mathbf{z}_i^1 | \mathbf{z}_i^1 = \left(\mathbf{w}_1^*\right)^T\mathbf{r}_i\right\}$ to detect the first endmember.
4. Apply the orthogonal subspace projector (OSP) specified by $P_{\mathbf{w}_1}^{\perp} = \mathbf{I} - \mathbf{w}_1\left(\mathbf{w}_1^T\mathbf{w}_1\right)^{-1}\mathbf{w}_1^T$ to the data set $\mathbf{X}$ to produce the first OSP-projected data set denoted by $\mathbf{X}^1$, $\mathbf{X}^1 = P_{\mathbf{w}_1}^{\perp}\mathbf{X}$.
5. Use the data set $\mathbf{X}^1$ and find the second projection vector $\mathbf{w}_2^*$ by maximizing the same PI again.
6. Apply $P_{\mathbf{w}_2}^{\perp} = \mathbf{I} - \mathbf{w}_2(\mathbf{w}_2^T\mathbf{w}_2)^{-1}\mathbf{w}_2^T$ to the data set $\mathbf{X}^1$ to produce the second OSP-projected data set denoted by $\mathbf{X}^2$, $\mathbf{X}^2 = P_{\mathbf{w}_2}^{\perp}\mathbf{X}^1$ that can be used to produce the third projection vector $\mathbf{w}_3^*$ by maximizing the same PI again. Or equivalently, we define a projection matrix $\mathbf{W}^2 = [\mathbf{w}_1\mathbf{w}_2]$ and apply $P_{\mathbf{W}^2}^{\perp} = \mathbf{I} - \mathbf{W}^2\left(\left(\mathbf{W}^2\right)^T\mathbf{W}^2\right)^{-1}(\mathbf{W}^2)^T$ to the data set $\mathbf{X}$ to obtain $\mathbf{X}^2 = P_{\mathbf{W}^2}^{\perp}\mathbf{X}$.
7. Repeat the procedure of steps 5 and 6 over and over again to produce $\mathbf{w}_3^*, \ldots, \mathbf{w}_k^*$ until a stopping criterion is met. It should be noted that a stopping criterion can be either a predetermined number of projection vectors required to generate or a predetermined threshold for the difference between two consecutive projection vectors.

6.5.2 Random Projection Index-Based Projection Pursuit

As noted earlier, PI-PP described in Section 6.5.1 uses a randomly generated vector as an initial condition to produce each of its desired projection vectors that are used to generate PICs. As a result, a different randomly generated initial condition may converge to a different projection vector. Accordingly, final produced projection vectors will be different and so are their generated PICs. In other words, if a PP algorithm is performed at different times or by different users, the final PICs will be different. In order to correct this problem, this section presents a random projection index-based projection pursuit (R-PIPP). The idea of R-PIPP is derived directly from the RICA-DR in Section 6.6.2. It runs PI-PP a number of times where a run is defined as one implementation of PI-PP. Since the initial projection vectors randomly generated by PI-PP for each run are different, the orders of the generated PICs are also different. Nevertheless, if the information contained in a PIC is significant, such information will always be preserved in each run. Consequently, R-PIPP is terminated when the PICs generated by all runs contain nearly the same information. The detailed implementation of R-PIPP is almost identical to that of the RICA-DR described in Section 6.4.2 and can be summarized as follows.

R-PIPP algorithm

1. Initialization: Set $n = 0$ and the number of dimensions to be retained to q.
2. At each n, run the PI-PP to find $2q$ independent components, $\left\{\mathrm{PIC}_i^{(n)}\right\}_{i=1}^{2q}$ where each independent component, $\mathrm{PIC}_i^{(n)}$, can be formed as a vector denoted by $\mathbf{v}_i^{(n)}$.

3. If $n < 1$, then $n \leftarrow n + 1$ and go to step 2. Otherwise, continue.
4. Find common PICs for all runs up to nth run. Two independent components for different runs, $\text{PIC}_i^{(\bar{n})}$ and $\text{PIC}_j^{(\tilde{n})}$, are considered to be distinct if the spectral angle mapper (SAM) between their corresponding vectors, $\text{v}_i^{\bar{n}}$ and $\text{v}_j^{\tilde{n}}$, is greater than a prescribed threshold ε. Let $\cap_{m=0}^{n}\left\{\text{PIC}_k^{(m)}\right\}_{k=1}^{2q}$ denote the common PICs obtained for all runs $0 \le m \le n$.
5. If $\cap_{m=0}^{n-1}\left\{\text{PIC}_k^{(m)}\right\}_{k=1}^{2q} \neq \cap_{m=0}^{n}\left\{\text{PIC}_k^{(m)}\right\}_{k=1}^{2q}$, go to step 2. Otherwise, the algorithm is terminated and $\cap_{m=0}^{n}\left\{\text{PIC}_k^{(m)}\right\}_{k=1}^{2q}$ is the desired set of PICs for DR.

Like RICA-DR that automatically determines the value of q for DR, R-PIPP also does the same. However, in order to save computing time as well as ensure that no PICs are left out in the process, a good empirical upper bound for R-PIPP can be chosen to be $2q$.

6.5.3 Projection Index-Based Prioritized Projection Pursuit

An alternative to R-PIPP to remedy the randomness resulting from random initial conditions used by PIPP is PI-based prioritized PP (PI-PRPP) that uses a PI as a prioritization criterion to rank PP-generated PICs so that all PICs will be prioritized in accordance with the priorities measured by the specific PI. In this case, the PICs will be always ranked and prioritized by this PI in the same order regardless of what initial condition is used to produce a projection vector. It should be noted that there is a major distinction in using a PI to generate PP and a PI to generate PI-PRPP. While PP uses a PI as a criterion to produce a projection vector for each of PICs, PI-PRPP uses a PI to prioritize PICs that are already produced by PP using a different PI. Therefore, the PI used in both PP and PI-PRPP is not necessarily the same. As a matter of fact, on many occasions, they are different PIs in applications. In order to derive PI-PRPP various criteria similar to those specified by (6.58)–(6.63) in Section 6.4.1 can be used to prioritize PI-PP generated PICs as follows.

Projection index (PI)-based criteria

1. Sample mean of third-order statistics: skewness for ζ_j.

$$\text{PI}_{\text{skewness}}(\text{PIC}_j) = \left[\kappa_j^3\right]^2 \tag{6.66}$$

where $\kappa_j^3 = E\left[\zeta_j^3\right] = (1/MN)\sum_{n=1}^{MN}\left(z_n^j\right)^3$ is the sample mean of the third-order statistics in the PIC_j.

2. Sample mean of fourth-order statistics: kurtosis for ζ_i.

$$\text{PI}_{\text{kurtosis}}(\text{PIC}_j) = \left[\kappa_j^4\right]^2 \tag{6.67}$$

where $\kappa_j^4 = E\left[\zeta_j^4\right] = (1/MN)\sum_{n=1}^{MN}\left(z_n^j\right)^4$ is the sample mean of the fourth-order statistics in the PIC_j.

3. Sample mean of kth-order statistics: kth moments for ζ_j.

$$\text{PI}_{\text{k-moment}}(\text{PIC}_j) = \left[\kappa_j^k\right]^2 \tag{6.68}$$

where $\kappa_j^k = E\left[\zeta_j^k\right] = (1/MN)\sum_{n=1}^{MN}\left(z_n^j\right)^k$ is the sample mean of the kth moment of statistics in the PIC_j.

4. Negentropy: combination of third- and fourth-order statistics for ζ_j:

$$\mathrm{PI}_{\mathrm{negentropy}}(\mathrm{PIC}_j) = (1/12)\left[\kappa_j^3\right]^2 + (1/48)\left[\kappa_j^4 - 3\right]^2 \tag{6.69}$$

   It should be note that (6.69) is taken from (6.35) in Hyvarinen and Oja (2001, p. 115), which is used to measure the negentropy by high-order statistics.

5. Entropy

$$\mathrm{PI}_{\mathrm{entropy}}(\mathrm{PIC}_j) = -\sum\nolimits_{j=1}^{MN} p_{ji} \log p_j \tag{6.70}$$

   where $p_j = \left(p_{j1}, p_{j2}, \ldots, p_{jMN}\right)^T$ is the probability distribution derived from the image histogram of PIC_i.

6. Information divergence (ID)

$$\mathrm{PI}_{\mathrm{ID}}(\mathrm{PIC}_j) = \sum\nolimits_{j=1}^{MN} p_{ji} \log\left(p_{ji}/q_i\right) \tag{6.71}$$

   where $p_j = \left(p_{j1}, p_{j2}, \ldots, p_{jMN}\right)^T$ is the probability distribution derived from the image histogram of PIC_i and $\mathbf{q}_j = \left(q_{j1}, q_{j2}, \ldots, q_{jMN}\right)^T$ is the Gaussian probability distribution with the mean and variance calculated from PIC_i.

6.5.4 Initialization Driven Projection Pursuit

The PI-PRPP in Section 6.5.3 was intended to circumvent the issue that PICs may appear in a random order due to the use of randomly generated initial conditions. PI-PRPP allows users to prioritize PICs according to information significance measured by a specified PI. Despite the fact that the PICs ranked by PI-PRPP may appear in the same order regardless of different sets of random initial conditions, they are not necessarily identical because of randomness introduced by their used initial conditions. Although the discrepancy in two corresponding PICs may be minor compared to the appearance order of PICs without prioritization, the inconsistency may still cause difficulty in data analysis. Therefore, this section further develops a new approach called initialization-drive PP (ID-PP) that custom-designs an initialization algorithm to produce a specific set of initial conditions for PP so that the same initial conditions will be used all along in different runs of PP. As a consequence, such ID-PP generated PICs are always identical. One such initialization algorithm is ATGP used by the IDICA-DR in Section 6.4.3 and can be also used for the purpose of ID-PP.

ID-PP algorithm

1. Assume that the number of dimensions required to be retained is q.
2. Perform sphering on the data matrix $\mathbf{X}$ and let the resulting sphered data matrix be denoted by $\hat{\mathbf{X}}$.
3. Apply the ATGP to $\hat{\mathbf{X}}$ to find p target pixel vector $\{\mathbf{t}_i\}_i^q$.
4. Use the PP to find q PICs, $\{\mathrm{PIC}_i\}_{i=1}^q$ where the ith PIC_i is generated by the PP with the ith ATGP-generated target pixel vector chosen to be the initial projection vector instead of being generated randomly. Since ID-PP uses the same initial condition all the time, it always produces the same PICs in the same order.

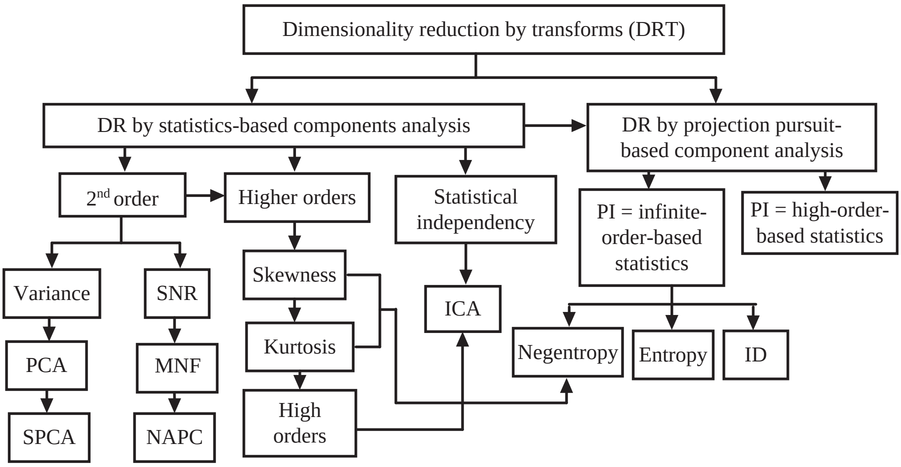

Dimensionality reduction by transforms (DRT)
DR by statistics-based components analysis
DR by projection pursuit-based component analysis
2nd order
Higher orders
Statistical independency
PI = infinite-order-based statistics
PI = high-order-based statistics
Variance
SNR
Skewness
ICA
Kurtosis
Negentropy
Entropy
ID
PCA
MNF
High orders
SPCA
NAPC

Figure 6.1 Various criteria used for DRT.

Finally, Figure 6.1 summarizes results in Sections 6.2–6.5.

6.6 Dimensionality Reduction by Feature Extraction-Based Transforms

Sections 6.2–6.4 present component transforms using criteria of statistics of any order. These transforms find a new set of component images that can represent the image data where each component image is specified by a projection vector produced by an appropriately selected criterion. In this section, we consider another type of transforms, feature transforms, to perform DR. Instead of representing image data by a set of component images, a feature transform projects image data into a feature space specified by a set of feature vectors that are obtained by a feature extraction-based criterion where each image data sample can be expressed in terms of its generated feature vectors. Two feature-based transforms, discriminant analysis and classification, are of interest and will be discussed in the following sections.

6.6.1 Fisher's Linear Discriminant Analysis

PCA uses data variances as an indication to point out the directions where the data cloud will be centered, but it does not necessarily point out the directions where different classes can be best separated. In order to resolve this issue, an approach called canonical analysis is developed (Richards and Jia, 1999) which uses a feature selection-based criterion that is the ratio of among-class variances to within-class variances so as to achieve best possible class separability. For two-class classification for image thresholding, the canonical analysis turns out to be exactly Otsu's thresholding method (Otsu, 1979) that is the most widely used to segment the foreground from the image background. As for multiclassification problems, the canonical analysis becomes a special case of Fisher's linear discriminant analysis (FLDA) discussed in Section 2.6.1.1 of Chapter 2. An application of FLDA to hyperspectral image classification was also explored (Du and Chang, 2001; Chang, 2003a; Ji et al., 2004). It finds a set of feature vectors that maximize Fisher's ratio described by (2.35) or (2.42) where each feature vector specifies one dimension. Since the number of feature vectors found by FLDA is determined by the number of classes, p, of interest, there are only $p - 1$ decision boundaries, each of which is determined by a feature vector. As a result, FLDA can achieve DR by letting $q = p - 1$ and converting the original data to only represent their classification features in a $(p - 1)$-dimensional feature space, where the feature dimensionality is generally much smaller than the data dimensionality. However, it should be noted that these $p - 1$ feature vectors are used to specify boundaries among p classes, not p class features as discussed in the OSP-based DR in the following section.

6.6.2 Orthogonal Subspace Projection

Another DR by feature transform is one recently proposed by Harsanyi and Chang (1994), orthogonal subspace projection (OSP), discussed in Section 2.3.2.1. The idea is to assume that each sample $\mathbf{r}$ in the original data can be represented by a linear mixture of a number of so-called endmembers, say p endmembers $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$, that are assumed to be known *a priori* with appropriate abundance fractions, α_1, α_2, $\ldots$, α_p associated with their respective endmembers. As a result, the entire data can be represented by a p-dimensional endmember space where each endmember is considered as a feature vector that one feature dimension. So, in order for the OSP to achieve DR, q must be set to p and a key issue is reduced to how to determine and find these p endmembers to represent the data. This issue will be addressed in four subsequent chapters, Chapters 7–10, 17 of this book.

Finally, it is worth noting that there is a prominent difference between FLDA and OSP in terms of feature transforms. FLDA represents data using $p - 1$ feature vectors to specify $p - 1$ class boundaries, while OSP expresses data in a linear representation of p feature vectors that are endmembers assumed to be present in the data.

6.7 Dimensionality Reduction by Band Selection

Since different spectral bands provide different levels of information of interest the primary goal of DRBS is to select an appropriate band subset from the original band set to represent the original data in some sense of optimality. Therefore, the information preserved by DRBS has significant impact on data analysis because the information of un-selected bands will be completely lost after BS. So, a key success in DRBS is how to design effective criteria for the BS to meet various applications. Solving a general BS problem generally requires an exhaustive search for all possible $\begin{pmatrix} L \\ |\Omega_{\mathrm{BS}}| \end{pmatrix} = \frac{L!}{(L-|\Omega_{\mathrm{BS}}|)!|\Omega_{\mathrm{BS}}|!}$ combinations, with L and $|\Omega_{\mathrm{BS}}|$ being the total number of spectral bands in Ω_{BS}, respectively.

More specifically, assume that $J(\cdot)$ is a generic objective function of Ω_{BS} for BS to be optimized. For a given number of selected bands, n_{BS}, a BS technique is to find an optimal band subset, Ω^*_{BS} with $|\Omega_{\mathrm{BS}}| = n_{BS}$, that satisfies the following optimization problem:

$$\Omega^*_{\mathrm{BS}} = \arg\left\{\max/\min_{\Omega_{BS}\subset\Omega,\ |\Omega_{BS}|=n_{\mathrm{BS}}} J(\Omega_{\mathrm{BS}})\right\} \tag{6.72}$$

Depending on how the objective function $J(\Omega_{\mathrm{BS}})$ is designed, the optimization in (6.72) can be performed by either maximization or minimization over all possible band subsets Ω_{BS} in Ω with $|\Omega_{\mathrm{BS}}| = n_{\mathrm{BS}}$.

According to (6.72), for a BS technique to be successful a key is how to define the objective function $J(\Omega_{\mathrm{BS}})$ that specifies an application of interest. An effective approach is to define a certain feature or criterion that can be used to specify the significance of a spectral band. For example, one of the most commonly criteria or features used for BS is to maximize data variance. Let $\mathbf{K}_{\Omega_{\text{BS}}}$ be the covariance matrix formed by all the bands selected in Ω_{BS}. We can define an objective function $J(\Omega_{\text{BS}})$ in (6.72) by

$$J_{\text{trace}}(\Omega_{\text{BS}}) = \text{tr}(\mathbf{K}_{\Omega_{\text{BS}}}) \tag{6.73}$$

where the operator "tr" is a trace of a matrix and defined as sum of all its diagonal entries. The optimal solution Ω_{BS}^* to (6.73) is the one that satisfies

$$\Omega_{\text{BS}}^* = \arg\left\{\max_{\Omega_{BS} \subset \Omega, |\Omega_{BS}| = n_{\text{BS}}} \text{tr}(\mathbf{K}_{\Omega_{\text{BS}}})\right\} \tag{6.74}$$

As an alternative, we can also perform PCA and then use data variance as a feature or criterion to represent the significance of a spectral band. In this case, let σ_l^2 be the data variance calculated for the lth spectral band, denoted by $\mathbf{B}_l$. Then the $J(\Omega_{\text{BS}})$ in (6.72) can be defined by

$$J_{\text{variance}}(\Omega_{\text{BS}}) = \sum\nolimits_{\mathbf{B}_l \in \Omega_{\text{BS}}} \sigma_l^2 \tag{6.75}$$

Replacing (6.73) with (6.75) yields

$$\Omega_{\text{BS}}^* = \arg\left\{\max_{\Omega_{BS} \subset \Omega, |\Omega_{BS}| = n_{\text{BS}}} \left[\sum\nolimits_{\mathbf{B}_l \in \Omega_{\text{BS}}} \sigma_l^2\right]\right\} \tag{6.76}$$

Since $\text{tr}(\mathbf{K}_{\Omega_{\text{BS}}}) = \sum_{\mathbf{B}_l \in \Omega_{BS}} \sigma_l^2$, the solutions given by (6.74) and (6.76) are identical. So, for a given n_{BS} the optimal solution to (6.74) or (6.76) is the band subset Ω_{BS}^* that consists of the n_{BS} spectral bands corresponding to the first n_{BS} largest data variances that turn out to be the first n_{BS} largest eigenvalues. Another common used criterion for the BS is to use an information-based criterion such as entropy to define the $J(\Omega_{\text{BS}})$ in (6.72) by

$$J_{\text{entropy}}(\Omega_{\text{BS}}) = H(\Omega_{\text{BS}}) \tag{6.77}$$

where $H(\Omega_{\text{BS}})$ is the entropy of the image cube formed by selected bands in Ω_{BS}. Over the past few years, many BS techniques have been investigated by designing various criteria or features to define $J(\Omega_{\text{BS}})$ in (6.72) (Mausel et al., 1990; Conese and Maselli, 1993; Stearns et al., 1993; Chang, 1999b). In what follows, we describe a rather different BS technique, constrained band selection algorithm, that has shown promise in BS.

6.8 Constrained Band Selection

By re-inventing a wheel Chang and Wang (2006) recently developed a new approach to BS, called constrained band selection (CBS). Its idea was derived from the Linearly Constrained Minimum Variance (LCMV) adaptive beamforming (Frost, 1972) and can be described as follows.

Let $\Omega_{\text{BS}} = \left\{\mathbf{b}_{\text{BS},1}, \mathbf{b}_{\text{BS},2}, \ldots, \mathbf{b}_{\text{BS},n_{\text{BS}}}\right\}$ denote the selected band subset where for each $1 \leq k \leq n_{\text{BS}}$ the kth spectral band in Ω_{BS} is represented by a column vector denoted by $\mathbf{b}_{\text{BS},k}$. Furthermore, let $\mathbf{b}_l$ represent the lth spectral band as a column vector. Then the entire image correlation matrix is defined by $\mathbf{Q} = \sum_{l=1}^{L} \mathbf{b}_l \mathbf{b}_l^T$. Following the LCMV approach outlined in Chang (2002b; 2003a) we can consider a CBS problem by finding a Finite Impulse Response (FIR) filter specified by a weighting matrix $\mathbf{W}_{\text{BS}}$ that linearly constrains the selected bands in Ω_{BS} with a constraint matrix, $\mathbf{C}$, while minimizing the band correlation matrix $\mathbf{Q}$ through the following optimization equation:

$$\min_{\mathbf{W}_{\mathrm{BS}}}\left\{\mathbf{W}_{\mathrm{BS}}^{T}\mathbf{Q}\mathbf{W}_{\mathrm{BS}}\right\} \text{ subject to } \mathbf{B}_{\mathrm{BS}}^{T}\mathbf{W}_{\mathrm{BS}}=\mathbf{C} \tag{6.78}$$

where $\mathbf{W}_{\mathrm{BS}}=\left[\mathbf{w}_{\mathrm{BS},1}\mathbf{w}_{\mathrm{BS},2}\cdots\mathbf{w}_{\mathrm{BS},n_{\mathrm{BS}}}\right]$, $\mathbf{B}_{\mathrm{BS}}=\left[\mathbf{b}_{\mathrm{BS},1},\mathbf{b}_{\mathrm{BS},2},\ldots,\mathbf{b}_{\mathrm{BS},n_{\mathrm{BS}}}\right]$, and $\mathbf{C}$ is a $n_{\mathrm{BS}}\times n_{\mathrm{BS}}$ constraint matrix. In light of the $J(\Omega_{\mathrm{BS}})$ in (6.72) the constrained objective function is defined by

$$J(\Omega_{\mathrm{BS}})=\mathbf{W}_{\mathrm{BS}}^{T}\mathbf{Q}\mathbf{W}_{\mathrm{BS}} \text{ subject to the constraint } \mathbf{B}_{\mathrm{BS}}^{T}\mathbf{W}_{\mathrm{BS}}=\mathbf{C} \tag{6.79}$$

The solution to (6.79), $\mathbf{W}_{\mathrm{BS}}^{\mathrm{LCMV}}$ is given by

$$\mathbf{W}_{\mathrm{BS}}^{\mathrm{LCMV}}=\mathbf{Q}^{-1}\mathbf{B}_{\mathrm{BS}}\left(\mathbf{B}_{\mathrm{BS}}^{T}\mathbf{Q}\mathbf{B}_{\mathrm{BS}}\right)^{-1}\mathbf{C} \tag{6.80}$$

Alternatively, we can exclude the selected band image correlation matrix $\frac{1}{|\Omega_{\mathrm{BS}}|}\sum_{\mathbf{b}\in\Omega_{\mathrm{BS}}}\mathbf{b}\mathbf{b}^{T}$ from the entire image correlation matrix $\mathbf{Q}$ and further define $\tilde{\mathbf{Q}}=\mathbf{Q}-\frac{1}{|\Omega_{\mathrm{BS}}|}\sum_{\mathbf{b}\in\Omega_{\mathrm{BS}}}\mathbf{b}\mathbf{b}^{T}$ as the selected band image dependence matrix. Replacing $\mathbf{Q}$ in (6.78) or (6.79) with $\tilde{\mathbf{Q}}$ results in a similar CBS problem given by

$$\min_{\mathbf{W}_{\mathrm{BS}}}\left\{\mathbf{W}_{\mathrm{BS}}^{T}\tilde{\mathbf{Q}}\mathbf{W}_{\mathrm{BS}}\right\} \text{ subject to } \mathbf{B}_{\mathrm{BS}}^{T}\mathbf{W}_{\mathrm{BS}}=\mathbf{C} \tag{6.81}$$

where $J(\Omega_{\mathrm{BS}})$ in (6.72) is now defined by

$$J(\Omega_{\mathrm{BS}})=\mathbf{W}_{\mathrm{BS}}^{T}\tilde{\mathbf{Q}}\mathbf{W}_{\mathrm{BS}} \text{ subject to the constraint } \mathbf{B}_{\mathrm{BS}}^{T}\mathbf{W}_{\mathrm{BS}}=\mathbf{C} \tag{6.82}$$

The solution to (6.82), $\tilde{\mathbf{W}}_{\mathrm{BS}}^{\mathrm{LCMV}}$ is

$$\tilde{\mathbf{W}}_{\mathrm{BS}}^{\mathrm{LCMV}}=\tilde{\mathbf{Q}}^{-1}\mathbf{B}_{\mathrm{BS}}\left(\mathbf{B}_{\mathrm{BS}}^{T}\tilde{\mathbf{Q}}^{-1}\mathbf{B}_{\mathrm{BS}}\right)^{-1}\mathbf{C} \tag{6.83}$$

that is exactly the same as the one in (6.80) with $\mathbf{Q}$ replaced by $\tilde{\mathbf{Q}}$.

Obviously, the major obstacle in using BS is to solve (6.72) for any given number of bands needed to be selected, n_{BS}. When n_{BS} is increased, the bands selected by (6.72) for a smaller number n_{BS} cannot be either included or used as prediction. The whole process of solving (6.72) must be re-carried out again. This is a significant disadvantage of using BS. In Chapter 21, a new concept called band prioritization is developed to resolve this issue where bands are prioritized and selected in accordance with their priorities calculated by a certain criterion, in which case previously selected bands in a smaller band subset are always included in a larger band subset without repeatedly solving (6.72) for different n_{BS}.

6.9 Conclusions

Dimensionality reduction (DR) is a commonly used data preprocessing technique to cope with vast amount of data volumes. This chapter presents two types of DR techniques: DR by Transform (DRT) and DR by band selection (DRBS). Two transformed-based techniques are developed for DR. One is statistics-based component transforms that represent image data by a set of component images that are statistically uncorrelated in terms of the used criterion. These include some popular and well-known PCA, MNF, and ICA transforms. The other is feature transforms that represent image data in a space specified by a set of feature vectors that can be obtained by a feature extraction-based criterion. Two such feature transforms are of interest: Fisher's linear discriminant analysis (FLDA) and orthogonal subspace projection (OSP). These DR transforms will be used as an initial pre-processing step in endmember extraction in PART II and exploitation-based hyperspectral information compression in PART V where a new concept of dimensionality prioritization (DP) extended from DRT will be introduced in Chapter 20 to perform progressive spectral dimensionality process (PSDP). In parallel to the development of DRT, a similar treatment can be also carried out for DRBS and will be discussed in detail in Chapter 21. Specifically, a concept similar to DP, referred to as band prioritization (BP), will be introduced in Chapter 21 as a counterpart of DP to perform progressive band dimensionality process (PBDP) as does DP for PSDP. However, there also exist other techniques developed for DR such as feature extraction-based wavelet transforms (Bruce et al., 2002) that can be used for the same purpose. The selection of DR techniques in this book is primarily based on the need of later chapters; we will use them as a preprocessing step prior to image processing. Finally, it should be noted that this chapter does not address a crucial and critical issue arising in DR, which is determination of the number of dimensions to be retained after DR, q. How to determine the value of the q can be investigated by a new concept called virtual dimensionality (VD) recently developed in Chang (2003a) and Chang and Du (2004) that is revisited and explored in detail in Chapter 5.

II

Endmember Extraction

Due to significantly improved high spatial and spectral resolution provided by hyperspectral imaging sensors endmember extraction has become increasingly important in hyperspectral image analysis. According to the definition given in Schwengerdt (1997), an endmember is an idealized, pure signature for a class, more specifically, spectral class. For multispectral imagery, an endmember may be difficult to find, due to the fact that many image pixels are generally heavily mixed because of its low spatial and spectral resolution. As a result, endmember extraction has received little interest in the past and its importance has been overlooked. By contrast, with recent advances in hyperspectral imaging sensors many subtle material substances that cannot be resolved by multispectral imagery can now be uncovered by hyperspectral imagery. These substances are generally not known *a priori* and can be only diagnosed by high spectral resolution. Endmembers are considered to be one of such substances. Their existence in image data cannot be generally detected by human eye. Most importantly, once endmembers are present, they may be very likely to appear as anomalies and their sample pools may be relatively small. Because of such characteristics, finding endmembers is very challenging. Many algorithms have been developed for this purpose, such as pixel purity index (PPI) (Boardman et al., 1995), N-finder algorithm (N-FINDR) (Winter, 1999a,b), iterative error analysis (IEA) (Neville et al., 1999), automated morphological endmember extraction algorithm (AMEEA) (Plaza et al., 2004), unsupervised fully constrained least squares (UFCLS) (Heinz and Chang, 2001), minimum volume transform (MVT) (Crag, 1994), convex geometry (Boardman, 1994), convex cone analysis (CCA) (Ifarraguerri and Chang, 1999), vertex component analysis (VCA) (Nascimento and Dias, 2005), simplex growing algorithm (SGA) (Chang et al., 2006) and independent component analysis-based endmember extraction algorithm (ICA-EEA) (Wang and Chang, 2006b), standardized principal components analysis (SPCA)-EEA, fully constrained least squares-EEA (FCLS-EEA), an alternative N-FINDR (AN-FINDR) (Ji, 2006), and statistics-based EEAs (Chu et al., 2007) that include PCA-EEA and high-order statistics-based EEA.

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

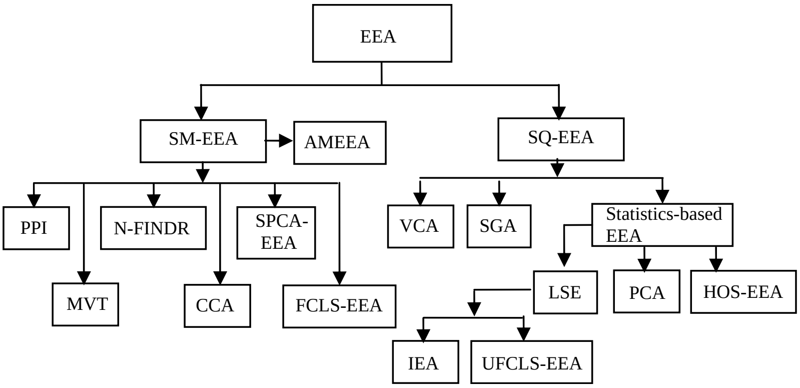


Figure II.1 Categorization of EEAs.

From an algorithmic implementation point of view, algorithms designed for endmember extraction can be performed in two different ways, simultaneous endmember extraction algorithms (SM-EEAs) and sequential endmember extraction algorithms (SQ-EEAs), as depicted in Figure II.1 where SM-EEA includes PPI, N-FINDR, MVT, CCA, SPCA-EEA, FCLS-EEA, and AMEEA, while the SQ-EEA includes IEA, VCA, SGA, UFCLS-EEA, and HOS-EEA.

Technically speaking, an optimal endmember extraction algorithm (EEA) must be an SM-EEA since all the endmembers should be selected at the same time rather than sequentially by SQ-EEAs. In general, finding all endmembers simultaneously requires tremendous computational complexity as a result of exhaustive search. On the other hand, an SQ-EEA may not be as optimal as an SM-EEA, but it may perform as well as an SM-EEA if the SQ-EEA is well-designed. Most advantageously, an SQ-EEA can reduce computations significantly.

For endmember extraction, two major criteria are of interest: convexity geometry and statistics. The criterion of convexity geometry can be further categorized into orthogonal projection (OP) and simplex volume. The criterion of OP is to orthogonally project data samples on a set of selected vectors so that the data samples whose orthogonal projections fall at the end (extreme) points of these selected vectors will be considered as endmember candidates. Such OP-based EEAs include PPI and VCA. The simplex volume criterion is to assume that a simplex formed by a set of pure signatures as vertices should yield the maximum volume among all simplexes formed by the same number of signatures as vertices. In this case, the vertices of the found simplex with the maximum volume will be considered as desired endmembers. Such simplex-based EEAs include MVT, CCA, N-FINDR, and SGA. Figure II.2 delineates various EEAs according to convexity geometry-based criteria.

The criterion of statistics has shown success in a variety of applications such as the dimensionality reduction (DR) in Chapter 6. The use of these statistics-based criteria for endmember extraction relies on the fact that a set of endmembers constitute the most uncorrelated sample pool among the same number of signatures where the correlation is measured by various statistics. Figure II.3 categorizes criteria of statistics into second-order statistics, statistical correlation and least squares error (LSE), and high-order statistics (HOS), skewness by third-order statistics, kurtosis by fourth-order statistics, *k*th moment by *k*th-order statistics, entropy specified by infinite order of statistics, and statistical independency measured by mutual information.

From a perspective of algorithmic initialization, an EEA can be implemented and initialized by two types of initial conditions: random initial endmembers and specific algorithm-generated initial

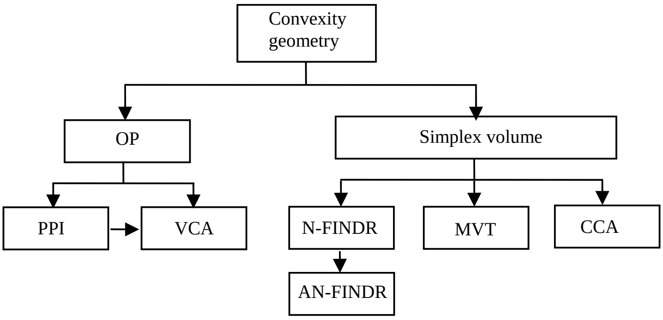


Figure II.2 Categorization of convexity geometry.

endmembers. In endmember extraction, most EEAs use randomly generated vectors as initial endmembers for initialization, such as PPI, N-FINDR, VCA, ICA-EEA, etc. Unfortunately, due to randomness a final set of endmembers produced by such an EEA is generally not repeatable. That is, if the same EEA is implemented in different runs with one run defined as one implementation of the EEA using one set of random initial endmembers, the final endmembers extracted by the same EEA will be different. In order to avoid such inconsistency in the final selection of endmembers, two approaches have been investigated and explored recently. One is to develop a custom-designed endmember initialization algorithm (EIA) to find an appropriate set of initial endmembers for an EEA, to eliminate uncertainty caused by randomness resulting from the use of random initial endmembers (Chang and Plaza, 2006a, 2006b). Such an EEA implemented in conjunction with an initialization algorithm is called initialization-driven EEA (ID-EEA). Another is to make the disadvantage of random initial endmembers an advantage. The idea is to consider an EEA that uses random initial endmembers as a random EEA where a single run resulting from a set of random initial endmembers is viewed as a realization of such a random EEA. If there is an endmember present in the data, it should eventually appear in the final set of endmembers generated by each run regardless of what initial endmembers are used by an EEA. In other words, if an EEA is implemented repeatedly with different sets of randomly generated initial endmembers, the desired endmembers should be among the common members in their final selection sets of endmembers generated by all runs. An REEA is terminated once the final generated endmember set remains unchanged in two consecutive runs. Such an EEA is called random EEA (REEA), also referred to

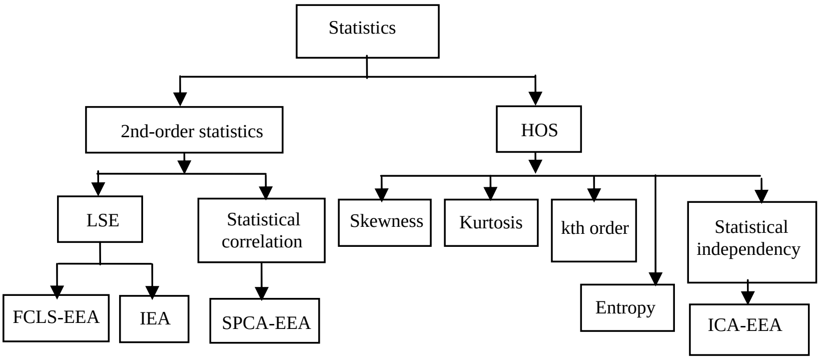


Figure II.3 Categorization of statistics.

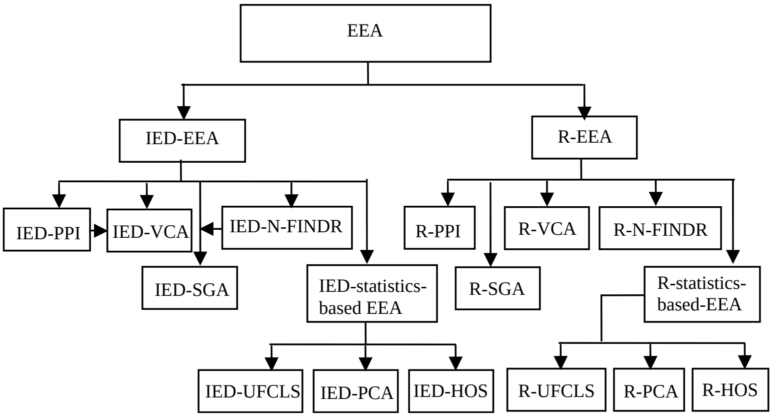


Figure II.4 Categorization of initial conditions, IED-EEA and REEA.

as automatic EEA (AEEA) in Chaudhry et al. (2006) and Wu and Chang (2006). Figure II.4 categorizes various EEAs according to initial conditions used in their algorithm design where IED-EEA is referred to as initial endmember-driven EEA.

PART II includes four chapters (Chapters 7–10) presented in the order of SM-EEA, SQ-EEA, ID-EEA, and REEA, and an additional chapter, Chapter 11, that is a comparative study and analysis among EEAs. Chapter 7 is focused on SM-EEA that generates all endmembers simultaneously. Algorithms of this type are PPI, N-FINDR, SPCA-EEA, FCLS-EEA, MVT, CCA, and AMEE. Chapter 8 deals with the other type of algorithms, SQ-EEA, that generates endmember one at a time sequentially, that is, one by one in succession. The group of this type is made up of UFCLS, IEA, VCA, SGA, and HOS-EEA. Since initial endmembers play a key role in the final selection of endmembers, two chapters, Chapters 9 and 10, are devoted to this issue. Chapter 9 develops custom-designed EIAs to produce an appropriate set of initial endmembers that can be used for an EEA. As a result, all the EEAs developed in Chapters 7 and 8 that make use of random initial endmembers can be extended to their counterpart ID-EEAs to resolve the inconsistency issue in their final selected endmembers. Contrary to Chapter 9, Chapter 10 makes the disadvantage of using random initial endmembers an advantage by extending all the EEAs using random initial endmembers developed in Chapters 7 and 8 to their random counterparts where a realization of an REEA is considered as an EEA using a particular set of random initial endmembers. By running an REEA as many times as possible all realizations should eventually converge to a common set of endmembers that should be the final set of desired endmembers.

Chapter 11 describes experiment-based comparative studies and analyses among EEAs and further investigates their relationships. Of particular interest is a comprehensive study on PPI where three versions of PPI, MATLAB-based PPI, ID-PPI, and RPPI are evaluated and an exploration of relationships among the three algorithms, PPI, VCA, and ATGP that provides interesting sights into rationales of algorithm design for EEA. More details are also discussed in Wu (2006, 2009).

In spite of an effort to categorize EEAs in what is believed to be logical, it is by no means the only way to do so. For example, PPI and N-FINDR are the most commonly used algorithms developed for endmember extraction. Many algorithms in the current literature, along with those discussed in PART II, are sort of their variants in one form or another. Therefore, using these two algorithms as a base for categorization many EEAs can be further categorized in Figures II.5 and II.6 where both PPI and N-FINDR can be implemented in accordance with their algorithmic implementations laid out in Chapters 7–10.

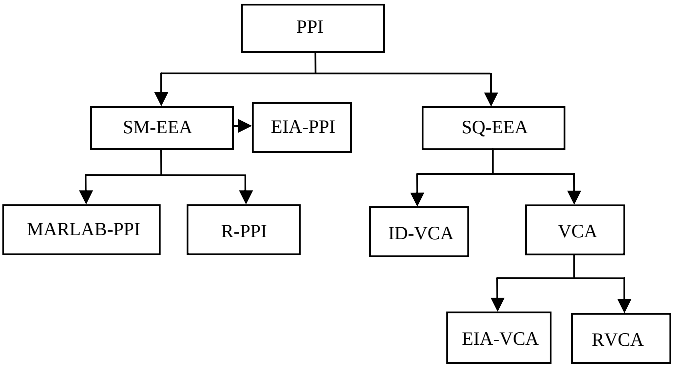


Figure II.5 Various versions of PPI.

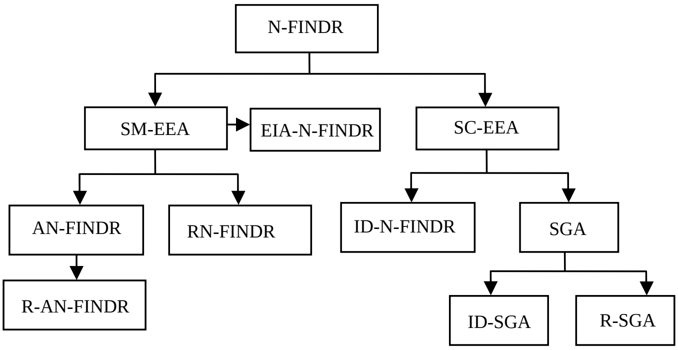


Figure II.6 Various versions of N-FINDR.

Table II.1 Categories of various EEAs

SM-EEA (random initial conditions)	SQ-EEA (random initial conditions)	ID-EEA	EIA-EEA	REEA
PPI			EIA-PPI	R-PPI
	VCA	IED-VCA	EIA-VCA	RVCA
N-FINDR			EIA-N-FINDR	R-N-FINDR
	SGA	IED-SGA		R-SGA
SPCA-EEA			EIA-SPCA-EEA	R-SPCA-EEA
	PCA	ID-PCA	ID-PCA	R-PCA
	HOS-EEA	IED-HOS-EEA	EIA-HOS-EEA	R-HOS
FCLS/IEA-EEA			EIA-FCLS/IEA-EEA	R-FCLS/IEA-EEA
	UFCLS-EEA	IED-UFCLS/IEA-EEA		R-UFCLS/IEA-EEA
	ATGP-EEA	ID-ATGP-EEA		R-ATGP-EEA

Table II.1 summarizes all the EEAs presented in Chapters 7–10 in terms of categorization of SM-EEA, SQ-EEA, ID-EEA, and REEA.

Finally, since details of implementing the original versions of PPI and N-FINDR are not available in the literature, the PPI and the N-FINDR used to carry out all the experiments are those developed in this book, referred to as MATLAB-PPI in Section 7.2.1 and iterative N-FINDR (IN-FINDR) in Section 7.2.3.2. In particular, when the term of "N-FINDR" is used without specification in this book, it is indeed the IN-FINDR, not the original version of N-FINDR developed by Winter (1999a, b). As matter of fact, the two terms of "N-FINDR" and "IN-FINDR" are used interchangeably. Specifically, when IN-FINDR is referenced, it is the version of the single-replacement IN-FINDR (1-IN-FINDR) derived in Section 7.2.3.3.1.

7

Simultaneous Endmember Extraction Algorithms (SM-EEAs)

Endmembers provide fundamental understanding of hyperspectral data where an endmember is defined as an idealized pure signature used to specify a particular spectral class. With the advent of recently developed hyperspectral imaging sensors, which utilize hundreds of contiguous spectral channels with significantly improved spatial and spectral resolutions, it is now possible to find endmembers, an important and crucial task in hyperspectral data exploitation. On many occasions endmembers appear as anomalies, rare substances, small unidentified targets, which cannot be resolved by multispectral imaging sensors but in fact provide vital information. Over the past few years, many endmember extraction algorithms (EEAs) have been developed and reported in the public domain. One of the early developments in endmember extraction is pixel purity index (PPI) developed by Boardman (1994). Since then it has emerged as one of the most widely used EEAs due to its availability in the environment for visualizing images (ENVI) commercialized by the analytical imaging and geophysics (AIG) (Research Systems Inc., 2001). In addition to PPI, many other EEAs have also been developed, for example, minimum-volume transform (MVT) (Craig, 1994), convex cone analysis (CCA) (Ifarraguerri and Chang, 1999), N-finder algorithm (N-FINDR) (Winter, 1999), automated morphological endmember extraction (AMEE) algorithm (Plaza et al., 2002), iterative error analysis (IEA) developed by Neville et al. (1999), iterated constrained endmember (ICE) (Berman et al., 2004), vertex component analysis (VCA) (Nascimento and Dias, 2005), independent component analysis-based EEA (ICA-EEA) (Wang and Chang, 2006), simplex growing algorithm (SGA) (Chang et al., 2006), and so on. Technically speaking, an EEA must extract all the desired endmembers simultaneously, referred to as simultaneous EEA (SM-EEA) in this book. However, practically speaking, finding all endmembers simultaneously is generally associated with high computational cost because it involves a brute-force search among all data sample vectors. This is particularly severe for hyperspectral data with enormous volumes. Therefore, it is highly desirable if an EEA can be carried out to find endmembers sequentially, while also producing endmembers as close as those produced by an SM-EEA in the sense of spectral similarity. An EEA implemented in such a fashion is referred to as sequential EEA (SQ-EEA) in this book. According to this categorization, PPI, N-FINDR, MVT, CCA, and AMEE are considered as SM-EEAs as opposed to VCA, ICA-EEA, and SGA, which are actually SQ-EEAs. These two types of EEAs will be discussed in detail in two separate chapters: SM-EEA in this chapter

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

followed by SQ-EEAs in Chapter 8. As will be seen in Chapter 8, SQ-EEAs are generally derived from their SM-EEA counterparts to ease computational complexity.

7.1 Introduction

What makes endmember extraction unique in hyperspectral data exploitation is that an endmember represents the purity of a spectral signature that can be used to specify a spectral class. Interestingly, endmember extraction has not received much attention in multispectral data analysis in the last decades because low spectral and spatial resolutions of multispectral imaging sensors result in collected data sample vectors, which are more likely to be mixed instead of being pure. Consequently, the likelihood of finding endmembers is rather small. Under such circumstances, there is no reason to perform endmember extraction in multispectral data other than conducting mixed data analysis. However, with the use of high spatial and spectral resolution bands many subtle material substances that cannot be resolved by multispectral sensors can be now revealed by hyperspectral imaging sensors as pure signatures. Such substances generally provide vital information in image analysis. One of such substances is endmembers that may appear as mixed sample vectors in multispectral data but turn out to be pure signature vectors in hyperspectral data. Accordingly, finding endmembers has emerged as a fundamental and critical data preprocessing that offers basic understanding of hyperspectral data. On the other hand, finding these substances is a big challenge since their existence generally cannot be known by prior knowledge or visualized by inspection from their spatial presence.

Despite the fact that many EEAs have been developed, several critical issues arising in endmember extraction have been either overlooked or not appropriately addressed. The first and foremost is the issue in determining how many endmembers are assumed to be present in the image data, denoted by p. This prior knowledge is usually not available and cannot be known *a priori*. It must be obtained from the data itself by unsupervised means. Another issue that needs to be addressed is how to find them once the value of p is determined. Over the past few years, algorithms developed for endmember extraction have focused on the second issue while avoiding the first issue by simply selecting p on an empirical basis or assuming that it is provided *a priori* by inspection or given by prior knowledge. Interestingly, the first issue of determining p in endmember extraction has not been tackled until recently a series of publications (Chaudhry et al., 2006; Chang and Plaza, 2007; Plaza and Chang, 2007; Chang et al., 2006) were reported on using a new concept called virtual dimensionality (VD), which was first introduced in Chapter 17 in Chang (2003a) as the number of spectrally distinct signatures and later published in Chang and Du (2004). Due to the fact that endmembers are always spectrally distinct signatures, VD provides a reasonable and good estimate for p even if it may not be accurate. Nevertheless, experiments show that VD is probably as close as we can get when it comes to estimation of p.

Two main streams have been developed to design an EEA. One is derived from the principle of orthogonal projection (OP). The well-known PPI (Boardman, 1994) was the first to materialize this concept to find endmembers. It assumes that endmembers are more likely to be those whose orthogonal projections on a set of randomly generated unit vectors, referred to as skewers, are either minimal or maximal. As a result, these endmembers should appear at extreme points of skewers due to convexity. In other words, when data sample vectors are orthogonally projected on a set of skewers, the desired endmembers usually have either maximal or minimal projections occurring at either end of skewers. The second mainstream for designing EEAs is to use minimal /maximal simplex volume as a criterion to find a simplex that embraces all data samples with the minimal volume such as minimal-volume transform (MVT) developed by Craig (1994) or a simplex that is embedded in the data space with the maximal volume such as N-finder algorithm (N-FINDR) developed by Winter (1999).

In addition to OP and simplex volume there is a third criterion, least-squares error (LSE), that is also derived from the notion of convexity geometry and can also be used to design EEAs. It is derived from linear spectral mixture analysis (LSMA) and assumes that a set of endmembers used for spectral unmixing produces the smallest LSE compared to the same number of data sample vectors that are used for spectral unmixing. One representative is a fully constrained least-squares-based EEA (FCLS-EEA) derived from the FCLS developed by Heinz and Chang (2001). There are also several EEAs in this same category that are very close to FCLS-EEA, for example, iterative error analysis (IEA) developed by Neville et al. (1999) and iterated constrained endmember (ICE) by Berman et al. (2004).

A fourth criterion is to use sample spectral statistics for designing EEAs, which can be categorized into second-order statistics and high-order statistics (HOS). Of particular interest in second-order statistics are second-order component analysis (CA)-based criteria, which assume that a set of endmembers produce the least possible spectral correlation among the same number of data sample vectors. One of such EEAs is referred to as standardized PCA (SPCA)-EEA developed by Ji and Chang (2006), which uses SPCA to find a set of endmembers that yield the least statistical spectral correlation among a given number of data sample vectors. This type of criterion does not satisfy the abundance constraints as FCLS does. As for HOS-based EEAs an independent component analysis (ICA)-based EEA, recently developed by Wang and Chang (2006b), represents one example in this category.

Most recently, a fifth criterion has also received interest where it uses sample spatial/spectral correlation for searching endmembers, such as automated morphological endmember extraction (AMEE) proposed by Plaza et al. (2002) and the one developed by Roggea et al. (2007).

While an EEA performs endmember extraction dimensionality reduction (DR) is required to reduce data dimensionality to cope with the problem of the so-called curse of dimensionality. Therefore, in addition to the number of endmembers, p, to be generated, another relevant issue is the number of components or dimensions to be retained after DR, denoted by q. Interestingly, the concept of VD presented in Chapter 5 has been also shown to be effective to estimate p (Chang and Plaza, 2006; Plaza and Chang, 2006) as well as q (Wang and Chang, 2006).

7.2 Convex Geometry-Based Endmember Extraction

Due to the nature of a mixed sample vector in a linear mixing model, convex geometry has been used as a major criterion behind EEA design. As a simple illustrative example, assume that $\mathbf{e}_1$ and $\mathbf{e}_2$ are two endmembers. Any data sample vector $\mathbf{x}$ lying in a line segment connecting these two endmembers as end points can be expressed as a point linearly mixed by $\mathbf{e}_1$ and $\mathbf{e}_2$ and described in the form of $\mathbf{x} = \beta\mathbf{e}_1 + (1-\beta)\mathbf{e}_2$, with $0 < \beta < 1$ based on convexity. For three endmembers $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$, a data sample vector that is linearly mixed by these three endmembers must lie in a triangle with $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ as its three vertices. Similarly, a data sample vector within a triangular pyramid/tetrahedron, square-pyramid, and p-vertex simplex is also considered to be linearly mixed by its four vertices, five vertices, and p vertices, respectively, all of which are considered as endmembers. The OP and simplex volume are two principal criteria to materialize the concept of convex geometry.

7.2.1 Convex Geometry-Based Criterion: Orthogonal Projection

PPI is a popular technique widely used in endmember extraction due to its availability in the ENVI software provided by the Research Systems. Also due to its propriety and limited publication, its

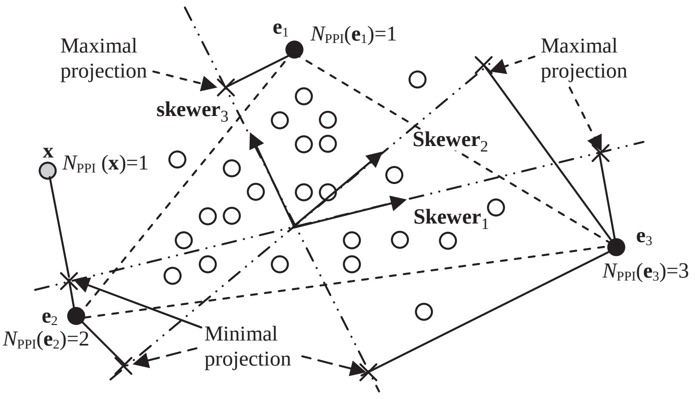


Figure 7.1 An illustration of PPI with three endmembers $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$.

detailed implementation has never been made available in the public domain. Therefore, most people who use PPI for endmember extraction either appeal for the ENVI software or implement their own versions of PPI based on whatever available in the literature. Its idea is to use OP to determine whether a data sample vector is an endmember. If a data sample is a potential endmember, its OP on a random vector should be very likely to yield either maximal or minimal projection. In order to produce such a vector, a random unit vector is generated by PPI, referred to as a skewer. Figure 7.1 shows that the three skewers. $\mathbf{skewer}_1$, $\mathbf{skewer}_2$, and $\mathbf{skewer}_3$, are randomly generated unit vectors where data sample vectors are denoted by open circles and three endmembers $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ by solid circles located at three vertices of the triangles and a maximal or a minimal projection of an endmember on a skewer is indicated by a cross “x.”

One of the major issues in PPI is sensitivity of two parameters used by PPI, which are K, the number of so-called skewers required for implementation, and τ, the cut-off threshold value for the scores, that is, counts produced by PPI, to extract candidate pixel vectors for final selection of endmembers. Another issue is the requirement of human intervention to manually select a final set of endmembers via a visualization tool provided by ENVI. Most importantly, PPI is a one-shot process, not an iterative process. Therefore, previous PPI-generated results cannot be used for next-stage new generation of endmembers if the number of skewers is increased.

This section presents our experience with PPI to offer our understanding of the ENVI’s PPI and provide our version of its implementation so that those who are interested in PPI can easily implement them to repeat our experimental results without appealing for particular software. In the mean time, we also investigate several issues resulting from its implementation. Since the MATLAB is the major software in engineering, it is highly desirable to develop a MATLAB-based PPI algorithm as described below to implement in the same way as ENVI’s PPI does.

One difference between the original PPI and the MATLAB-PPI is that the former provides no guideline to select the number of dimensions or components required to be retained after DR, while the latter uses VD to estimate the value of q for DR. In addition, the proposed MATLAB-based PPI algorithm also allows users to keep track of the PPI count for each of sample pixel vectors, which is defined as the number of skewers on which the pixel vector is orthogonally projected and falls at their ends, compared to the ENVI’s PPI that does not automatically provide PPI counts for all the sample pixel vectors (i.e., we need to do it manually). Our MATLAB-based PPI is verified by PPI in the ENVI in the sense that both produced very close results.

MATLAB PPI Algorithm

1. *Initialization*:
   i. Use VD to determine the number of dimensions, q, required to be retained after DR.
   ii. Apply a DR technique such as a maximum noise fraction (MNF) transform (Green et al., 1988; Lee et al. 1990) to reduce the dimensionality of the data set to q component images.
   iii. Randomly generate a set of K unit vectors, called "skewers," $\{\mathbf{skewer}_k\}_{k=1}^{K}$ where K is a preassumed sufficiently large positive integer.
2. *PPI count calculation*:
   Project all the data sample vectors orthogonally onto all the skewers $\{\mathbf{skewer}_k\}_{k=1}^{K}$. Then for each data sample vector $\mathbf{r}$, find those skewers on which the $\mathbf{r}$ produces either the maximal or minimal projection at their end points. These skewers form a set, called $S_{\text{extrema}}(\mathbf{r})$, for the $\mathbf{r}$. The PPI count of the $\mathbf{r}$ is defined as the number of skewers in $S_{\text{extrema}}(\mathbf{r})$, that is,

$$n_{\text{PPI}}(\mathbf{r}) = |S_{\text{extrema}}(\mathbf{r})| \tag{7.1}$$

   where $|A|$ is defined as the number of elements in a set A.
3. *Candidate selection*:
   Find an appropriate value t to threshold the PPI count $n_{\text{PPI}}(\mathbf{r})$ for all the sample vectors to select potential candidates for endmembers.
4. *Endmember extraction*:
   Extract all the sample vectors with $n_{\text{PPI}}(\mathbf{r}) \geq t$ as endmembers.

In order to illustrate the steps implemented in the MATLAB PPI, let $\{\mathbf{r}_i\}_{i=1}^{N}$ be a given set of data sample vectors. For a given value of K a random generator is used to produce a set of K random unit vectors, referred to as skewers, $\{\mathbf{skewer}_k\}_{k=1}^{K}$, which cover K different random directions. All the data sample vectors $\{\mathbf{r}_i\}_{i=1}^{N}$ are then orthogonally projected on this randomly generated skewer set, $\{\mathbf{skewer}_k\}_{k=1}^{K}$. According to geometry of convexity, an endmember that is considered as a pure signature should occur at end points of some of these skewers with either maximal or minimal projection. For each sample vector $\mathbf{r}_i$ we further calculate the number of skewers, denoted by $N_{\text{PPI}}(\mathbf{r}_i)$, at which this particular sample vector occurs as an end point to tally the PPI count for $\mathbf{r}_i$. As an example by letting $K=3$, three skewers, $\mathbf{skewer}_1$, $\mathbf{skewer}_2$, and $\mathbf{skewer}_3$, are randomly generated as shown in Figure 7.1. Assume that the data sample vectors are denoted by open circles. Then each of these data sample vectors is projected on these three skewers to see whether it yields maximal or minimal projection. For example, the data sample vector indicated by $\mathbf{e}_3$ has two maximal projections along skewers $\mathbf{skewer}_1$ and $\mathbf{skewer}_2$, and one minimal projection on skewer $\mathbf{skewer}_3$. So, in this case, its PPI count is $N_{\text{PPI}}(\mathbf{e}_3)=3$. Similarly, the data sample vector indicated by $\mathbf{e}_2$ produces minimal projections on two skewers, $\mathbf{skewer}_1$ and $\mathbf{skewer}_2$, but neither maximal nor minimal projection on $\mathbf{skewer}_3$. This gives $\mathbf{e}_2$ a PPI count, $N_{\text{PPI}}(\mathbf{e}_2)=2$. In addition, the data sample vector indicated by $\mathbf{e}_1$ produces maximal projection on $\mathbf{skewer}_3$, but neither maximal nor minimal projection on the other two skewers, $\mathbf{skewer}_1$ and $\mathbf{skewer}_2$. As a result, its PPI count, $N_{\text{PPI}}(\mathbf{e}_1)=1$. Except these three data sample vectors $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ shown by solid circles none of data sample vectors shown by open circles can produce nonzero PPI counts. Interestingly, there is a data sample vector shown by a gray circle and specified by $\mathbf{x}$ that also produces $N_{\text{PPI}}(\mathbf{e}_1)=1$. However, if we calculate the volumes formed by various triangles with their vertices selected from three out of these four data sample vectors, $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, and $\mathbf{x}$, the only one triangle specified by dashed lines and formed by the three vertices, $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ produces the maximal volume. This indicates that $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$ are the three desired endmembers.

Because of convexity, all the sample vectors in Figure 7.1 inside the triangle are supposed to have their PPI counts = 0, which implies that they are mixtures of the three endmembers at the vertices of the triangle indicated by dashed lines. It should be noted that a projection can be positive or negative depending on whether the projection occurs in the same or opposite direction of a skewer. A data sample with its maximal or minimal projection that occurred at a skewer is the one that is a potential candidate for an endmember.

Obviously, the above PPI algorithm is not an iterative process. It is performed for a given set of K skewers. Once it is done, the process is terminated. In order to ensure that our MATLAB PPI algorithm operates in the same way as the ENVI's PPI does, all the steps described above have been verified via experiments by PPI in the ENVI 3.6 version with no use of a visualization tool provided in ENVI's PPI, in which case both versions produce the same results. Finally, PPI extracts endmembers according to their PPI counts, which are thresholded by a value t that must be determined *a priori*. As a result, all sample vectors whose PPI counts are higher than t will be extracted as endmembers. In this case, many endmembers representing the same pure signature may also be extracted.

In what follows, we summarize several drawbacks resulting from the PPI in ENVI.

1. Since the MATLAB-PPI algorithm is not an iterative process, it does not guarantee that all the PPI-generated endmembers are actually true endmembers due to the fact that the K skewers are randomly generated. Different runs of implementing PPI algorithm may produce different sets of endmembers. Most importantly, for different values of K PPI must be re-run over again without taking advantage of previous results.
2. The PPI in the ENVI uses MNF transformation (or noise-adjusted principal component (NAPC)) to perform DR. Since PPI algorithm is very sensitive to noise, the noise estimation in MNF transform is crucial. In addition, when MNF transform is implemented for DR in the ENVI, no guidelines are provided to help users select the dimensions needed to be retained after DR.
3. Besides, no criteria are provided for how to select appropriate values of two parameters K and threshold t to determine the number of endmembers.
4. Finally, it requires human intervention to manually select endmembers via a visualization tool available in ENVI.

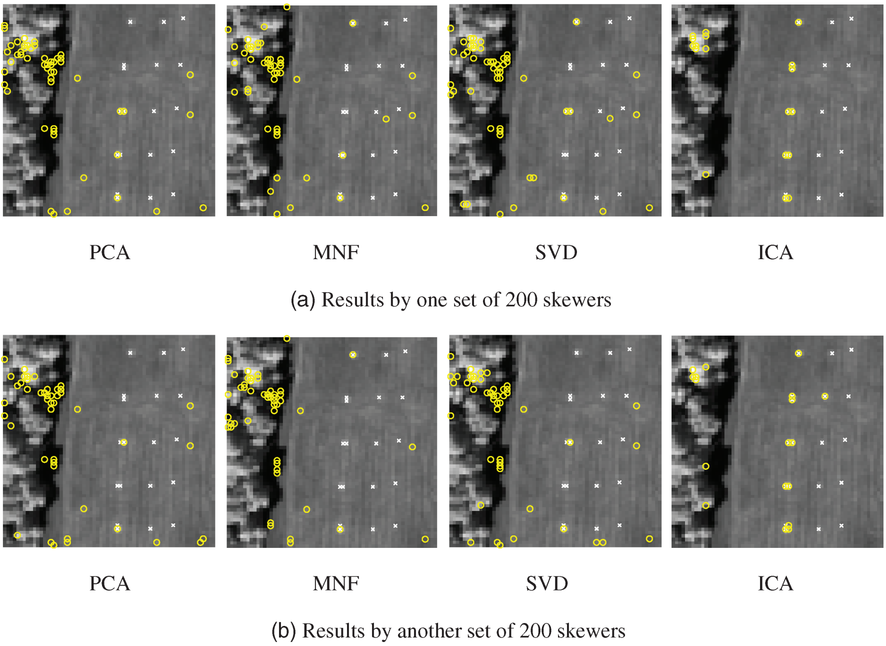
Figures 7.2 and 7.3 show two runs of HYDICE data in Figure 1.15(a) results by MATLAB-PPI using 200 and 1000 skewers, respectively, where four DR methods, PCA, MNF, SVD, and ICA, were used to perform DR.

As shown in Figures 7.2 and 7.3, only PPI using ICA to perform DR could extract panel pixels, representing five endmembers in both cases of 200 and 1000 skewers. Three interesting findings observed from Figures 7.2 and 7.3 are worthwhile.

1. *Inconsistency in finally selected endmembers resulting from the use of random initial endmembers.*

   For example, PPI using 200 skewers and SVD for DR extracted three panel pixels in one run with one set of random initial endmembers shown in Figure 7.2(a), but only two panel pixels in another run with a second set of random initial endmembers shown in Figure 7.3.
2. *Selection of an appropriate transform to perform DR, which is crucial in preserving endmember information.*

   Obviously, according to the above experiments ICA was the only effective DR transform to achieve this goal. This is because compared to PCA, MNF, and SVD, which are all second-order



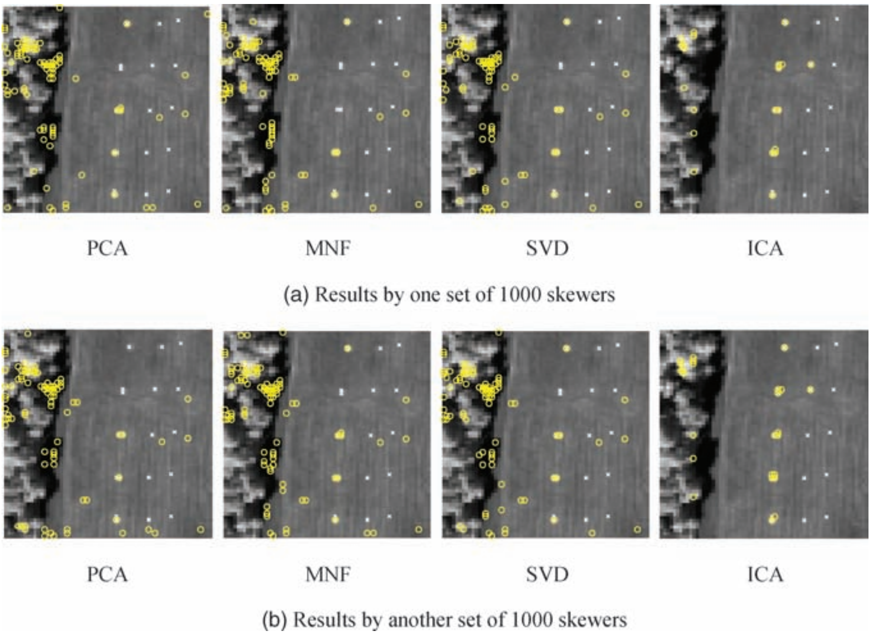
PCA MNF SVD ICA

(a) Results by one set of 200 skewers

PCA MNF SVD ICA

(b) Results by another set of 200 skewers

Figure 7.2 Panel pixels extracted by MATLAB-PPI with 200 skewers with two different runs using PCA, MNF, SVD, and ICA.



PCA MNF SVD ICA

(a) Results by one set of 1000 skewers

PCA MNF SVD ICA

(b) Results by another set of 1000 skewers

Figure 7.3 Panel pixels extracted by MATLAB-PPI with 1000 skewers with two different runs using PCA, MNF, SVD and ICA.

statistics-based transforms, ICA is a high-order statistics-based transform that can better capture characteristics of subtle substances.

3. *The impact of the number of skewers used by PPI on the performance.*
   Regardless of DR transform, PPI using 1000 skewers extracted three panel pixels in Figure 7.3 compared to only two panel pixels extracted in Figure 7.2 by PPI using 200 skewers except the case of PPI using SVD. This evidence suggested that selecting a sufficiently large number of skewers was crucial for PPI to succeed in endmember extraction.

7.2.2 Convex Geometry-Based Criterion: Minimal Simplex Volume

In addition to the OP criterion described in Section 7.2.1, another way to realize the convex geometry is to use the simplex volume as a measure to determine whether a set of data sample vectors are desired endmembers. If there are p endmembers in the data space, two criteria can be used for simplex volume measure. One is to find a set of p endmembers whose simplex volume is minimal among all possible p-vertex simplexes that embrace all data sample vectors. In other words, it begins with an initial simplex that includes all data sample vectors and then gradually deflates simplexes by replacing data sample vectors repeatedly until it reaches a simplex with minimal volume while still embracing all data sample vectors, in which case the vertices of the resulting simplex are assumed to be endmembers. This is an early approach used in MVT (Crag, 1994) and CCA (Ifarragaerri and Chang, 1999), and will be briefly described below.

7.2.2.1 Minimal-Volume Transform (MVT)

The minimal-volume transform (MVT) proposed by Craig (1994) is a nonorthogonal linear transformation that transforms the image data to a new set of coordinates with the origin set to be at so-called dark point so that the new coordinate planes can embrace the data as tightly as possible. In doing so, two linear nonorthogonal transforms, called dark-point-fixed (DPF) transform and fixed-point-free (FPF) transform, are introduced to carry out the MVT transform. The DPF transform is based on absolute coordinates centered at the origin, while the FPF transform is barycentric center at the data. The connection of MVT with linear unmixing is to find endmembers so that data samples can be circumscribed by the minimal-volume data simplex formed by the set of endmembers.

7.2.2.2 Convex Cone Analysis (CCA)

Convex cone analysis (CCA) was developed by Ifarragaerri and Chang (1999) for endmember extraction. Their idea is very similar to MVT. It also models individual component spectral signatures as vertices of a convex cone with spectra strictly non-negative. The vectors formed by the spectra that are inside the cone region will be considered to be mixed spectra. The objective of CCA is to find the boundaries of this region as defined by its vertices with minimal possible volume. The corner spectra are then the desired endmembers. In other words, CCA finds a convex cone with smallest possible volume that will embrace all data sample vectors resulting from mixtures of its vertices. It imposes a non-negativity constraint and finds the boundaries of the region that contains all positive linear combinations of the first p eigenvectors of a sample spectral correlation matrix $\mathbf{R}$ formed by $\mathbf{R} = (1/N)\mathbf{X}\mathbf{X}^T = (1/N)\sum_{i=1}^{N}\mathbf{r}_i\mathbf{r}_i^T$, where p is the number of endmembers to be generated, N is the total number of L-dimensional pixel vectors $\{\mathbf{r}_i\}_{i=1}^{N}$ in the entire image data, L is the total number of spectral bands, and $\mathbf{X}$ is a normalized data matrix formed by

$\mathbf{X} = [\mathbf{r}_1\mathbf{r}_2 \cdots \mathbf{r}_N]$. For a given set of p L-dimensional eigenvectors $\{\mathbf{v}_j\}_{j=1}^{p}$ of the matrix $\mathbf{R}$ with $\mathbf{v}_j = (v_{j1}, v_{j2}, \ldots, v_{jL})^T$, CCA solves a set of L algebraic equations

$$v_{1l} + \alpha_1 v_{2l} + \cdots + \alpha_{p-1} v_{pl} = 0 \quad \text{for} \quad l = 1, 2, \ldots, L \tag{7.2}$$

for p–1 abundance fractions $\{\alpha_j\}_{j=1}^{p-1}$ with $\alpha_j \geq 0$ for $1 \leq j \leq p - 1$. Since the exact solution for $\{\alpha_j\}_{j=1}^{p-1}$ can be only found by exhausting p–1 equations among the L algebraic equations, there are $\binom{L}{p-1}$ combinations of equations that need to be solved for $\{\alpha_j\}_{j=1}^{p-1}$. These solutions will produce linear combinations of the eigenvectors that have at least p–1 zeros. So, a boundary occurs when at least one of the vector elements in the linear combination of eigenvectors is zero while the other elements are non-negative. It should be noted that MVT and CCA share the same concept in the sense that the coordinate planes used in MVT to embrace data cloud correspond to the faces of the convex cone used in CCA. However, both MVT and CCA are different in essence. MVT finds endmembers that form a minimal-volume data simplex that circumscribes a mixing simplex formed by the same set of endmembers via linear unmixing, whereas CCA looks for corners or vertices of a convex cone that are as far away from each other as possible.

7.2.3 Convex Geometry-Based Criterion: Maximal Simplex Volume

As an alternative, a second simplex volume-based criterion is to find a simplex formed by a p- endmember simplex embedded in a data space whose volume is maximal among all possible p-vertex simplexes formed by any set of p data sample vectors. This approach can be considered as a simplex inflation process as opposed to the simplex deflation process used by MVT. It starts off with an initial simplex embedded in the data space and then keeps inflating simplexes within the data space by replacing vertices with other data sample vectors until it reaches a simplex whose volume is maximal, in which case the vertices of the found simplex are considered as endmembers. The best representative using this criterion is Winter's N-FINDR (Winter, 1999a,b). Since N-FINDR is preferred to other simplex-volume criteria in the literature, it will be described in detail in Section 7.2.4 along with its several variants.

The MVT and CCA described in Sections 7.2.2.1 and 7.2.2.2 make use of the nature in linear spectral unmixing (LSU) to find a minimal-volume data simplex to embrace all data sample vectors or data cloud via linear mixing. This section presents an idea slightly different from that used in MVT and CCA. Rather than finding a minimal-volume simplex to embrace data cloud from outside, it finds a maximal-volume simplex inside (i.e., embedded in) the data cloud so that the desired simplex can include as many data sample vectors as possible. The vertices of such found maximal-volume data simplex are designated as endmembers. In other words, a simplex with its all vertices specified by endmembers has the maximal volume compared to all simplexes embedded in the data space with same number of vertices. This is the key idea behind the N-FINDR developed by Winter (1999a,b). More specifically, within the data cloud N-FINDR finds a simplex whose volume is maximal among all simplexes with the same number of vertices. In this case, the vertices of a simplex with maximal volume are assumed to be endmembers. Unfortunately, a detailed step-by-step algorithmic implementation of N-FINDR was not provided in Winter (1999a,b). This is similar to the case of PPI where its detailed implementation is also missing. Following the same logical approach, we also develop several versions that can implement the Winter's N-FINDR, which will be referred to as simultaneous N-FINDR (SM N-FINDR).

7.2.3.1 Simultaneous N-FINDR (SM N-FINDR)

In this section, we summarize the steps to implement Winter's N-FINDR based on our understanding and experience. It by no means claims that our interpreted N-FINDR is identical to the one developed by Winter (1999a,b). Nevertheless, the idea used in both algorithms should be the same.

SM N-FINDR

1. Preprocessing.
   a. Let p be the number of endmembers to be generated.
   b. Apply a DR transform such as MNF to reduce the data dimensionality from L to p–1, where L is the total number of spectral bands.
2. Exhaustive search.
   For any p data sample vectors $\mathbf{e}_1,\ \mathbf{e}_2, \ldots,\ \mathbf{e}_p$ form a p-vertex simplex specified by $S(\mathbf{e}_1,\ \mathbf{e}_2,\ \ldots,\ \mathbf{e}_p)$ and define its volume, $V(\mathbf{e}_1,\ \mathbf{e}_2, \ldots,\ \mathbf{e}_p)$ by

$$V(\mathbf{e}_1, \ldots, \mathbf{e}_p) = \frac{\left|\det\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \mathbf{e}_1 & \mathbf{e}_2 & \ldots & \mathbf{e}_p \end{bmatrix}\right|}{(p-1)!}. \tag{7.3}$$

   Find a set of p data sample vectors, denoted by $\{\mathbf{e}_1^*, \mathbf{e}_2^*, \ldots, \mathbf{e}_p^*\}$, that yields the maximal value of (7.3), that is,

$$\{\mathbf{e}_1^*, \mathbf{e}_2^*, \ldots, \mathbf{e}_p^*\} = \arg\{\max_{\{\mathbf{e}_1,\mathbf{e}_2,\ldots,\mathbf{e}_p\}} V(\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p)\} \tag{7.4}$$

3. Let $\{\mathbf{e}_1^*, \mathbf{e}_2^*, \ldots, \mathbf{e}_p^*\}$ be the desired set of endmembers.

Figure 7.4 depicts a flow chart of implementing the SM N-FINDR.

7.2.3.2 Iterative N-FINDR (IN-FINDR)

According to step 2 described in the above SM N-FINDR an exhaustive search must be conducted via (7.4) by finding an optimal set of endmembers, $\{\mathbf{e}_1^*, \mathbf{e}_2^*, \ldots, \mathbf{e}_p^*\}$ that requires tremendous computing time to accomplish this process. One way to mitigate this dilemma is to break up the exhaustive search process into two processes, inner loop and outer loop, each of which can be implemented iteratively. The resulting N-FINDR is called iterative N-FINDR (IN-FINDR) and can be described as follows.

Iterative N-FINDR (IN-FINDR)

1. Preprocessing.
   a. Let p be the number of endmembers to be generated.
   b. Apply a DR transform such as MNF to reduce the data dimensionality from L to p–1, where L is the total number of spectral bands.
2. Initialization.
   Let $\{\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}\}$ be a set of initial vectors randomly selected from the data. $k = 0$.

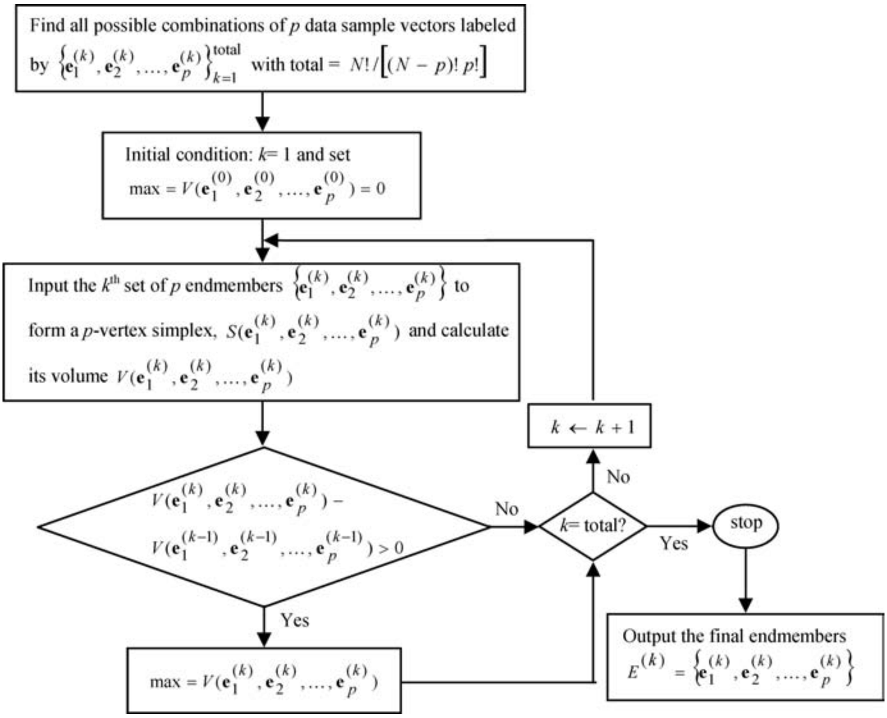

Find all possible combinations of p data sample vectors labeled by $\{\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}\}_{k=1}^{\text{total}}$ with total = $N!/[(N-p)!\,p!]$
Initial condition: k= 1 and set max = $V(\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}) = 0$
Input the k^{th} set of p endmembers $\{\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}\}$ to form a p-vertex simplex, $S(\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ and calculate its volume $V(\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)})$
$V(\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}) - V(\mathbf{e}_1^{(k-1)}, \mathbf{e}_2^{(k-1)}, \ldots, \mathbf{e}_p^{(k-1)}) > 0$
No
Yes
max = $V(\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)})$
k= total?
No
$k \leftarrow k+1$
Yes
stop
Output the final endmembers $E^{(k)} = \{\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}\}$

Figure 7.4 A flow chart of SM N-FINDR implementation.

3. Outer Loop.

   At iteration $k \geq 0$, find the volume of the simplex specified by the p vertices $\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}$, $V(\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ defined by (7.3).
4. Inner loop.

   For $1 \leq j \leq p$, we recalculate $V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j-1}^{(k)}, \mathbf{r}, \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ for all data sample vectors $\mathbf{r}$. If none of these p recalculated volumes, $V(\mathbf{r}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)})$, $V(\mathbf{e}_1^{(k)}, \mathbf{r}, \mathbf{e}_3^{(k)}, \ldots, \mathbf{e}_p^{(k)})$,..., $V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{p-1}^{(k)}, \mathbf{r})$, is greater than $V(\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)})$, no endmember in $\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}$ will be replaced. The algorithm is terminated. Otherwise, continue.
5. Replacement rule.

   The endmember that is absent in the largest volume among the p simplexes, $S(\mathbf{r}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)})$, $S(\mathbf{e}_1^{(k)}, \mathbf{r}, \mathbf{e}_3^{(k)}, \ldots, \mathbf{e}_p^{(k)})$, ... , $S(\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_{p-1}^{(k)}, \mathbf{r})$, will be replaced by the sample vector $\mathbf{r}$. Assume that such an endmember is denoted by $\mathbf{e}_j^{(k+1)}$. A new set of endmembers is then produced by letting $\mathbf{e}_j^{(k+1)} = \mathbf{r}$ and $\mathbf{e}_i^{(k+1)} = \mathbf{e}_i^{(k)}$ for $i \neq j$. Let $k \leftarrow k+1$ and go to step 3.

Figure 7.5 describes a flow chart of implementing IN-FINDR step by step in two loops, inner and outer loops.

Interestingly, a recent version of N-FINDR developed in Winter (2004) is actually implemented in two loops in a reverse order of the two loops carried out by the IN-FINDR.

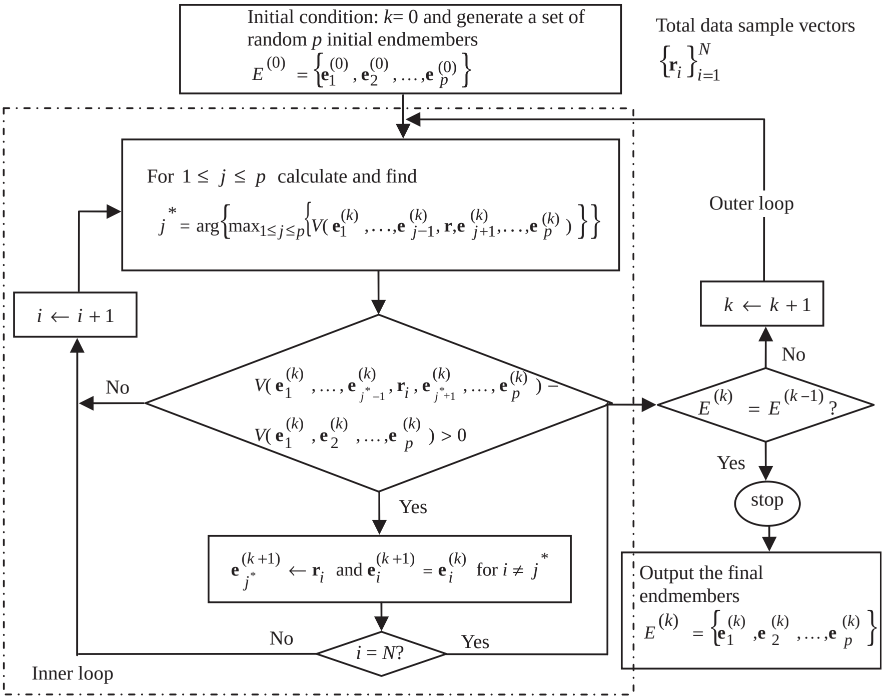

Initial condition: k= 0 and generate a set of random p initial endmembers
$E^{(0)} = \{\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}\}$
Total data sample vectors
$\{\mathbf{r}_i\}_{i=1}^N$
For $1 \leq j \leq p$ calculate and find
$j^* = \arg\{\max_{1 \leq j \leq p}\{V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j-1}^{(k)}, \mathbf{r}, \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)})\}\}$
Outer loop
$i \leftarrow i+1$
$k \leftarrow k+1$
No
No
$V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j^*-1}^{(k)}, \mathbf{r}_i, \mathbf{e}_{j^*+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}) -$
$V(\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}) > 0$
$E^{(k)} = E^{(k-1)}$?
Yes
stop
Yes
$\mathbf{e}_{j^*}^{(k+1)} \leftarrow \mathbf{r}_i$ and $\mathbf{e}_i^{(k+1)} = \mathbf{e}_i^{(k)}$ for $i \neq j^*$
Output the final endmembers
$E^{(k)} = \{\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}\}$
No
$i = N$?
Yes
Inner loop

Figure 7.5 A flow chart of IN-FINDR implementation.

It should be noted that the IN-FINDR developed in this section is generally not as optimal as SM N-FINDR. However, it can be considered as nearly optimal. The similarity between SM N-FINDR and IN-FINDR can be illustrated by the similarity between finding double integral and iterated integrals where the region to be integrated is generally a 2D dimensional space compared to the regions to be calculated by iterated integrals, which are usually one-dimensional space. Using this context as interpretation, the exhaustive search implemented in SM N-FINDR is similar to finding a multiple integral that integrates a multidimensional region in the original data space, whereas the inner and outer loops carried out in IN-FINDR iteratively are similar to finding iterated integrals over one-dimensional space. In most cases, the results obtained by iterated integrals are the same as those obtained by directly finding multiple integrals according to Fubini's theorem (Reed and Simon, 1972).

7.2.3.3 Various Versions of Implementing IN-FINDR

The replacement rule described in steps 4 and 5 of IN-FINDR is a general concept. For practical implementation the replacement rule can be designed to ease computation depending on how endmembers are replaced. There are several versions that can be designed to implement the replacement rule used by IN-FINDR. In this section, three variants of IN-FINDR are developed: single-replacement N-FINDR (1-IN-FINDR), multiple-replacement IN-FINDR, and successive IN-FINDR (SC IN-FINDR).

Single-Replacement IN-FINDR

The IN-FINDR presented in this section uses a simple replacement rule that replaces one endmember at a time. The resulting IN-FINDR is referred to as 1-IN-FINDR and the IN-FINDR described in Section 7.2.4.2 can be further specified as follows.

Single-Replacement IN-FINDR (1-IN-FINDR)

1. Preprocessing.
   a. Let p be the number of endmembers to be generated.
   b. Apply a DR transformation to reduce the data dimensionality from L to p–1, where L is the total number of spectral bands.
2. Initialization.

   Let $\{\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}\}$ be a set of initial vectors randomly generated from the data. Let $k = 0$
3. Form a simplex $S(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_j^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ with $\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_j^{(k)}, \ldots, \mathbf{e}_p^{(k)}$ as the p vertices and define $V(\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ as its volume defined by (7.3).
4. For $1 \le j \le p$ find $\mathbf{e}_j^{(k,*)}$ that yields the maximal volume of $V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j-1}^{(k)}, \mathbf{r}, \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ over all sample vectors $\mathbf{r}$, that is,

   $$\mathbf{e}_j^{(k,*)} = \max_{\mathbf{r}} V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j-1}^{(k)}, \mathbf{r}, \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}), \tag{7.5}$$

   while fixing other endmembers $\mathbf{e}_l^{(k)}$ with $l \neq j$.
5. Check whether

   $$\mathbf{e}_j^{(k)} = \mathbf{e}_j^{(k,*)} \text{ for all} js \text{with } 1 \le j \le p. \tag{7.6}$$

   If yes, the algorithm is terminated. Otherwise, let $\mathbf{e}_j^{(k+1)} = \mathbf{e}_j^{(k,*)}$. Set $k \leftarrow k+1$ and go to step 3.

Multiple-Replacement IN-FINDR

In the 1-IN-FINDR described above, the replacement carried out in step 5 is done by one single endmember at a time for each iteration, which is the endmember $\mathbf{e}_i^{(k,*)}$ found in step 4. It is a suboptimal algorithm to perform endmember replacement in step 5 to save computational complexity. A possible extension of 1-IN-FINDR is to replace two endmembers at a time rather than single replacement implemented in 1-IN-FINDR. The following algorithm is designed for this purpose. It is a two-replacement IN-FINDR (2-IN-FINDR) where the single endmember replacement process carried out in steps 4 and 5 in 1-IN-FINDR is replaced by two endmembers at a time for each iteration described in the following steps 4 and 5.

Two-Replacement IN-FINDR (2-IN-FINDR)

4. For each pair (i,j) with $1 \le i < j \le p$, find a pair of $(\mathbf{e}_i^{(k,*)}, \mathbf{e}_j^{(k,*)})$ that yields the maximum volume of $V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{i-1}^{(k)}, \mathbf{r}, \mathbf{e}_{i+1}^{(k)}, \cdots, \mathbf{e}_{j-1}^{(k)}, \mathbf{r}', \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ over all sample vectors $\mathbf{r}$, that is,

$$(\mathbf{e}_i^{(k,*)}, \mathbf{e}_j^{(k,*)}) = \max_{(\mathbf{r},\mathbf{r}')} V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{i-1}^{(k)}, \mathbf{r}, \mathbf{e}_{i+1}^{(k)}, \ldots, \mathbf{e}_{j-1}^{(k)}, \mathbf{r}', \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}), \tag{7.7}$$

while fixing other endmembers $\mathbf{e}_l^{(k)}$ with $l \neq i,j$. It generally requires an extensive search for an optimal pair $(\mathbf{e}_i^{(k,*)}, \mathbf{e}_j^{(k,*)})$ to satisfy (7.7), which is very time-consuming. One way to alleviate such a searching process is to solve two endmembers successively as follows:

$$(\mathbf{e}_i^{(k,*)}, \mathbf{e}_j^{(k,*)}) = \max_{\mathbf{r}'}\left\{\max_{\mathbf{r}}\left[V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{i-1}^{(k)}, \mathbf{r}, \mathbf{e}_{i+1}^{(k)}, \ldots, \mathbf{e}_{j-1}^{(k)}, \mathbf{r}', \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)})\right]\right\}, \tag{7.8}$$

where the $(\mathbf{e}_i^{(k,*)}, \mathbf{e}_j^{(k,*)})$ obtained by (7.8) may not be the same pair of $(\mathbf{e}_i^{(k,*)}, \mathbf{e}_j^{(k,*)})$ obtained from (7.5), but their volumes are generally very close.

5. Check whether

$$\mathbf{e}_l^{(k)} = \mathbf{e}_l^{(k,*)} \quad \text{for all } ls \text{ with } 1 \le l \le p. \tag{7.6}$$

If yes, the algorithm is terminated. Otherwise, let $\mathbf{e}_i^{(k+1)} = \mathbf{e}_i^{(k,*)}$ and $\mathbf{e}_j^{(k+1)} = \mathbf{e}_j^{(k,*)}$. Then set $k \leftarrow k+1$ and go to step 3.

Once a 2-IN-FINDR is developed, it can be easily extended to multiple-replacement IN-FINDR where more than two endmembers can be replaced in steps 4 and 5 in 2-IN-FINDR. More specifically, for any given integer s with $2 < s \le p$ the multiple-replacement IN-FINDR can replace s endmembers simultaneously to generate its new endmembers for next iteration, referred to as s-IN-FINDR described below.

s-Replacement IN-FINDR (s-IN-FINDR)

4. For any given set of s components $(j_1, \ldots, j_s)$ with $1 \le j_1 < \cdots < j_s \le p$ find a set of s components of $(\mathbf{e}_{j_1}^{(k,*)}, \ldots, \mathbf{e}_{j_s}^{(k,*)})$ that yields the maximal volume of $V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j_1-1}^{(k)}, \mathbf{r}_1, \mathbf{e}_{j_1+1}^{(k)}, \ldots, \mathbf{e}_{j_2-1}^{(k)}, \mathbf{r}_2, \mathbf{e}_{j_2+1}^{(k)}, \ldots, \mathbf{e}_{j_s-1}^{(k)}, \mathbf{r}_s, \mathbf{e}_{j_s+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ over all s vector sample vectors $\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_s$, that is,

$$\begin{aligned}&(\mathbf{e}_{j_1}^{(k,*)}, \ldots, \mathbf{e}_{j_s}^{(k,*)}) \\ &\quad = \max_{\mathbf{r}_1,\ldots,\mathbf{r}_s} V\left(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j_1-1}^{(k)}, \mathbf{r}_1, \mathbf{e}_{j_1+1}^{(k)}, \ldots, \mathbf{e}_{j_2-1}^{(k)}, \mathbf{r}_2, \mathbf{e}_{j_2+1}^{(k)}, \ldots, \mathbf{e}_{j_s-1}^{(k)}, \mathbf{r}_s, \mathbf{e}_{j_s+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}\right)\end{aligned} \tag{7.9}$$

while fixing other endmembers $\mathbf{e}_l^{(k)}$ with $l \notin \{j_1, \ldots, j_s\}$. It generally requires an extensive search for an optimal pair $(\mathbf{e}_{j_1}^{(k,*)}, \ldots, \mathbf{e}_{j_s}^{(k,*)})$ to satisfy (7.6), which is very time-consuming. One way to alleviate such a searching process is to solve two endmembers successively as follows:

$$\begin{aligned}&(\mathbf{e}_{j_1}^{(k,*)}, \ldots, \mathbf{e}_{j_s}^{(k,*)}) \\ &\quad = \max_{\mathbf{r}_s}\left\{\max_{\mathbf{r}_{s-1}}\left\{\cdots\left\{\max_{\mathbf{r}_1}\left[V\left(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j_1-1}^{(k)}, \mathbf{r}_1, \mathbf{e}_{j_1+1}^{(k)}, \ldots, \mathbf{e}_{j_s-1}^{(k)}, \mathbf{r}_s, \mathbf{e}_{j_s+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}\right)\right]\right\}\right\}\right\},\end{aligned} \tag{7.10}$$

where $(\mathbf{e}_{j_1}^{(k,*)}, \ldots, \mathbf{e}_{j_s}^{(k,*)})$ obtained by (7.10) may not be the same set of $(\mathbf{e}_{j_1}^{(k,*)}, \ldots, \mathbf{e}_{j_s}^{(k,*)})$ obtained from (7.7), but their volumes are generally very close.

5. Check whether

$$\mathbf{e}_l^{(k)} = \mathbf{e}_l^{(k,*)} \quad \text{for all } l\text{'s with } \quad 1 \le l \le p. \tag{7.6}$$

If yes, the algorithm is terminated. Otherwise, let

$$\mathbf{e}_{j_1}^{(k+1)} = \mathbf{e}_{j_1}^{(k,*)}, \ldots, \mathbf{e}_{j_s}^{(k+1)} = \mathbf{e}_{j_s}^{(k,*)}. \tag{7.11}$$

Then set $k \leftarrow k+1$ and go to step 3.

It should be noted that as s is increased, the computational complexity of s-IN-FINDR also grows exponentially until it reaches p, that is, when $s = p$ the s-IN-FINDR becomes the original version of IN-FINDR that replaces p endmembers simultaneously specified by (7.10).

Successive IN-FINDR (SC IN-FINDR)

As noted in the s-IN-FINDR, a major issue in its implementation is computational cost, which is expected to be very high. In order to mitigate this problem, an alternative suboptimal version of

s-IN-FINDR can be derived as follows by replacing the simultaneous replacement of s endmembers in step 5 implemented in the s-IN-FINDR with a successive s-endmember replacement that is analogous to (7.10) but more efficient at the expense of optimality.

s-Successive replacement N-FINDR (s-SC IN-FINDR)

1. Preprocessing.
   a. Let p be the number of endmembers to be generated.
   b. Apply a DR transformation to reduce the data dimensionality from L to $p-1$, where L is the total number of spectral bands.
2. Initialization.
   Let $\{\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}\}$ be a set of initial vectors randomly generated from the data. Set $k = 0$
3. Form a simplex $S(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_j^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ with $\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_j^{(k)}, \ldots, \mathbf{e}_p^{(k)}$ as its p vertices with its volume $V(\mathbf{e}_1^{(k)}, \mathbf{e}_{\mathbf{2}}^{(\mathbf{k})}, \ldots, \mathbf{e}_p^{(k)})$ defined by (7.3).
4. For any given set of s components $(j_1, \ldots, j_s)$ with $1 \le j_1 < \cdots < j_s \le p$ find $\mathbf{e}_{j_l}^{(k,*)}$ that yields the maximal volume of $V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j_{l-1}-1}^{(k)}, \mathbf{e}_{j_{l-1}}^{(k)}, \mathbf{e}_{j_{l-1}+1}^{(k)}, \ldots, \mathbf{e}_{j_l-1}^{(k)}, \mathbf{r}_{j_l}, \mathbf{e}_{j_l+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ over all sample vectors $\mathbf{r}_{j_l}$, while fixing other endmembers $\mathbf{e}_{j_i}^{(k,*)}$ with $i < l$ and $\mathbf{e}_i^{(k)}$ with $i \notin \{j_1, j_2, \ldots, j_{l-1}\}$. That is,

$$\mathbf{e}_{j_l}^{(k,*)} = \max\nolimits_{\mathbf{r}} V\Big(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j_{l-1}-1}^{(k)}, \mathbf{e}_{j_{l-1}}^{(k,*)}, \mathbf{e}_{j_{l-1}+1}^{(k)}, \ldots, \mathbf{e}_{j_l-1}^{(k)}, \mathbf{r}_{j_l}, \mathbf{e}_{j_l+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}\Big). \tag{7.12}$$

5. Check whether the stopping rule specified by (7.6) is satisfied. If yes, the algorithm is terminated. Otherwise, let

$$\mathbf{e}_{j_l}^{(k+1)} = \mathbf{e}_{j_l}^{(k,*)}. \tag{7.13}$$

   Set $k \leftarrow k+1$ and go to step 3.

   In particular, if $s = p$, steps 4 and 5 in the above s-SC IN-FINDR can be further modified as follows

p-Successive replacement N-FINDR (p-SC IN-FINDR)

4. For $1 \le j \le p$ find $\mathbf{e}_j^{(k,*)}$ that yields the maximum volume of $V(\mathbf{e}_1^{(k,*)}, \ldots, \mathbf{e}_{j-1}^{(k,*)}, \mathbf{r}, \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ over all sample vectors $\mathbf{r}$, while fixing other endmembers $\mathbf{e}_i^{(k,*)}$ with $i < j$ and $\mathbf{e}_i^{(k)}$ with $i > j$.
   That is,

$$\mathbf{e}_j^{(k,*)} = \max\nolimits_{\mathbf{r}} V\Big(\mathbf{e}_1^{(k,*)}, \ \ldots, \ \mathbf{e}_{j-1}^{(k,*)}, \mathbf{r}, \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}\Big). \tag{7.14}$$

5. Check whether the stopping rule specified by (7.6) is satisfied. If yes, the algorithm is terminated. Otherwise, use (7.13) and set $k \leftarrow k+1$. Then go to step 3.

The major difference between s-IN-FINDR and s-SC IN-FINDR is the equation used to generate a new endmember that yields the maximal simplex volume, (7.8) in s-IN-FINDR and (7.12) in s-SC IN-FINDR. More specifically, in (7.8) the endmembers other than $\mathbf{e}_j^{(k)}$ are fixed at the original endmembers $\mathbf{e}_i^{(k)}$ for $i \notin \{j_1, \ldots, j_s\}$. However, in (7.12), only the endmembers in (7.7) after $\mathbf{e}_{j_l}^{(k)}$ are fixed at $\mathbf{e}_i^{(k)}$ for $i > j_l$ or $i \notin \{j_1, \ldots, j_s\}$, while those endmembers $\mathbf{e}_i^{(k)}$ with

$i \in \{j_1, \ldots, j_{l-1}\}$ are updated and replaced by setting $\mathbf{e}_i^{(k)} = \mathbf{e}_i^{(k,*)}$ for $i \in \{j_1, \ldots, j_{l-1}\}$. As a result, with $s = p$ an EEA, which implements step 4 without step 5 in the version of p-SC IN-FINDR, can be considered as a SQ-EEA which will be discussed in Chapter 8 in detail. Nevertheless, since both versions implement step 5 to avoid a possible trap in local optima, they both eventually produce close results except that p-SC IN-FINDR may converge faster than p-IN-FINDR.

7.2.3.4 Discussions on Various Implementation Versions of IN-FINDR

The main reason for developing various approaches to implementing IN-FINDR is the following. Since SM N-FINDR must conduct an exhaustive search among all data sample vectors to ensure that an optimal set of endmembers is produced at the end, even for a moderate size of hyperspectral data, this approach is nearly impossible to accomplish within limited computing resources. This is an inevitable issue encountered in most of SM-EEAs, specifically convex hull-based EEAs such as MVT and CCA. As an example, for N-FINDR to be an optimal EEA, it must find all p endmembers simultaneously among all possible combinations, $\begin{pmatrix} N \\ P \end{pmatrix} = \frac{N!}{(N-p)!p!}$ of p endmembers with N being the total number of data sample vectors. As N grows very large, its computational complexity becomes increasingly formidable in reality. One way to resolve this dilemma is to limit its search process to a feasible region in which the desired endmembers are most likely to occur. In doing so, two loops are designed to replace the exhaustive search required by N-FINDR, which results in IN-FINDR. This can be seen in the two loops designed for s-IN-FINDR and s-SC IN-FINDR, both of which use inner loop indexed by j_l in step 4 and outer loop indexed by k in step 5 to accomplish what the SM N-FINDR does. The inner loop runs $1 \leq l \leq s$ by replacing the (j_l)th endmember $\mathbf{e}_{j_l}^{(k)}$ with the most probable endmember $\mathbf{e}_{j_1}^{(k,*)}$ that yields the maximal volume of $V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j_{l-1}-1}^{(k)}, \mathbf{e}_{j_{l-1}}^{(k)}, \mathbf{e}_{j_{l-1}+1}^{(k)}, \ldots, \mathbf{e}_{j_l-1}^{(k)}, \mathbf{r}_{j_l}, \mathbf{e}_{j_l+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ in s-IN-FINDR via (7.10) or $V(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j_{l-1}-1}^{(k)}, \mathbf{e}_{j_{l-1}}^{(k,*)}, \mathbf{e}_{j_{l-1}+1}^{(k)}, \ldots, \mathbf{e}_{j_l-1}^{(k)}, \mathbf{r}_{j_l}, \mathbf{e}_{j_l+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)})$ in s-SC IN-FINDR via (7.12). When $l = s$, s-IN-FINDR and s-SC IN-FINDR check whether the stopping rule described by (7.6) is satisfied. If it is, then s-IN-FINDR and s-SC IN-FINDR are terminated. Otherwise, they go to the outer loop indexed by k by implementing a replacement rule described by (7.6) for s-IN-FINDR or (7.13) for s-SC IN-FINDR and setting $k \leftarrow k + 1$ to go back to the inner loop again in step 4. According to our extensive experimental studies it seems that the p-SC IN-FINDR is the best in the sense that the outer loop is designed to accomplish the same task as IN-FINDR does by repeatedly replacing a set of p endmembers found by the inner loop in a successive manner until it obtains the maximal simplex volume.

Interestingly, as discussed in Chapter 8, when only the inner loop of s-SC IN-FINDR is implemented without iterating the outer loop, s-SC IN-FINDR becomes a successive EEA, specifically $s = p$, s-SC IN-FINDR is reduced to SC IN-FINDR discussed in Section 7.2.3.3, while with $s = 1$ s-SC IN-FINDR is reduced to SC N-FINDR in Section 8.2.

7.2.3.5 Comparative Study Among Various Versions of IN-FINDR

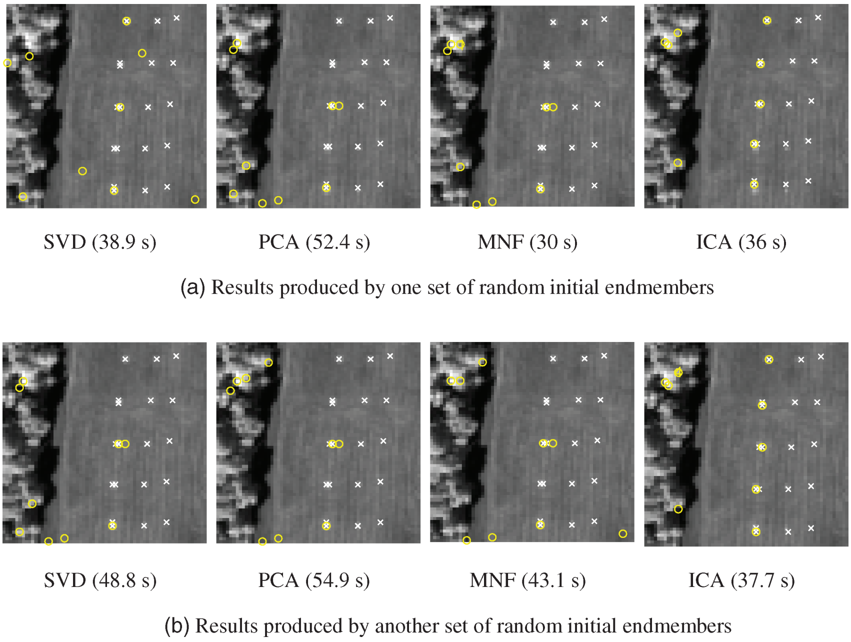
In order to see how each of various versions of IN-FINDR works in terms of its performance and computation, it is worthwhile conducting experiments using the same computing environment. The data to be used for illustration is the HYDICE image in Figure 1.15. Since IN-FINDR requires a

DR, four DR techniques, singular value decomposition (SVD), principal components analysis (PCA), MNF, and independent component analysis (ICA) discussed in Chapter 6, are used for experiments. Additionally, according to the results in Chapter 5, the effective dimensionality to be retained after DR is estimated by VD to be 9. Therefore, $q = 9$ is used for our experiments. Finally, due to the use of random initial endmembers by IN-FINDR the experiments are also conducted using two different sets of random initial conditions to demonstrate inconsistency in final selected endmembers as shown in Figures 7.6–7.8, which present the nine endmember extraction results of two runs from two different sets of random initial endmembers along with their computing times in parentheses.

As demonstrated in Figures 7.6–7.8, 9-SC IN-FINDR required the least time of 11.4 s on average as expected, while 1-IN-FINDR had the worst time of about 42.5 s on average, which was roughly four times in computation compared to that needed by 9-SC IN-FINDR. It is also demonstrated that all the three versions of IN-FINDR using ICA to perform DR could extract all the five endmembers regardless of what random initial endmembers were used for initialization. In addition, inconsistent results were clearly shown by two different runs. It is particularly evidential for the three versions of IN-FINDR with SVD used to reduce dimensionality where two endmembers were extracted in one run while three endmembers were extracted in another run.

7.2.3.6 Alternative SM N-FINDR

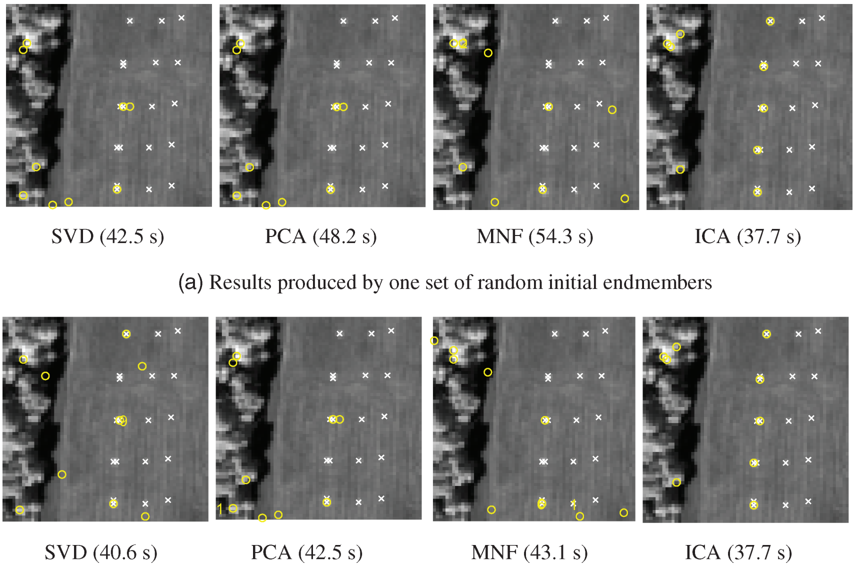
Also interestingly, SM N-FINDR can be modified by introducing a scale parameter δ that controls effectiveness of the abundance sum-to-one constraint (ASC) imposed on selected endmembers. The resulting SM N-FINDR is called alternative SM N-FINDR (ASM N-FINDR) and its idea is summarized in the following algorithmic implementation.



(a) Results produced by one set of random initial endmembers

(b) Results produced by another set of random initial endmembers

Figure 7.6 1-IN-FINDR.



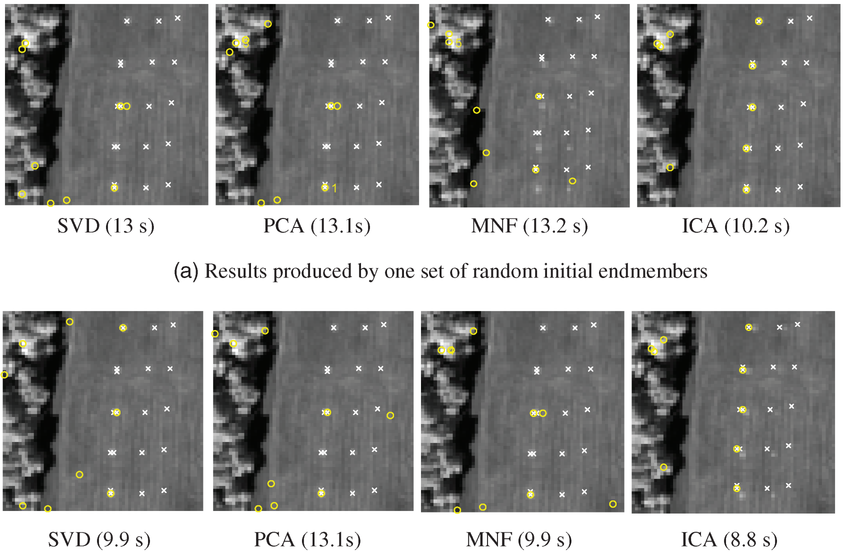
SVD (42.5 s) PCA (48.2 s) MNF (54.3 s) ICA (37.7 s)

(a) Results produced by one set of random initial endmembers

SVD (40.6 s) PCA (42.5 s) MNF (43.1 s) ICA (37.7 s)

(b) Results produced by another set of random initial endmembers

Figure 7.7 2-IN-FINDR.



SVD (13 s) PCA (13.1s) MNF (13.2 s) ICA (10.2 s)

(a) Results produced by one set of random initial endmembers

SVD (9.9 s) PCA (13.1s) MNF (9.9 s) ICA (8.8 s)

(b) Results produced by another set of random initial endmembers

Figure 7.8 9-SC IN-FINDR.

ASM N-FINDR Algorithm

1. Set p to the number of endmembers to be generated; L is the number of bands.
   Let $\{\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}\}$ be set of initial endmembers.
2. Let $\{\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}\}$ be the set of p endmembers generated at iteration $k \geq 0$ and $\mathbf{E}_{L\times p}^{(k)} = \left[\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}\right]$ be an $L \times p$ kth p-endmember matrix. We define an $(L+1)\times p$ matrix $\bar{\mathbf{E}}_{(L+1)\times p}^{(k)}(\delta)$ by augmenting the matrix $\mathbf{E}_{L\times p}^{(k)}$ using $1_p = \underbrace{(1, 1, \ldots, 1)^T}_{p}$, which is a p-dimensional unity vector as follows:

$$\bar{\mathbf{E}}_{(L+1)\times p}^{(k)}(\delta) = \begin{bmatrix} \delta \mathbf{E}_{L\times p}^{(k)} \\ \mathbf{1}_p^T \end{bmatrix} = \begin{bmatrix} \delta \mathbf{e}_1^{(k)} & \delta \mathbf{e}_2^{(k)} & \cdots & \delta \mathbf{e}_{p-1}^{(k)} & \delta \mathbf{e}_p^{(k)} \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}, \tag{7.15}$$

where δ is a scalar parameter that is used to control effectiveness of the ASC. Generally, δ can be chosen to be $\delta = (10 \times \max_{r_{il}}\{r_{il}\})^{-1}$, where $\mathbf{r}_i = (r_{i1}, r_{i2}, \ldots, r_{iL})^T$ for $1 \leq i \leq N$ and N is the total number of image pixels in the image. Now we can form a $p \times p$ matrix by

$$\left|\mathbf{R}_{p\times p}^{(k)}(\delta)\right|^{1/2} = \left|\left(\bar{\mathbf{E}}_{L\times p}^{(k)}(\delta)\right)^T \left(\bar{\mathbf{E}}_{L\times p}^{(k)}(\delta)\right)\right|^{1/2} = \left|\bar{\mathbf{E}}_{L\times p}^{(k)}(\delta)\right|, \tag{7.16}$$

where $\left|\left(\bar{\mathbf{E}}_{L\times p}^{(k)}(\delta)\right)^T\right| = \left|\bar{\mathbf{E}}_{L\times p}^{(k)}(\delta)\right|$ and $|\mathbf{R}_{p\times p}^{(k)}(1)|^{1/2} = V(\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)})(p-1)!$ from (7.3).

The criterion used by ASM N-FINDR is to iterate the p endmembers and make the determinant, $|\mathbf{R}_{p\times p}^{(k)}|$, as large as possible where the iterative process can be carried out in the same manner as described in steps 3–5 in SM N-FINDR with (7.4) being replaced by (7.16).

7.2.4 Convex Geometry-Based Criterion: Linear Spectral Mixture Analysis

Linear spectral mixture analysis (LSMA) has been widely used in remote sensing image classification where the least-squares error (LSE) is considered as an optimal criterion for LSMA. It is highly desirable if it can be also used as a criterion for endmember extraction. Two LSMA-based techniques are of interest and take completely opposite approaches. Assume that there are p endmembers present in the data. One was proposed by Berman et al. (2004), referred to as iterated constrained endmember (ICE) to minimize the unmixed error resulting from a p-vertex simplex while constraining the size of a simplex via the sum of Euclidean distance among all p selected endmembers instead of the simplex volume used by MVT and N-FINDR. The other approach developed in this section is based on an underlying assumption that the unmixed error using all p endmembers via LSMA is always greater than that caused by using any set of p signatures for LSMA regardless of whether or not these signatures are pure or mixed. Such resulting unmixed LSE is referred to as maximal linear spectral unmixed error (MLSUE). Furthermore, two abundance constraints, sum-to-one and non-negativity, must be also imposed on LSMA to satisfy the constraint of convexity. This leads to a natural choice proposed in Ji (2005), referred to as FCLS-EEA, which is derived from the fully constrained least-squares (FCLS) developed in Heinz and Chang (2001). FCLS-EEA is also known as unsupervised fully constrained least squares (UFCLS) in Chang (2003a). It first starts with p randomly select endmembers and then calculates error vectors from the difference between the endmembers and the linear mixture model mixed by other endmembers with abundance estimated by the FCLS. The error vectors are used as projection vectors to find a new endmember that has the maximal length on the projections. The details of the algorithm are described as follows.

FCLS-EEA Algorithm

1. Initialization.
   Let p be the number of endmembers to generate and $\{e_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}\}$ be a set of randomly generated initial endmembers.
2. At iteration $k \geq 0$, for each $1 \leq j \leq p$ we assume that $\mathbf{e}_j^{(k)}$ is a mixed pixel and can be represented by $\hat{\mathbf{e}}_j^{(k)}$ that is generated by the remaining $p{-}1$ endmembers, $\{\mathbf{e}_i^{(k)}\}_{i=1,j}^{p}$ using the FCLS, that is, $\hat{\mathbf{e}}_j^{(k)} = \alpha_1^{(k)}\mathbf{e}_1^{(k)} + \alpha_2^{(k)}\mathbf{e}_2^{(k)} + \cdots + \alpha_{j-1}^{(k)}\mathbf{e}_{j-1}^{(k)} + \alpha_{j+1}^{(k)}\mathbf{e}_{j+1}^{(k)} + \cdots + \alpha_p^{(k)}\mathbf{e}_p^{(k)}$, where the $p{-}1$ abundance fractions $\{\alpha_i^{(k)}\}_{i \neq j, i=1}^{p}$ are obtained by the FCLS.
3. Subtract $\hat{\mathbf{e}}_j^{(k)}$ from $\mathbf{e}_j^{(k)}$ to form an error vector $\boldsymbol{\varepsilon}_j^{(k)} = \mathbf{e}_j^{(k)} - \hat{\mathbf{e}}_j^{(k)}$, which will be used as a projection vector.
4. Project all image pixels onto the projection vector $\boldsymbol{\varepsilon}_j^{(k)}$and find the pixel, $\mathbf{e}_j^*$ that yields the maximum length, that is, $\mathbf{e}_j^* = \max_{\mathbf{r}}\{\mathbf{r}^T\boldsymbol{\varepsilon}_j^{(k)}\} = \max_{\mathbf{r}}\{\mathbf{r}^T\mathbf{e}_j^{(k)} - \mathbf{r}_j^T\hat{\mathbf{e}}_j^{(k)}\}$. Let $j^* = \arg\{\max_j \mathbf{e}_j^*\}$. We form a new set of p endmembers by letting $\mathbf{e}_i^{(k+1)} = \mathbf{e}_i^{(k)}$ for $1 \leq i \leq p, ij^*$and $\mathbf{e}_{j^*}^{(k+1)} = \mathbf{e}_{j^*}^*$
5. If $\{\mathbf{e}_1^{(k+1)}, \mathbf{e}_2^{(k+1)}, \ldots, \mathbf{e}_p^{(k+1)}\} = \{\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}\}$. Otherwise, let $k\ k+1$. Go to step 2.

Figure 7.9 shows the results of nine endmembers extracted by FCLS-EEA using two different sets of random initial endmembers where inconsistency caused by random initial conditions in final selected endmembers was also evidential with two panel pixels extracted in the first run and only one panel pixel extracted in the second run.

From Figure 7.9, it can be seen that FCLS-EEA did not work as well as it was designed for. Three major factors contribute to such poor performance.

1. According to Heinz and Chang (2001) and Chang (2003a) it required at least more than 30 target pixels for FCLS to perform well. Obviously the number of endmembers used by FCLS-EEA, which was 9 estimated by VD, was simply not large enough for FCLS-EEA to work effectively.

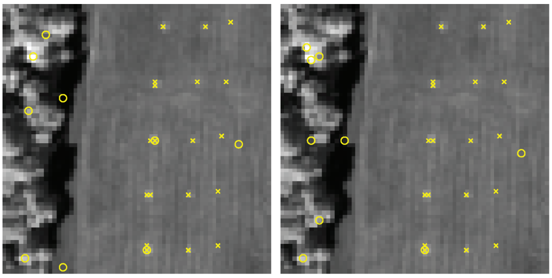

(a) run 1
(b) run 2

Figure 7.9 Nine endmembers extracted by the FCLS-EEA on the HYDICE data with two different runs.

2. The new endmembers found by different error vectors were usually the same. This is because these error vectors behaved like skewers used by PPI, which were independent of each other.
3. The stopping rule made the algorithm stop too early since the data sample vector at same position was usually replaced by the same endmember at the second iteration.

In order to alleviate the second and third problems, OSP is applied as a new learning rule and the difference between error vectors in length is used as a stopping rule. Here, OSP is chosen to find a new endmember because it can find the most distinct candidate from the existing ones. Unlike the error vectors, the new endmember generated by OSP will not be the same at different iterations. In this case, the stopping rule takes advantage of an underlying principle for LSMA, that is, the error between a new real endmember found at $(k + 1)$ st iteration and any data sample vector linearly unmixed by using previously found endmembers at the kth iteration is maximal. If the new endmember found at $(k + 1)$st iteration does not satisfy the condition, then the stopping rule terminates the algorithm. The above FCLS-EEA can be modified as follows.

FCLS-EEA Algorithm

1. Initialization.
   Set p to be the number of endmembers to be generated.
   Let $\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}$ be a set of initial endmembers and $\mathbf{U}^{(0)} = \left[\mathbf{e}_1^{(0)}\,\mathbf{e}_2^{(0)} \cdots \mathbf{e}_p^{(0)}\right]$ be its corresponding endmember matrix.
2. At iteration $k \geq 0$, apply an orthogonal subspace projector $P_{\mathbf{U}^{(k)}}^{\perp}$ to all data sample vectors $\mathbf{r}$ and find a sample vector $\mathbf{e}^{(k,*)}$ that yields the maximum orthogonal projection by
$$\mathbf{e}^{(k,*)} = \arg\left\{\max_{\mathbf{r}}\left[\left(P_{\mathbf{U}^{(k)}}^{\perp}\mathbf{r}\right)^T\left(P_{\mathbf{U}^{(k)}}^{\perp}\mathbf{r}\right)\right]\right\}. \tag{7.17}$$
3. For $1 \leq j \leq p$, assume that the jth endmember at the kth iteration, $\mathbf{e}_j^{(k)}m$, is not a pure signature but rather a mixed sample $\hat{\mathbf{e}}_j^{(k)}$ expressed as a linear mixture of all other endmembers, $p-1$ endmembers, $\left\{\mathbf{e}_i^{(k)}\right\}_{i=1,i\neq j}^{p}$ given by
$$\hat{\mathbf{e}}_j^{(k)} = \alpha_1^{(k)}\mathbf{e}_1^{(k)} + \cdots + \alpha_{j-1}^{(k)}\,\mathbf{e}_{j-1}^{(k)} + \alpha_{j+1}^{(k)}\,\mathbf{e}_{j+1}^{(k)} + \cdots + \alpha_p^{(k)}\mathbf{e}_p^{(k)}, \tag{7.18}$$
   where $\{\alpha_i^{(k)}\}_{i=1,i\neq j}^{p}$ are abundance fractions obtained by FCLS.
4. Calculate the error between the pure signature $\mathbf{e}_j^{(k)}$ and the mixed signature $\hat{\mathbf{e}}_j^{(k)}$by $\boldsymbol{\varepsilon}_j^{(k)} = \mathbf{e}_j^{(k)} - \hat{\mathbf{e}}_j^{(k)}$ and find the index that yields the smallest error, that is, $j^* = \arg\{\min_j |\boldsymbol{\varepsilon}_j^{(k)}|\}$.
5. Now calculate the error vector $\boldsymbol{\varepsilon}^{(k)} = \mathbf{e}^{(k,*)} - \mathbf{e}_{j^*}^{(k)}$ and check whether $|\boldsymbol{\varepsilon}^*| \leq \left|\boldsymbol{\varepsilon}_{j^*}^{(k)}\right|$,
6. If it is, the algorithm is terminated and $\mathbf{U}^{(k)} = \left[\mathbf{e}_1^{(k)}\mathbf{e}_2^{(k)} \cdots \mathbf{e}_p^{(k)}\right]$ is the set of the final endmembers. Otherwise, let $\mathbf{e}_i^{(k+1)} = \mathbf{e}_i^{(k)}$ for $1 \leq i \leq p$, $i \neq j^*$ and $\mathbf{e}_{j^*}^{(k+1)} = \mathbf{e}^{(k,*)}$; go to step 2.

Figure 7.10 shows the results of the modified FCLS-EEA with two different runs. The improvement is very obvious from these results.

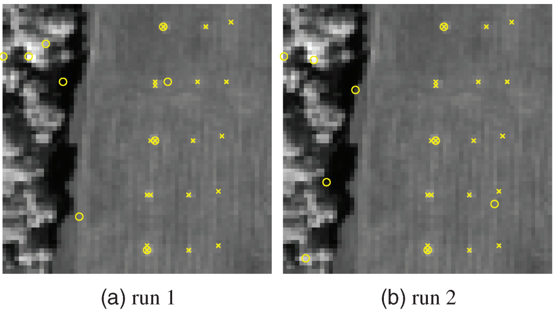


Figure 7.10 Nine endmembers extracted by the modified FCLS-EEA on the HYDICE data with two different runs.

7.3 Second-Order Statistics-Based Endmember Extraction

Based on the nature of pure signatures using the concept of convexity geometry as a criterion to find endmembers seems natural and logical, but may not be the only way to accomplish the task of endmember extraction. Instead of appealing for geometric features such as convex hull, convex cone, and simplex described above, a rather different approach that uses the statistical spectral profile of a signature as a base to determine an endmember is of interest. More specifically, a set of spectrally distinct endmembers should constitute the least statistical spectral correlation among all possible data sets with the same number of data samples. In other words, if there is a data sample that is a mixture of other data samples in the same set, the statistical spectral correlation among members in this set should be greater than that of a set with the same number of data samples that are all endmembers. In this section, we explore such statistics-based approaches to endmember extraction. In particular, we are interested in second-order statistics with two criteria, statistical sample spectral correlation and LSE that can be used to design EEAs.

The second-order statistical sample correlation was previously explored by Singh and Harison (1985) who expressed the second-order statistics information in terms of correlation coefficients. Their idea was used by Eklundh and Singh (1993) to develop the so-called standardized principal components analysis (SPCA). Here, we further take advantage of SPCA developed by Eklundh and Singh to derive an approach to endmember extraction, called standardized PCA-based EEA (SPCA-EEA). The key idea behind the proposed SPCA-EEA is to assume that the information represented by correlation coefficients among endmembers is minimal. More specifically, for a given set of p signatures the least amount of second-order statistical information among all possible p signatures is one that is formed by p distinct endmembers. If one of p signatures is a mixed signature by the other $(p-1)$ endmembers, the shared second-order statistical information must be at least equal to or greater than that represented by the p endmembers.

More specifically, let $\{\mathbf{e}_j\}_{j=1}^p$ be a set of p endmembers and μ_j and σ_j be the mean and standard deviation of the jth endmember $\mathbf{e}_j$. $\mathbf{E} = \left[(\mathbf{e}_1 - \mu_1)/\sigma_1(\mathbf{e}_2 - \mu_2)/\sigma_2 \cdots \left(\mathbf{e}_p - \mu_p\right)/\sigma_p\right]$ be a normalized endmember matrix. Then we can define an inner product of $\mathbf{E}$ as $\mathbf{K} = \mathbf{E}^T\mathbf{E}$. Assume that $\{\lambda_j\}_{j=1}^p$ are eigenavlues of $\mathbf{K}$. Then the determinant of $\mathbf{K}$ is given by $\det(\mathbf{K}) = \prod_{j=1}^p \lambda_j$ and can be used as a criterion to design an SPCA-based EEA as follows.

SPCA-EEA Algorithm

1. Initialization.
   Set p to be the number of endmembers to be generated.
   Assume that $\{\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}\}$ is a set of initial endmembers.
2. Outer loop index by k
   At iteration $k \geq 0$, for $1 \leq j \leq p$ we normalize $\bar{\mathbf{e}}_j^{(k)}$ to zero mean and unit variance by $\bar{\mathbf{e}}_j^{(k)} = (\mathbf{e}_j^{(k)} - \mu_j^{(k)})/\sigma_j^{(k)}$, where $\mu_j^{(k)}$ and $\sigma_j^{(k)}$ are the mean and standard deviation of $\mathbf{e}_j^{(k)}$. Now we form a $p \times p$ normalized endmember matrix $\mathbf{K}_e^{(k)} = \left(\bar{\mathbf{E}}^{(k)}\right)^T \bar{\mathbf{E}}^{(k)}$. Define the determinant of $\mathbf{K}_{\mathbf{e}}^{(k)}$ by

$$\varsigma^{(k)}\left(\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}\right) = \det\left(\mathbf{K}_e^{(k)}\right). \tag{7.19}$$

3. Inner loop index by j
   For $j \geq 1$, we calculate $\varsigma^{(k)}\left(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j-1}^{(k)}, r, \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}\right)$ via (7.19) for all image pixels $\mathbf{r}$ and find

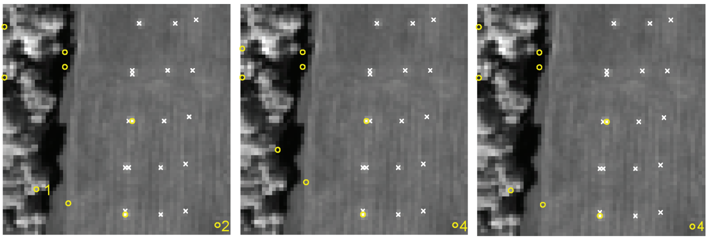
$$\mathbf{e}_j^* = \arg\left\{\max_{\mathbf{r}}\left[\varsigma^{(k)}\left(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j-1}^{(k)}, \mathbf{r}, \mathbf{e}_{j+1}^{(k)} \ldots, \mathbf{e}_p^{(k)}\right)\right]\right\}. \tag{7.20}$$

4. If $\varsigma^{(k)}\left(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j-1}^{(k)}, \mathbf{e}_j^*, \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}\right) > \varsigma^{(k)}\left(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j-1}^{(k)}, \mathbf{e}_j^{(k)}, \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}\right)$, then let $\mathbf{e}_j^{(k,j)} = \mathbf{e}_j^{(*)}$ and $\varsigma^{(k,j)} = \varsigma^{(k)}\left(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j-1}^{(k)}, \mathbf{e}_j^*, \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}\right)$. Increase j by one by setting $j \leftarrow j+1$ and go to step 4. Otherwise, let $\mathbf{e}_j^{(k,j)} = \mathbf{e}_j^{(k)}$ and $\varsigma^{(k,j)} = \varsigma^{(k)}\left(\mathbf{e}_1^{(k)}, .\,^{\cdot}, \mathbf{e}_{j-1}^{(k)}, \mathbf{e}_j^{(k)}, \mathbf{e}_{j+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}\right)$ and continue.
5. If $j < p$, then go to step 4. Otherwise, continue.
6. Let $j^* = \arg\{\max_{1 \leq j \leq p}\{\varsigma^{(k,j)}\}\}$ with $\varsigma^{(k,j^*)} = \varsigma^{(k)}\left(\mathbf{e}_1^{(k)}, \ldots, \mathbf{e}_{j^*-1}^{(k)}, \mathbf{e}_{j^*}^{(k,j^*)}, \mathbf{e}_{j^*+1}^{(k)}, \ldots, \mathbf{e}_p^{(k)}\right)$. Define $\mathbf{e}_{j^*}^{(k+1)} = \mathbf{e}_{j^*}^{(k,j^*)}$ and $\mathbf{e}_j^{(k+1)} = \mathbf{e}_j^{(k)}$for $j \neq j^*$. If the newly obtained $(k+1)$st endmember set $\{\mathbf{e}_1^{(k+1)}, \ldots, \mathbf{e}_{j^*}^{(k+1)}, \ldots, \mathbf{e}_p^{(k+1)}\}$ is identical to the kth endmember set $\{\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}\}$, that is, $\{\mathbf{e}_1^{(k+1)}, \ldots, \mathbf{e}_{j^*}^{(k+1)}, \ldots, \mathbf{e}_p^{(k+1)}\} = \{\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}\}$, the algorithm is terminated. Otherwise, let $k \leftarrow k+1$. Go to step 2.

To demonstrate the utility of SPCA-EEA in endmember extraction Figure 7.11 shows nine endmembers extracted by SPCA-EEA using three different sets of random initial endmembers where only two panel pixels in rows 3 and 5 were extracted to correspond to two endmembers.

In order to take into account the mixing coefficients of endmembers constrained by convexity, an FCLS method developed by Heinz and Chang (2001) is included in the proposed SPCA-EEA to ensure that all the SPCA-EEA found endmembers encompass all possible data samples as their mixing signatures.

In order to take care of the sum-to-one constraint into the algorithm, a p-dimensional column vector $\mathbf{1}_p = (\underbrace{1, 1, \ldots, 1}_{p})^T$ should be included. This is similar to FCLS and N-FINDR. That is, let $\mathbf{E} = \left[\mathbf{e}_1\, \mathbf{e}_2 \cdots \mathbf{e}_p\right]$ be an $L \times p$ endmember matrix and $\mathbf{1}_p = (\underbrace{1, 1, \ldots, 1}_{p})^T$ be a p-dimensional unity



(a) run 1 (b) run 2 (c) run 3

Figure 7.11 Nine endmembers extracted by SPCA-EEA using random initial conditions in three different runs.

vector. We introduce an $(L+1) \times p$ matrix $\bar{\mathbf{E}}^{\Delta}_{(L+1)\times p}$ by augmenting the matrix $\mathbf{E}$ by including a p-dimensional unity vector $\mathbf{1}_p$ as follows:

$$\bar{\mathbf{E}}^{\Delta}_{(L+1)\times p} = \begin{bmatrix} \mathbf{E} \\ \Delta \mathbf{1}_p^T \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \cdots & \mathbf{e}_{p-1} & \mathbf{e}_p \\ \Delta & \Delta & \cdots & \Delta & \Delta \end{bmatrix} = \begin{bmatrix} e_{11} & e_{12} & \cdots & e_{1(p-1)} & e_{1p} \\ e_{21} & e_{22} & \cdots & e_{2(p-1)} & e_{2p} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ e_{L1} & e_{L2} & \ddots & e_{L(p-1)} & e_{Lp} \\ \varDelta & \varDelta & \cdots & \varDelta & \varDelta \end{bmatrix} \tag{7.21}$$

where the parameter Δ is included to control the effect of ASC. It should be noted that (7.21) is the same matrix used in FCLS algorithm in Heinz and Chang (2001), the same endmember matrix $\mathbf{E}$ used in N-FINDR and $\mathbf{E}^{(k)}_{L\times p}$ used in step 2 of AN-FINDR. In this case, the p-dimensional $\mathbf{q}_l^{(k)} = \left(e_{l1}^{(k)}, e_{l2}^{(k)}, \ldots, e_{lp}^{(k)}\right)^T$ column vector used in step 2 of the PCA-EEA is extended to a $(p+1)$ -dimensional column vector defined by $\bar{\mathbf{q}}_l^{(k)} = \left(e_{l1}^{(k)}, e_{l2}^{(k)}, \ldots, e_{lp}^{(k)}, \Delta\right)^T$.

A note is worthwhile. One may wonder "why cannot we use the commonly used PCA instead of SPCA to design an EEA?". The answer to this question can be very enlightening. PCA only considers data variances while discarding co-variances that are crucial in endmember extraction. On the other hand, SPCA explores statistical correlation by correlation coefficients that are indeed obtained from both variances and co-variances. However, since an endmember is relatively rare, its presence may be more appropriately characterized by high-order statistics (HOS) rather than second-order statistics such as variance. This fact gives rise to the possibility of using HOS as criteria to develop EEAs, which will be discussed in Chapter 8. One disadvantage of using HOS is that there is no analytical form that can be derived for an HOS-based SM-EEA in a similar way that PCA solves the characteristic polynomial equation to find all eigenvalues simultaneously. Instead, an HOS-based EEA must rely on a numerical algorithm to generate one projection vector to find one endmember at a time. Therefore, an HOS-based EEA is an SQ-EEA and cannot be an SM-EEA.

7.4 Automated Morphological Endmember Extraction (AMEE)

The AMEE algorithm is an endmember extraction algorithm that makes simultaneous use of spatial and spectral information via multi-channel morphological processing (Plaza et al., 2002). The input to AMEE is the full image data cube, with no need of dimensionality reduction. Let $\mathbf{r}$ denote the input data cube and $\mathbf{r}(x,y)$ denote the pixel vector at spatial location (x,y). Similarly, let $\mathbf{K}$ be a kernel defined in the spatial domain of the image (the *x*–*y* plane). This kernel, usually called structuring element (SE) in mathematical morphology terminology, is translated over the image. The SE acts as a probe for extracting or suppressing specific structures of the image objects, according to the size and shape of the SE. Having the above definitions in mind, AMEE method is based on the application of multichannel erosion and dilation operations to the data. The above operations are defined as follows:

$$\begin{aligned}(\mathbf{r}\otimes K)(x,y)&=\arg\left\{\min\nolimits_{(s,t)\in K}\left[\textstyle\sum_s\sum_t \mathrm{SAM}(\mathbf{r}(x,y),\mathbf{r}(x+s,y+t))\right]\right\}\\(\mathbf{r}\oplus K)(x,y)&=\arg\left\{\max\nolimits_{(s,t)\in K}\left[\textstyle\sum_s\sum_t \mathrm{SAM}(\mathbf{r}(x,y),\mathbf{r}(x-s,y-t))\right]\right\}\end{aligned}\tag{7.22}$$

where SAM is the spectral angle mapper (SAM). Multichannel erosion (respectively, dilation) selects the pixel vector that minimizes (respectively, maximizes) a cumulative distance-based cost function, based on the sum of the SAM distance scores between each pixel in the spatial neighborhood and all the other pixels in the neighborhood. As a result, multichannel erosion extracts the pixel vector that is more similar to its neighbors as opposed to multichannel dilation, which extracts the most spectrally distinct pixel in the neighborhood (endmember candidate). It should be noted that, according to the definition of morphological erosion and dilation, the above operations are sensitive to the size and shape of the SE used in the computation. In our application, a morphological eccentricity index (MEI) is defined for each endmember candidate by calculating the SAM distance between the pixel provided by the dilation operation and the pixel provided by the erosion. This operation is repeated for all the pixels in the scene, using SEs with a range of different sizes, until a final MEI image is generated. Endmember selection is then accomplished by a fully automated approach that consists of two steps (Plaza et al., 2002): (1) automated segmentation of the MEI image using Otsu's method (Otsu, 1979; Chang et al., 2006); (2) spatial/spectral region growing of resulting regions. Like PPI that extracted pixels according to PPI counts, AMEE also extracts all pixels according to their MEIs. As a result, both PPI and AMEE generally extract more than one endmember, usually many endmembers representing a single signature. In practical implementation, the AMEE is modified by including a process such as OSP-based classifier to select only one endmember representing each signature, while discarding redundant endmembers that also represent the same signature.

7.5 Experiments

This section presents experimental studies on performance analysis of six SM-EEAs representing four different criteria, that is, PPI from convexity geometry via OP, N-FINDR and AN-FINDR from finding maximum simplex volume, SPCA-EEA from statistical correlation, FCLS-EEA from LSE-based fully abundance-constrained spectral unmixing, and AMEE from morphology. Three sets of experiments are conducted for performance evaluation, which are (1) six synthetic image-based scenarios discussed in Chapter 4, (2) AVIRIS and (3) HYDICE real image experiments. Since MNF is one of the most widely used techniques to perform DR in the literature, it is used for all EEAs that require DR.

7.5.1 Synthetic Image Experiments

The synthetic images used for experiments were the three scenarios of target implantation (TI), TI1, TI2, and TI3, and the three scenarios of target embeddedness, (TE), TE1, TE2, and TE3, are described in Chapter 4 and reproduced in Figure 7.12 for reference.

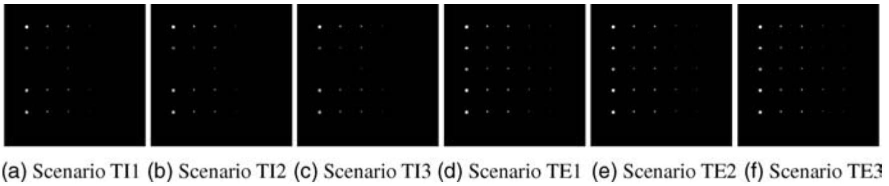


Figure 7.12 Three scenarios designed for TI, TI1, TI2, and TI3 and three scenarios of TE, TE1, TE2, and TE3.

Six SM-EEAs, MATLAB-PPI (PPI) with 500 skewers, 1-IN-FINDR, AN-FINDR, SPCA-EEA, FCLS-EEA, and AMEE, were implemented on these six scenarios to extract endmember pixels. Based on the ground truth in Figure 4.2 and Tables 4.1 and 4.2, there are 100 pure pixels, 20 mixed pixels, and 10 subpixels, all of which are simulated by five distinct pure mineral signatures. So, there are a total of six spectrally distinct signatures, five endmembers $\{\mathbf{m}_i\}_{i=1}^5$ plus a mixed background signature **b**. Therefore, two sets of experiments were conducted by assuming that $p = 5$ to represent only five endmembers (pure signatures) and $p = 6$ to indicate that there are six spectrally distinct signatures (five endmembers plus a mixed background signature) present in the data. Two preprocesses were generally required for SM-EEAs: (1) DR and (2) use of a random generator to produce initial endmembers. So, in all experiments MNF was used to perform DR for TI2, TI3, TE2, and TE3, while PCA was used for TI1 and TE1 because MNF was not applicable to noise-free scenarios TI1 and TE1. Furthermore, the reduced dimensionality is set to the same value as p. In addition, to demonstrate inconsistent results caused by the use of random initial endmembers, two sets of randomly generated initial endmembers were used to initialize the six EEAs for illustration.

It should be also noted that in the following experiments the results of FCLS-EEA are absent in scenarios of TI1, TE1, TI2, and TE2 due to the fact that the endmember matrix found by FCLS for spectral unmixing was not of full rank. This is particularly true for TI and TE1 because no noise is present in the data and the number of bands is greater than the number of endmembers unless a DR is performed to reduce the data dimensionality to the number of endmembers. In scenarios of TI2 and TE2, the same issue may occur to FCLS-EEA when randomly generated initial endmembers happened to be pixels in the same target panel. Since scenarios TI3 and TE3 have simulated noise added to all image pixels including endmember panel pixels, there is no issue of ill rank in FCLS-EEA.

7.5.1.1 Scenario TI1 (Endmembers Implanted in a Clean Background)

Scenario TI1 is an idealistic case where pure panel pixels considered as endmember pixels are implanted in a clean (i.e., noise-free) background. So, there are five endmembers $\{\mathbf{m}_i\}_{i=1}^5$ plus a mixed background signature **b**. Figure 7.11 shows respective results of five SM-EEAs, PPI with 500 skewers, 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, and AMEE using two different sets of randomly genertaed initial endmembers with $p = 5$ where extracted endmembers are marked by yellow circles. Interestingly, in this noise-free scenario none of five SM-EEAs extracted all five endmembers representing five distinct pure mineral signatures. Also, a comparison among results in Figures 7.13 and 7.14 shows that the results obtained by two different sets of random initial endmembers were not consistent.

The results produced by PPI were also worth noting. Due to no noise present in the data, the background signature dominated the entire data and was considered as a pure signature. As a result, in this scenario the PPI counts of all background pixels produced by PPI were constant and greater than the PPI counts of the five pure mineral signatures. So, the PPI image shown in Figure 7.6 is actually a PPI count map of all image pixels, which turns out to be a binary image

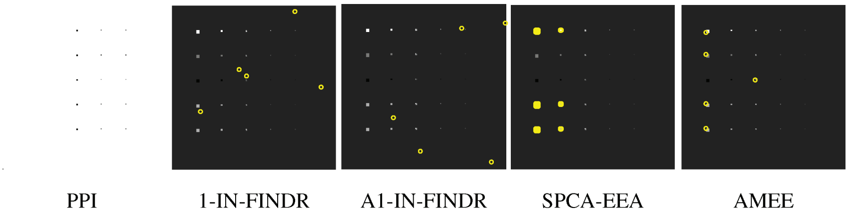


Figure 7.13 Results of endmember pixels extracted from TI1 by PPI along with five endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, and AMEE using two sets of random initial endmembers.

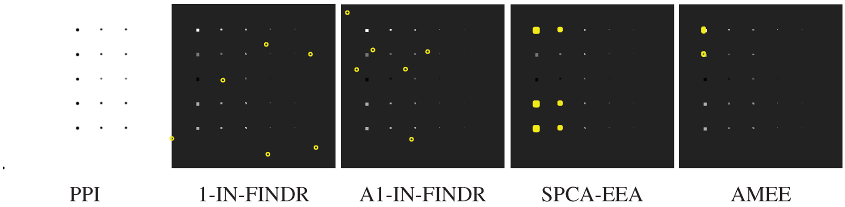


Figure 7.14 Results of endmember pixels extracted from TI1 by PPI along with six endmembers extracted by N-FINDR, AN-FINDR, SPCA-EEA, and AMEE using random initial endmembers.

with black pixels and white pixels representing panel pixels and background pixels, respectively. This scenario demonstrated a crucial fact that the PPI count of an endmember did not necessarily yield the highest value. On the contrary, in many cases the PPI counts of endmembers were usually very low but they were never equal to zero.

If the same experiments are repeated for $p = 6$, the results are shown in Figure 7.14 where the performance of the five SM-EEAs was not improved but rather worse than those obtained by $p = 7$.

7.5.1.2 Scenario TI2 (Endmembers Implanted in a Noisy Background)

In scenario TI2, pure panel pixels considered as endmembers are implanted into a Gaussian noise background with SNR = 20 : 1. Like scenario TI1, this particular synthetic image scene has 100 pure pixels, 20 mixed pixels, and 10 subpixels, all of which are simulated by five distinct pure mineral signatures $\{\mathbf{m}_i\}_{i=1}^{5}$. Similar to scenario TI1, this scenario also has a total of five endmembers plus a mixed signature, **b**. Figures 7.15 and 7.16 show the results of endmembers extracted by the five EEAs, PPI with 500 skewers, 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, and AMEE, with

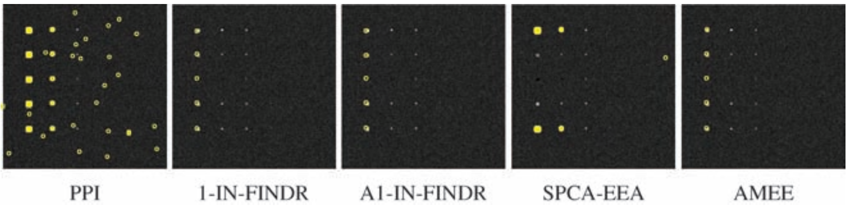


Figure 7.15 Results of endmember pixels extracted from TI2 by PPI along with five endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, and AMEE using random initial endmembers.

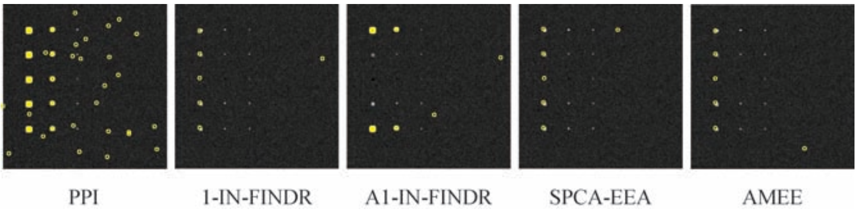


Figure 7.16 Results of endmember pixels extracted from TI2 by PPI along with six endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, and AMEE using random initial endmembers.

two different sets of randomly generated initial endmembers for $p=5$ and $p=6$, respectively, where extracted endmembers are marked by yellow circles.

As shown in Figures 7.15 and 7.16, all the five endmembers representing five distinct pure mineral signatures were successfully extracted by all the evalauted EEAs except SPCA-EEA. These results provide evidence that $p=5$ is sufficient for SM-EEAs to be able to find all required five endmembers in Figure 7.15. This is because the performance was not necessarily improved if $p=6$ is used, as shown in Figure 7.16. However, as discussed in Chapter 8, it will require $p=6$, not $p=5$, for an SQ-EEA to extract all the desired five endmembers because a background pixel will be always extracted before the last and fifth endmembers. Also noted in Figures 7.15 and 7.16 are PPI-extracted endmmembers, which included not only all 100 pure panel pixels but also many background pixels. This is because all pixels extracted by PPI had their PPI counts greater than zero and the PPI counts of many background pixels were actually higher than the PPI counts of panel pixels.

7.5.1.3 Scenario TI3 (Noisy Endmembers Implanted in a Noisy Background)

Scenario TI3 is simulated by directly adding a Gaussian noise with $\text{SNR}=20:1$ to scenario TI1 where all the pure panel pixels are noise-corrupted signatures and no longer pure signatures. In this case, there are no endmembers in this synthetic image scene. Under such circumstances, an EEA intends to find the purest signatures in the image scene. Nevertheless, this scenario still contains a total of six spectrally distinct signatures, which are the five noise-corrupted endmembers plus a mixed background signature. Figures 7.17 and 7.18 show the results of endmembers extracted by the six EEAs, PPI with 500 skewers, N-FINDR, AN-FINDR, SPCA-EEA, FCLS-EEA, and AMEE with two different sets of randomly generated initial endmembers for $p=5$ and $p=6$, respectively, where extracted endmembers are marked by yellow circles.

Since the experimental results in Figures 7.17 and 7.18 were very similar to those obtained in Figures 7.15 and 7.16, the conclusions drawn for TI2 also hold for TI3. Only difference between TI2 and TI3 is that FCLS-EEA, which was absent in Figure 7.15 due to an ill-rank of the

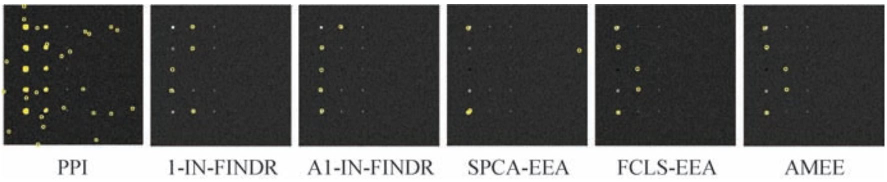


Figure 7.17 Results of endmember pixels extracted from TI3 by PPI along with five endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, FCLS-EEA, and AMEE using random initial endmembers.

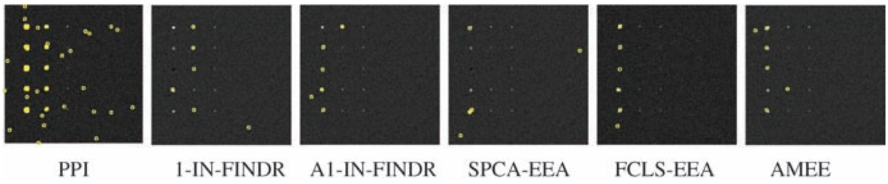

PPI 1-IN-FINDR A1-IN-FINDR SPCA-EEA FCLS-EEA AMEE

Figure 7.18 Results of endmember pixels extracted from TI3 by PPI along with six endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, FCLS-EEA, and AMEE using random initial endmembers.

endmember signature matrix used for spectral unmixing, performed at least as well as any other SM-EEAs in extracting all the five endmembers.

7.5.1.4 Scenario TE1 (Endmembers Embedded into a Clean Background)

This scenario has 25 simulated panels added to and superimposed on background pixels. Therefore, there are no pure target endmember pixels in this scenario; rather six spectrally distinct signatures and five background-superimposed pure signatures can be considered as the purest signatures. The experiments conducted for Scenario TI are also performed for TE1 and the results are shown in Figures 7.19 and 7.20 for $p = 5$ and $p = 6$ using two different sets of randomly generated initial endmembers where the extracted endmembers by five SM-EEAs, PPI with 500 skewers, 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, and AMEE, are marked by yellow circles. As evident from these results, only AMEE was able to extract all the five endmembers.

7.5.1.5 Scenario TE2 (Endmembers Embedded into a Noisy Background)

TE1 is different from TE1 in that it embeds clean signatures into noisy background instead of clean background. Nevertheless, TE2 still has six spectrally distinct signatures and five noise-added and background-superimposed pure signatures can be also considered as the purest

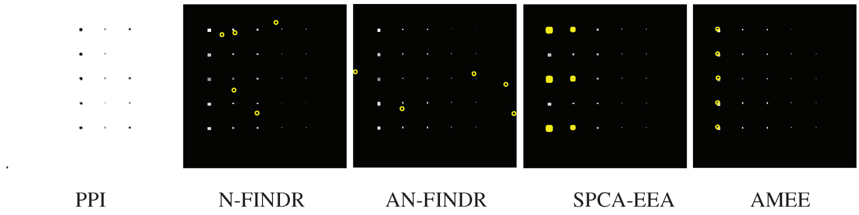

PPI N-FINDR AN-FINDR SPCA-EEA AMEE

Figure 7.19 Results of endmember pixels extracted from TE1 by PPI along with five endmembers extracted by N-FINDR, AN-FINDR, SPCA-EEA, and AMEE using random initial endmembers.

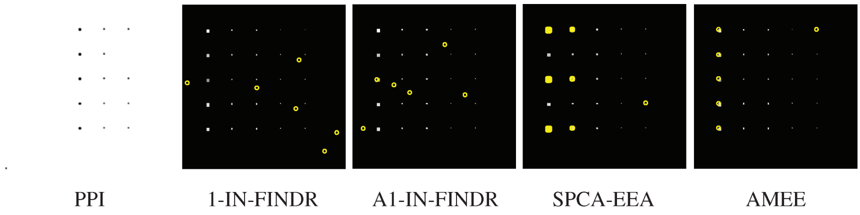

PPI 1-IN-FINDR A1-IN-FINDR SPCA-EEA AMEE

Figure 7.20 Results of endmember pixels extracted from TE1 by PPI along with six endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, and AMEE using random initial endmembers.

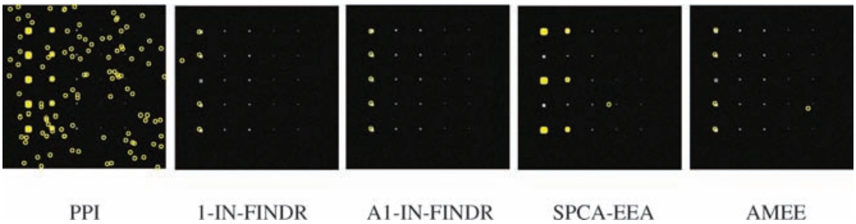


Figure 7.21 Results of endmember pixels extracted from TE2 by PPI along with five endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, and AMEE using random initial endmembers.

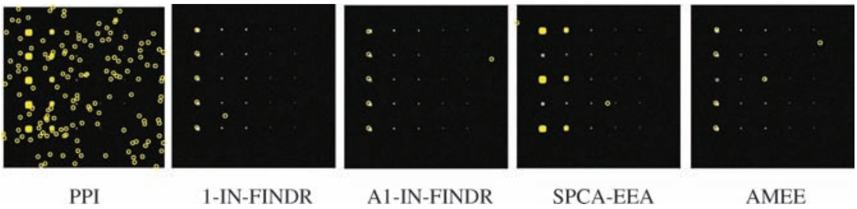


Figure 7.22 Results of endmember pixels extracted from TE2 by PPI along with six endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, and AMEE using random initial endmembers.

signatures. Figures 7.21 and 7.22 show the results of endmembers extracted by the five EEAs, PPI with 500 skewers, N-FINDR, AN-FINDR, SPCA-EEA, and AMEE, with two different sets of randomly generated initial endmembers for $p=5$ and $p=6$, respectively, where extracted endmembers are marked by yellow circles. As shown in Figures 7.21 and 7.22, despite that the pure signaures have been added by noise-corrupted background signatures, the two SM-EEAs, PPI and A1-IN-FINDR, still managed to extract all five endmember pixels for $p=5$ and $p=6$, while 1-IN-FINDR misses one endmember pixel in the third row for $p=5$ but extracts it for $p=6$, AMEE failed to extract one endmember pixel in the third row for both endmember pixels in the second and fourth rows for both $p=5$ and $p=7$. The experiments show that except for N-FINDR increasing p from 5 to 6 did not really help.

7.5.1.6 Scenario TE3 (Noisy Endmembers Embedded into a Noisy Background)

In this scenario, a white Gaussian noise is added to Scenario TE1 where pure signatures and background signatures are corrupted by noise. In this case, the pure signatures are no longer clean as they are in TE2, but noise-corrupted. Nevertheless, these five noise-corrupted as well as background-superimposed pure signatures can be considered as the purest signatures. Figures 7.23 and 7.24 show the results of endmembers extracted by the five EEAs, PPI with 500 skewers, N-FINDR, AN-FINDR, SPCA-EEA, FCLS-EEA, and AMEE with two different sets of randomly generated initial endmembers for $p=5$ and $p=6$, respectively, where extracted endmembers are marked by yellow circles. As expected, the results in Figures 7.23 and 7.24 were not be as good as but comparable to those obtained for TE2, where PPI and AN-FINDR were the only two SM-EEAs able to extract endmember pixels in all the five rows for $p=5$ and 7. Like TE2, N-FINDR misses endmember pxiels in the third row for $p=5$, but picks it up for $p=7$. Both SPCA-EEA and FCLS-EEA missed endmember pixels in one row for both $p=5$ and $p=6$, whereas AMEE missed endmember pixels in two rows for $p=5$ and one row for $p=7$.

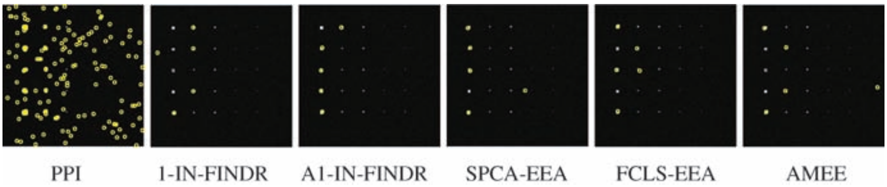


Figure 7.23 Results of endmember pixels extracted from TE3 by PPI along with five endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, FCLS-EEA, and AMEE using random initial endmembers.

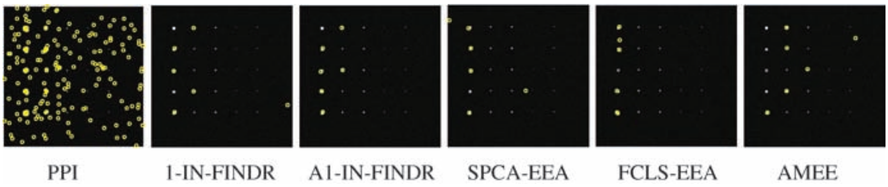


Figure 7.24 Results of endmember pixels extracted from TE3 by PPI along with six endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, FCLS-EEA, and AMEE using random initial endmembers.

7.5.2 Cuprite Data

In analogy with the experiments conducted for the simulated data the six EEAs, PPI with 1000 skewers, N-FINDR, AN-FINDR, SPCA-EEA, FCLS-EEA, and AMEE are implemented on the Cuprite image scene in Figure 1.11(a) with initial endmembers randomly generated and the number of endmembers, p, estimated by $\mathrm{VD}_{\mathrm{NP}}^{\mathrm{HFC}}(P_F = 10^{-4}) = 22$ in Chapter 5. The results using two different sets of randomly generated initial endmembers are shown in Figure 7.25 where target pixels extracted by algorithms are marked by yellow circles, pixels marked by yellow crosses “x” are the five ground truth mineral pixels, and the pixels marked by yellow triangles are identified by SM-EEAs corresponding to the five true mineral signatures.

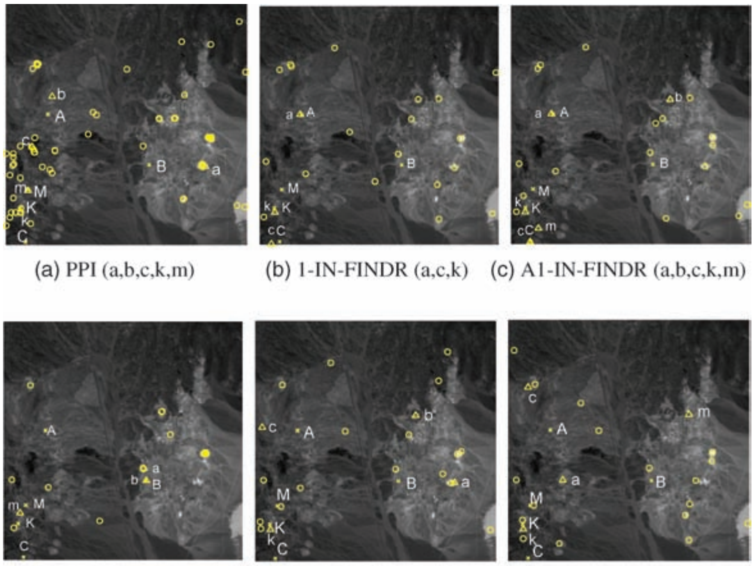
As shown in Figure 7.25, the endmember pixels extracted by SM-EEAs and labeled by “lower case letters” are not necessarily the same ground truth pixels labeled by “upper case letters.” These endmember pixels are identified using the correlation matched filter-based distance (RMFD) developed by Chang and Liu (2004), which is similar to (16.21) in Chapter 16 and defined as

$$\mathrm{RMFD}(\mathbf{t}_i, \mathbf{t}_j) = \mathbf{t}_i^T \mathbf{R}^{-1} \mathbf{t}_j \tag{7.23}$$

and has been shown to perform significantly better and more effectively than the commonly used pixel level-based SAM in discrimination and identification of subpixels and mixed pixels for real hyperspectral images. In (7.23) the matrix $\mathbf{R}$ is the sample correlation matrix and $\mathbf{t}_i$ and $\mathbf{t}_j$ are two target pixels to be discriminated. Therefore, the RMFD was used to identify the 22 target pixels against the five minerals of interest, A, B, C, K, and M by RMFD via (7.23) where the signatures of the five minerals in Figure 1.11(c) were used.

7.5.3 HYDICE Data

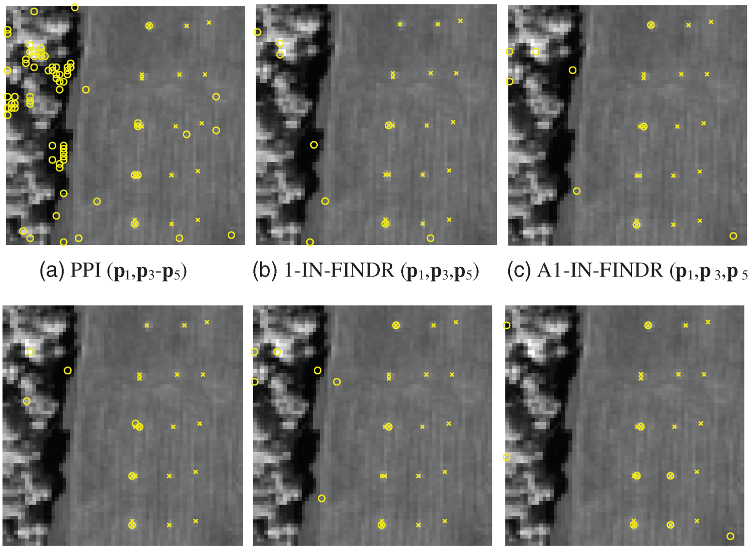
The HYDICE scene shown in Figure 1.15(a) is also used for experiments. Figures 7.26(a)–(f) and 7.27(a)–(f) show results obtained by PPI using $k = 1000$ skewers, 1-IN-FINDR, A1-IN-FINDR,



(a) PPI (a,b,c,k,m) (b) 1-IN-FINDR (a,c,k) (c) A1-IN-FINDR (a,b,c,k,m)

(d) SPCA-EEA (a,b,k,m) (e) FCLS-EEA (a,b,c,k) (f) AMEE (a,c,k,m)

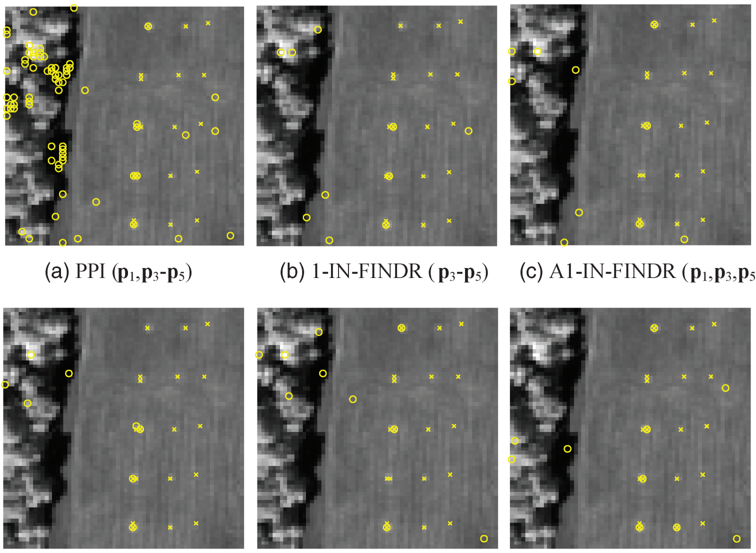
Figure 7.25 Results of endmember pixels extracted by PPI along with 22 endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, FCLS-EEA, and AMEE using one set of random initial endmembers.



(a) PPI ($\mathbf{p}_1$,$\mathbf{p}_3$-$\mathbf{p}_5$) (b) 1-IN-FINDR ($\mathbf{p}_1$,$\mathbf{p}_3$,$\mathbf{p}_5$) (c) A1-IN-FINDR ($\mathbf{p}_1$,$\mathbf{p}_3$,$\mathbf{p}_5$)

(d) SPCA-EEA ($\mathbf{p}_3$-$\mathbf{p}_5$) (e) FCLS-EEA ($\mathbf{p}_1$,$\mathbf{p}_3$,$\mathbf{p}_5$) (f) AMEE ($\mathbf{p}_1$,$\mathbf{p}_3$-$\mathbf{p}_5$)

Figure 7.26 Results of endmember pixels extracted by PPI and AMEE along with nine endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, FCLS-EEA, and AMEE using one set of random initial endmembers.



(a) PPI ($\mathbf{p}_1$,$\mathbf{p}_3$-$\mathbf{p}_5$) (b) 1-IN-FINDR ($\mathbf{p}_3$-$\mathbf{p}_5$) (c) A1-IN-FINDR ($\mathbf{p}_1$,$\mathbf{p}_3$,$\mathbf{p}_5$)

(d) SPCA-EEA ($\mathbf{p}_3$-$\mathbf{p}_5$) (e) FCLS-EEA ($\mathbf{p}_1$,$\mathbf{p}_3$,$\mathbf{p}_5$) (f) AMEE ($\mathbf{p}_1$,$\mathbf{p}_3$-$\mathbf{p}_5$)

Figure 7.27 Results of endmember pixels extracted by PPI and AMEE along with 10 endmembers extracted by 1-IN-FINDR, A1-IN-FINDR, SPCA-EEA, FCLS-EEA, and AMEE using random initial endmembers.

SPCA-EEA, FCLS-EEA, and AMEE for $p = 9$, and 10, respectively, where none of six SM-EEAs could extract panel pixels in row 2.

The results in Figures 7.26 and 7.27 demonstrate that no matter what value of p was selected, $p = 9$ or 10, the six EEAs had difficulty with extracting panel pixels in row 2. This is mainly due to the fact that panel pixels in rows 2 and 3 are made by the same fabrics with slightly different paints, Oliver and light Oliver. As a result, the spectral signatures of panel pixels in row 2 are very close to those in row 3. In this case, if a panel pixel in row 3 was extracted earlier, the panel pixels in row 2 were considered as the same signature with small variations, in which case no panel pixels in row 2 would be extracted later. A similar phenomenon is also witnessed in Figure 4.12(a) and (b) where the calcite was very close to the sample mean used as the background signature so that EEAs failed to extract it once the background signature was extracted first.

7.6 Conclusions

SM-EEAs are always desirable for endmember extraction. Unfortunately, a genuine SM-EEA is generally impractical because of its very high computational cost resulting from an exhaustive search, specifically when the number of endmembers, p, is large and data volume is huge. In addition, an SM EEA also suffers from several drawbacks: (1) requirement of precise knowledge about p, which is practically unknown; (2) assumption of endmembers present in the data, which is generally not true in many real applications; (3) necessity of dimensionality reduction (DR) due to enormous data volume, in which case selecting an effective DR transform is crucial; and (4) inconsistent results caused by the use of randomly generated initial endmembers. While some drawbacks such as p that can be estimated by the virtual dimensionality in Chapter 5, some

Table 7.1 Summary of design criteria of SM-EEAs and their drawbacks and disadvantages

SM-EEAs	Design criteria	Drawbacks/disadvantages
MATLAB-PPI	Orthogonal projection	1. Determination of the number of skewers, K 2. Determination of a threshold for PPI counts, t 3. Determination of the number of dimensions, q, for DR 4. Inconsistent results due to the use of random conditions 5. Many falsely extracted endmembers
SM N-FINDR, IN-FINDR, ASM N-FINDR	Maximal simplex volume	1. Precise knowledge about p 2. Determination of the number of dimensions, q, for DR 3. High computational complexity 4. Inconsistent results due to the use of random conditions 5. Assumption on presence of pure signatures
MVT	Minimal simplex volume	
CCA	Convex cone	1. Precise knowledge about p 2. Determination of the number of dimensions, q, for DR 3. High computational complexity 4. Assumption on presence of pure signatures
SPCA-EEA	Statistical spectral correlation	1. Precise knowledge about p 2. Inconsistent results due to the use of random conditions 3. Less effective
FCLS-EEA	Linear spectral unmixing	1. Precise knowledge about p 2. High computational complexity 3. Ill-rank of endmember signature matrix 4. Inconsistent results due to the use of random conditions
AMEE	Morphology	1. Precise knowledge about p 2. All endmembers of the same type are extracted before endmembers of another type 3. Inconsistent results due to the use of random conditions 4. A process is needed to discriminate extracted endmembers

other drawbacks such as computational complexity inheriting from the algorithm design. Table 7.1 summarizes design criteria of SM-EEAs discussed in this chapter and their drawbacks and disadvantages.

8

Sequential Endmember Extraction Algorithms (SQ-EEAs)

One major disadvantage of implementing a simultaneous endmember extraction algorithm (SM-EEA), as discussed in Chapter 7, is its high computational complexity and exceedingly high computing cost. This is because of the fact that an SM-EEA does not use the results produced by previous searches and a new full search must be resumed as long as previously found endmembers are not desired ones. In addition, its searching process must be conducted in an exhaustive manner for which the computation will become formidable once the number of endmembers to generate grows large. The sequential EEAs (SQ-EEAs), presented in this chapter, are the result of a need for addressing these two issues. An SQ-EEA is developed to sequentially find one endmember after another to address the first issue, where previously generated endmembers can be retained and included as part of a subsequent search for a new endmember. Because an SQ-EEA can only find one endmember at a time, its computational complexity is tremendously reduced which also resolves the second issue. The outcome of these two advantages is that an SQ-EAA may not necessarily produce an optimal solution as does an SM-EEA. Nevertheless, experiments demonstrate that the trade off is small and an SQ-EEA can perform almost equally like an SM-EEA provided that an SQ-EEA is appropriately designed to match its SM-EEA version. Interestingly, many SQ-EEAs have actually been derived from their SM-EEA counterparts in Chapter 7. Examples include vertex component analysis (Nascimento and Dias, 2005) from PPI, SQ N-FINDR (Wu et al., 2008), and simplex growing algorithm (Chang et al., 2006) from N-FINDR and unsupervised fully constrained least-squares (Heinz and Chang, 2002) from FCLS-EEA.

8.1 Introduction

In order to find endmembers, two general approaches have been used in the past. One approach extracts all required endmembers simultaneously, referred to as simultaneous endmember extraction algorithm (SM-EEA) that has already been studied in Chapter 7. The other approach extracts endmembers one at a time sequentially, referred to as sequential EEA (SQ-EEA) which will be discussed in this chapter. In general, extraction of endmembers must be performed by finding all endmembers at once. Hence, an optimal EEA should be an SM-EEA. Unfortunately, this also requires an SM-EEA to conduct an exhaustive search for an optimal set of endmembers. Computationally speaking, this may not be a good option because the process will be extremely slow,

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

especially when the number of endmembers grows. On the other hand, an SQ-EEA can become an acceptable alternative even if it may not be as optimal as an SM-EEA. As will be demonstrated by experiments, in most cases, an SQ-EEA is comparable to an SM-EEA with regard to performance of endmember extraction. Most importantly, two benefits gained by an SQ-EEA can really remedy two major drawbacks suffered from an SM-EEA. One benefit is significant reduction in computing time resulting from SQ-EEA that only needs to find one endmember at a time sequentially without finding all endmembers simultaneously as required by SM-EEA. The second benefit is that SQ-EEA retains previously generated endmembers while continuing to add new endmembers, an advantage that cannot be gaineded from SM-EEA. In particular, if the total number of endmembers to be extracted changes, SM-EEA must resume its exhaustive search as a new process for finding all new endmembers, and endmembers previously generated by SM-EEA for a small number of endmembers cannot be used as part of new endmembers. By contrast, SQ-EEA can always use previously generated endmembers as the starting point and continue to produce new endmembers afterwards. Interestingly, as will be shown in this chapter, an SM-EEA can always find its SQ-EEA counterpart so that both algorithms can produce similar results.

As noted in Chapter 7, for an EEA to be optimal it ought to be an SM-EEA that must conduct an exhaustive search for all endmembers at once. For example, for the N-FINDR to produce p endmembers among N data sample vectors a total of $\begin{pmatrix} N \\ p \end{pmatrix} = \frac{N!}{(N-p)!p!}$ searches must be conducted and the simplex volume specified by (7.5) must be recalculated for each of the searches. If the value of N becomes large, the computational complexity will increase exponentially and become formidable very quickly and going out of control. Therefore, two options can be adopted to alleviate this dilemma. One option narrows down a searching region to a feasible range. This practice is actually exercised in step 5 in SM N-FINDR, steps 4 and 5 in SPCA-EEA, and step 4 in FCLS-EEA discussed in Chapter 7, where their replacement rules are carried out in two loops (inner and outer loops) instead of being performed simultaneously. However, even in this case, the computation task can still be very high. Thus, a more pragmatic solution is to convert an SM-EEA into an SQ-EEA so that the computation can be reduced greatly by finding endmembers one at a time rather than all endmembers being generated at the same time. For example, we may implement an SM N-FINDR via its inner loop without running its outer loop, as discussed in Section 7.2.3.4. However, this is easier said than done. It is, in general, not a trivial matter since we need to make ensure that such an SM-EEA-to-SQ-EEA conversion via the use of only the inner loop will not be traded for poor performance. This chapter deals with this issue and develops such SQ-EEAs for this reason.

Like SM-EEA, the first issue that needs to be addressed for SQ-EEA is to determine how many endmembers, p, are required for SQ-EEA to generate before it is terminated. The virtual dimensionality (VD) developed in Chapter 5 that is used to determine the value of p for SM-EEAs in Chapter 7 is also applicable to SQ-EEAs. Once the p is determined, the second issue is to design an algorithm that can search endmembers one by one sequentially. Four types of SQ-EEAs are of particular interest and will be presented in this chapter.

The first type of SQ-EEAs is orthogonal projection (OP)-based EEAs. A representative is vertex component analysis (VCA) that has recently developed by Nascimento and Dias (2005). This implements a similar idea to that used in the MVT and CCA to find a maximal projection convex hull, but has two different aspects. One is that VCA is an SQ-EEA, not an SM-EEA, as MVT and CCA are. The other is that VCA performs OP to find endmembers instead of finding the maximal simplex volume, as performed by MVT and CCA. Because VCA uses OP in the same manner as PPI, developed in Chapter 7, which also uses OP to find endmembers, it can be considered an SQ-EEA counterpart of PPI. Their relationship will be further explored further in Chapter 11. In addition, as a result of its use of a sequential search via OP rather than solving maximal simplex volume optimization, VCA is considered to be among the fastest EEAs that are currently being used in the literature. However, its performance is generally not as good as what we expect due to the fact that finding maximal OP does not necessarily yield a convex hull with maximal-volume.

A second type of SQ-EEAs of interest is sequential versions of IN-FINDR, as proposed in Chapter 7. One is called successive IN-FINDR (SC IN-FINDR) presented in Section 7.2.3. Since IN-FINDR repeatedly searches for a set of new *p* vertices simultaneously by using two loops until it finds a *p*-vertex simplex with maximal volume, it can be viewed as simultaneous N-FINDR (SM-N-FINDR). So, if the number *p* is large, IN-FINDR becomes very slow. In this case, SC IN-FINDR can be used instead to reduce computing time. The other is called the simplex growing algorithm (SGA) developed by Chang et al. (2006), which finds a desired *p*-dimensional simplex with maximal volume by gradually growing simplexes with maximal volumes vertex by vertex. In other words, instead of directly finding a *p*-vertex simplex with maximal volume as N-FINDR and its variants do, it first finds a two-vertex simplex with largest volume from which it begins to grow new simplexes with largest volumes by increasing vertices from 2 to *p*. Since it generates desired endmembers one by one by a simplex growing process, SGA is indeed an SQ-EEA. Using such a simplex growing implementation, SGA only has to find one endmember at a time until it reaches a desired number of endmembers, *p*. This is completely different from IN-FINDR that replaces vertices of simplexes in the outer loop with a complete new set of vertices repeatedly obtained from its inner loop. Like VCA, SGA is also among the fastest EEAs due to its use of a sequential search process. However, since the OP used in VCA requires less amount of computing time than the calculation of a simplex volume used by SGA, VCA performs faster than SGA in computation. Nevertheless, it should be noted that a faster EEA is not necessarily a better EEA. As demonstrated by experiments, SGA actually performed better than VCA in terms of endmember extraction performance at the expense of higher computation.

The third type of SQ-EEAs is least-squares error (LSE)-based EEAs that include automatic target generation process EEA (ATGP-EEA) originated by Ren and Chang (2003) which implements successive orthogonal subspace projections (OSPs) to find endmembers, unsupervised nonnegativity least-squares EEA (UNCLS-EEA) derived by Chang and Heinz (2000) that uses the abundance nonnegativity constraint to find successive endmembers, unsupervised fully constrained least-squares EEA (UFCLS-EEA) developed by Heinze and Chang (2001) that is derived from FCLS-EEA in Chapter 7, iterative error analysis-EEA (IEA-EEA) proposed by Neville et al. (1999) that is essentially identical to UFCLS-EEA in the sense that both use a full abundance constrained spectral unmixing technique to find endmembers.

Finally, the fourth type of SQ-EEAs is projection pursuit (PP)-based EEAs that include high-order statistics (HOS)-based EEAs and independent component analysis (ICA)-based EEAs; both of them have no SM-EEA counterparts in Chapter 7 but can be derived from dimensionality reduction by transform as discussed in Chapter 6. SQ-EEAs of this type produce a sequence of successive components characterized by projection indices, such as skewness specified by the third-order statistics, kurtosis described by the fourth-order statistics, *k*th moment, statistical independence measured by mutual information so that each component can be used to extract a particular endmember. The idea behind PP-based EEAs is to assume that endmembers can be characterized by statistics of high orders due to their rarity, small sample pool, and occurrence with low probabilities. This rationale is based on several observations. First, since distinct endmembers represent different spectral classes, their statistical dependency must be least correlated. Second, because the nature of endmembers is of pure signatures the probability of occurrence of endmembers is generally low. Third, when endmembers do occur, the spatial extent of their presence is usually very limited and small. Of particular interest among HOS-based EEAs is the ICA (Hyvarinen et al., 2001), which is a blind source separation technique that unmixes a linear mixture of statistically independent signal sources. If the signal sources to be unmixed are interpreted as sources of pure signatures, its applicability to endmember extraction seems natural and justifiable (Wang and Chang, 2006b).

8.2 Successive N-FINDR (SC N-FINDR)

As noted in the *s*-SC IN-FINDR in Chapter 7, a major issue in its implementation is computational cost that is expected to be very high. In order to mitigate this problem, a sequential version of *s*-SC IN-FINDR developed in Section 7.2.3.3 can be derived by replacing simultaneous *s* endmembers carried out in step 5 of the *s*-SC IN-FINDR with successive *p* endmember replacements to achieve more computational efficiency at the expense of optimality. The resulting sequential version of *p*-SC IN-FINDR is referred to as SuCcessive N-FINDR (SC N-FINDR).

Successive N-FINDR (SC N-FINDR)

1. Preprocessing:
   a. Let *p* be the number of endmembers required to generate.
   b. Apply a DR transform such as MNF to reduce the data dimensionality from *L* to *p*–1, where *L* is the total number of spectral bands.
2. Initialization:
   Let $\{\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}\}$ be a set of initial vectors randomly generated from the data.
3. For $1 \leq j \leq p$, find $\mathbf{e}_j^{(*)}$ which yields the maximum volume of $V(\mathbf{e}_1^{(*)}, \ldots, \mathbf{e}_{j-1}^{(*)}, \mathbf{r}, \mathbf{e}_{j+1}^{(0)}, \ldots, \mathbf{e}_p^{(0)})$ defined by (7.3) over all sample vectors **r**, while fixing other endmembers $\mathbf{e}_i^{(k,*)}$ with $i < j$ and $\mathbf{e}_i^{(0)}$ with $i > j$. That is,

$$\mathbf{e}_j^{(*)} = \max_{\mathbf{r}} V\left(\mathbf{e}_1^{(*)}, \ldots, \mathbf{e}_{j-1}^{(*)}, \mathbf{r}, \mathbf{e}_{j+1}^{(0)}, \ldots, \mathbf{e}_p^{(0)}\right) \tag{8.1}$$

The major difference between *s*-SC IN-FINDR in Section 7.2.3.3 and SC N-FINDR described above is that SC N-FINDR only executes the inner loop indexed by *j* in *s*-SC IN-FINDR without going through *s*-SC IN-FINDR's outer loop indexed by *k*. Specifically, the equations used to generate an endmember that yields the maximal simplex volume, that is, (7.8) in *s*-IN-FINDR and (7.12) in *s*-SC IN-FINDR, are replaced by (8.1). In addition, even in the best case where the original initial endmembers turn out to be final desired endmembers and no replacements are required, (8.1) still needs to be calculated *p* times.

8.3 Simplex Growing Algorithm (SGA)

In this section, we present an SQ-EEA, called SGA developed by Chang et al. (2006), which can be considered another sequential version of SM N-FINDR. Unlike SC N-FINDR, which starts with a randomly generated *p*-vertex simplex and then successively replaces one vertex at a time via (8.1), SGA starts with one vertex and then begins to grow a simplex by one vertex at a time until it reaches *p*.

The key to making SGA work hinges on how to appropriately select new vertices to augment growing simplexes. According to N-FINDR for a given positive integer *p*, a simplex formed by *p* endmembers is the one that produces the maximal volume among all possible simplexes formed by any set of *p* data sample vectors. Using this as a criterion, SGA grows the current *k*-vertex simplex $S(\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(k-1)})$ to a $(k+1)$-vertex simplex $S(\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(k-1)}, \mathbf{e}^{(k)})$ by finding a new

$(k+1)$st vertex $\mathbf{e}^{(k)}$ so that the new $(k+1)$-vertex simplex $S(\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(k-1)}, \mathbf{e}^{(k)})$ produces its volume that is not less than volumes of all possible $(k+1)$-vertex simplexes $S(\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(k-1)}, \mathbf{r})$ augmented by any other data sample vector $\mathbf{r}$. The detailed implementation of the above growing simplex process is summarized as follows.

Simplex Growing Algorithm (SGA)

1. Initialization:
   a. Let p be the number of endmembers to be generated.
   b. There are two ways to generate random initial endmembers for SGA.
      i. Select a data sample vector randomly as an initial endmember $\mathbf{e}^{(0)}$ and set $k = 0$. In this case, SGA is referred to as 1-SGA.
      ii. Select a pair of two data sample vectors ($\mathbf{e}^{(0)}$, $\mathbf{e}^{(1)}$) randomly to form a random degenerate two-dimensional simplex that is a line segment connecting $\mathbf{e}^{(0)}$ and $\mathbf{e}^{(1)}$. Set $k = 1$. In this case, SGA is referred to as 2-SGA.
2. At $k \geq 0$ and for each sample vector $\mathbf{r}$, we calculate $V(\mathbf{e}^{(0)}, \ldots, \mathbf{e}^{(k)}, \mathbf{r})$ defined by

$$V(\mathbf{e}^{(0)}, \ldots, \mathbf{e}^{(k)}, \mathbf{r}) = \frac{\left| det \begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ \mathbf{e}^{(0)} & \mathbf{e}^{(2)} & \cdots & \mathbf{e}^{(k)} & \mathbf{r} \end{bmatrix} \right|}{k!} \tag{8.2}$$

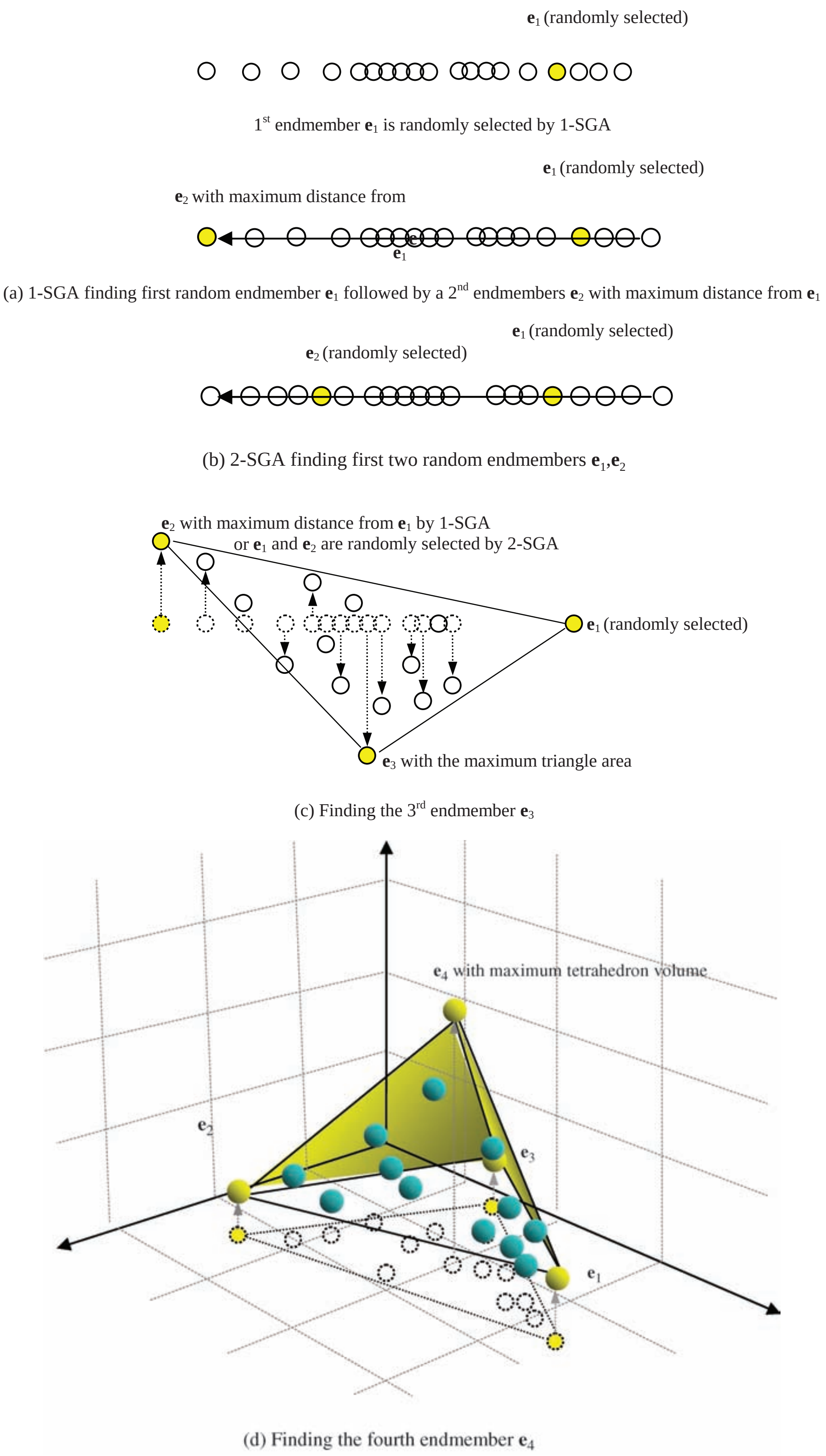
   which is the volume of the simplex specified by vertices $\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(k)}, \mathbf{r}$, denoted by $S(\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(k)}, \mathbf{r})$. Since the matrix det$\begin{bmatrix} 1 & 1 & \cdots & 1 & 1 \\ \mathbf{e}^{(0)} & \mathbf{e}^{(1)} & \cdots & \mathbf{e}^{(k)} & \mathbf{r} \end{bmatrix}$ in (8.2) is not necessarily a square matrix, a dimensionality reduction technique, such as PCA or MNF, is required to reduce the original data dimensionality L to the dimension k.
3. Find $\mathbf{e}^{(k+1)}$ that yields the maximum of (8.3), that is,

$$\mathbf{e}^{(k+1)} = \arg\left\{\max_{\mathbf{r}}\left[V(\mathbf{e}^{(0)}, \ldots, \mathbf{e}^{(k)}, \mathbf{r})\right]\right\}. \tag{8.3}$$

4. Stopping rule:
   If $k = p-1$, then $k \leftarrow k+1$ and go to step 2. Otherwise, the final set of $\{\mathbf{e}^{(1)}, \mathbf{e}^{(2)}, \ldots, \mathbf{e}^{(p)}\}$ is the desired p endmembers.

   Figure 8.1 shows growing simplexes of finding the first four endmembers, $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, and $\mathbf{e}_4$, by 1-SGA and 2-SGA.

Despite that both SGA and SC N-FINDR can be viewed as sequential versions of SM N-FINDR, two fundamental differences are noteworthy. One difference is that SGA grows simplexes via (8.3) starting from a single-vertex simplex compared to SC N-FINDR, which performs in a completely opposite manner in the sense that it actually shrinks simplexes by reducing the number of new vertices to be found via (8.1). More specifically, SC N-FINDR starts off a p-vertex simplex whose vertices are randomly generated. Then, it begins to replace these random vertices one by one with a new vertex that maximizes the volume of the simplex with a reduced number of vertices remaining to be replaced while keeping those new found vertices in tact. The advantage of SGA over SC N-FINDR is that SGA keeps previously found vertices in tact while growing its simplexes as p is increased, whereas SC N-FINDR must restart the entire process once p is changed. In other words, SGA finds its new generated endmembers and works its way up one at a time by growing


$\mathbf{e}_1$ (randomly selected)
1st endmember $\mathbf{e}_1$ is randomly selected by 1-SGA
$\mathbf{e}_1$ (randomly selected)
$\mathbf{e}_2$ with maximum distance from
$\mathbf{e}_1$
$\mathbf{e}_1$ (randomly selected)
$\mathbf{e}_2$ (randomly selected)
$\mathbf{e}_2$ with maximum distance from $\mathbf{e}_1$ by 1-SGA
or $\mathbf{e}_1$ and $\mathbf{e}_2$ are randomly selected by 2-SGA
$\mathbf{e}_1$ (randomly selected)
$\mathbf{e}_3$ with the maximum triangle area
$\mathbf{e}_4$ with maximum tetrahedron volume
$\mathbf{e}_2$
$\mathbf{e}_3$
$\mathbf{e}_1$

(a) 1-SGA finding first random endmember $\mathbf{e}_1$ followed by a 2nd endmembers $\mathbf{e}_2$ with maximum distance from $\mathbf{e}_1$

(b) 2-SGA finding first two random endmembers $\mathbf{e}_1$,$\mathbf{e}_2$

(c) Finding the 3rd endmember $\mathbf{e}_3$

(d) Finding the fourth endmember $\mathbf{e}_4$

Figure 8.1 Growing simplexes of finding the first four endmembers, $\mathbf{e}_1$, $\mathbf{e}_2$, $\mathbf{e}_3$, and $\mathbf{e}_4$, by 1-SGA (a, b, c, d) and (a′, b, c, d) by 2-SGA.

simplexes versus SC N-FINDR, which works its way down one at a time by shrinking simplexes in order to reduce the number of new endmembers to be found. Nevertheless, from a computational complexity viewpoint, both SGA and SC N-FINDR require the same number of comparisons, $p(p-1)/2$ for each data sample vector. Another fundamental difference is the use of random initial conditions. SGA only requires one randomly generated endmember to initialize the algorithm, compared to SC N-FINDR that needs p randomly generated endmembers for initialization. In this case, both SGA and SC N-FINDR share one thing in common that is, their performance is determined by their initial conditions. As a result, judicious selection of initial conditions becomes a key issue that is the major topic of Chapter 9.

8.4 Vertex Component Analysis (VCA)

The VCA developed by Nascimento and Dias (2005) is also designed to reduce costly computational complexity suffered in MVT and CCA by replacing simple volume calculation with OP and growing convex hulls vertex by vertex until it reaches a p-vertex convex hull instead of replacing p-vertex convex hulls all together as MVT and CCA do. Its idea is similar to SGA in the sense that VCA also grows convex hulls one vertex at a time sequentially in succession, but has a major difference from that of SGA in how to find a vertex to grow a convex hull. More specifically, VCA grows convex hulls with maximal orthogonal projections instead of simplexes with maximal volume used by SGA. In other words, VCA appeals for the maximum orthogonal projection as a criterion as PPI does to grow its convex hulls compared to SGA that uses the maximal volume of a simplex as a criterion to grow simplexes as N-FINDR does. In light of this interpretation, VCA can be considered a sequential version of PPI, and SGA can be viewed as a sequential version of N-FINDR. A comparative analysis between VCA and SGA was conducted by Chang et al. (2006) with more details to be discussed in Chapters 9 and 11. The algorithmic implementation of VCA can be described as follows.

Vertex Component Analysis (VCA)

1. Let the number of endmembers to be generated be p. Set a counter to $k = 0$.
2. Perform a dimensionality reduction transform to reduce the original space $\mathbf{X}$ with dimensionality L to a reduced data space $\hat{\mathbf{X}}$ with dimensionality p.
3. Set the initial vector, $\mathbf{e}^{(0)} = (\underbrace{0, 0, \ldots, 1}_{p})$, and let a $p \times p$ auxiliary matrix $\mathbf{A}^{(0)}$ be $\mathbf{A}^{(0)} = \begin{bmatrix}\mathbf{e}^{(0)} & \mathbf{0} & \cdots & \mathbf{0}\end{bmatrix}$.
4. At iteration, $k \geq 0$, generates a Gaussian random vector, $\mathbf{w}^k$, to be used to produce $\mathbf{f}^k$:

$$\mathbf{f}^{(k)} = \left(\left(\mathbf{I} - \mathbf{A}^{(k-1)}\left(\mathbf{A}^{(k-1)}\right)^{\#}\right)\mathbf{w}^k\right) \Big/ \left(\left\|(\mathbf{I} - \mathbf{A}^{(k-1)}\left(\mathbf{A}^{(k-1)}\right)^{\#}\right)\mathbf{w}^k\right\|\right) \qquad (8.4)$$

5. Find $\mathbf{e}^{(k)}$ that maximizes $\mathbf{f}^{(k)^T}\hat{\mathbf{x}}$ over $\hat{\mathbf{x}} \in \hat{\mathbf{X}}$, that is,

$$\mathbf{e}^{(k)} = \arg\left\{\max_{\hat{\mathbf{x}}\in\hat{\mathbf{X}}}\left[|\mathbf{f}^{(k)^T}\hat{\mathbf{x}}|\right]\right\}. \qquad (8.5)$$

6. Use $\mathbf{e}^{(\boldsymbol{k})}$ to replace the kth column of $\mathbf{A}^{(k)}$ to let $\mathbf{A}^{(k)} = \left[\mathbf{e}^{(1)} \cdots \mathbf{e}^{(k)}\mathbf{0} \cdots \mathbf{0}\right]$.
7. If $k = p - 1$, the algorithm is terminated. Otherwise, let $k \leftarrow k + 1$ and go to step 4.

According to the above algorithm, VCA repeatedly performs orthogonal subspace projections to produce a sequence of convex hulls from which convex hulls can gradually grow vertex by vertex to find new vertices. It should be noted that the description of the above algorithm may be slightly different from that proposed by Nascimento and Dias (2005); the main ideas should be the same.

8.5 Linear Spectral Mixture Analysis-Based SQ-EEAs

In Section 7.2.4, a linear spectral mixture analysis (LSMA)-based SM-EEA is developed on the basis of LSE, called FCLS-EEA, where a set of endmembers are found simultaneously by maximizing LSE among all possible subsets that contain the same number of samples. Since all the endmembers must be found simultaneously, full abundance constraints, ASC and ANC, should be imposed on the searching process. However, in a case of successive endmember extraction, there is no need for imposing both ASC and ANC on SQ-EEAs. So, in this section, three second-order statistics-based SQ-EEAs will be presented, all of which are LSE-based spectral unmixing techniques with/without abundance constraints.

The first LSMA-based SQ-EEA of interest is an unconstrained-abundance least-squares algorithm, called ATGP developed by Ren and Chang (2003), which makes use of a sequence of OSP to find target sample vectors successively. In other words, ATGP extracts endmembers from a sequence of nested orthogonal subspaces with reduced dimensionality. It can be considered an unsupervised OSP (UOSP) that extends OSP developed by Harsanyi and Chang (1994) to an unsupervised version of OSP (Chang et al., 1998a; Chang, 2003a). A second LSMA based SQ-EEA is an unsupervised abundance nonnegativity constrained least-squares algorithm, called UNCLS, which is based on nonnegativity constrained least-squares (NCLS) developed by Chang and Heinz (2000). Since it also uses OSP to perform linear unmixing, it can be considered a partially abundance constrained ATGP. A third LSMA-based SQ-EEA is a successive version of FCLS-EEA, called UFCLS-EEA, which imposes an additional abundance sum-to-one constraint on UNCLS. In this case, the UFCLS can also be considered a fully abundance constrained ATGP. An algorithm, recently developed by Neville et al. (1999), called the IEA, also uses LSE as a criterion, and is a fully abundance constrained linear spectral unmixing technique to find endmembers one by one sequentially. It turns out to be that both IEA and UFCLS-EEA are essentially the same technique in the sense that they use an LSE-based fully abundance constrained linear unmixing technique to find endmembers one at a time. So, in the context of OSP, all these LSE-based algorithms are actually OSP-based techniques. In what follows, the implementation of each of these SQ-EEAs is described.

8.5.1 Automatic Target Generation Process-EEA (ATGP-EEA)

ATGP is previously developed to find potential target pixels that can be used to generate a target signature matrix used in an OSP approach (Harsanyi and Chang, 1994; Chang, 2003a, 2003b). It is one of the two processes used in the automatic target detection and classification algorithm developed by Ren and Chang (2003). It repeatedly makes use of an orthogonal subspace projector defined by Harsanyi and Chang (1994) and Chang (2003a)

$$P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\mathbf{U}^{\#} \text{ with } \mathbf{U}^{\#} = \left(\mathbf{U}^{T}\mathbf{U}\right)^{-1}\mathbf{U}^{T} \tag{2.78}$$

to find data sample vectors of interest from the data without prior knowledge regardless of what types of data sample vectors are. It can be described as follows.

Assume that $\mathbf{t}^{(0)}$ is an initial data sample vector. ATGP begins with the initial data sample vector $\mathbf{t}_0$ by applying an orthogonal subspace projector $P^{\perp}_{\mathbf{t}^{(0)}}$, specified by (2.86) with $\mathbf{U}^{(0)} = [\mathbf{t}^{(0)}]$, to all data sample vectors. Then, it finds a data sample vector, denoted by $\mathbf{t}^{(1)}$ with the maximum orthogonal projection in the orthogonal complement space, denoted by $\left\langle \mathbf{t}^{(0)} \right\rangle^{\perp}$ that is orthogonal to the space, $\left\langle \mathbf{t}^{(0)} \right\rangle$ spanned linearly by $\mathbf{t}^{(0)}$. The reason for this selection is that the selected $\mathbf{t}^{(1)}$ has, in general, the most distinct features from $\mathbf{t}^{(0)}$ in the sense of orthogonal projection because $\mathbf{t}^{(1)}$ has the largest magnitude of the projection in $\left\langle \mathbf{t}^{(0)} \right\rangle^{\perp}$ produced by $P^{\perp}_{\mathbf{t}^{(0)}}$. A second data sample vector $\mathbf{t}^{(2)}$ can be found by applying an orthogonal subspace projector $P^{\perp}_{[\mathbf{t}^{(0)}\mathbf{t}^{(1)}]}$ with $\mathbf{U}^{(1)} = [\mathbf{t}^{(0)}\mathbf{t}^{(1)}]$ to the original data set, and a data sample vector that has the maximum orthogonal projection in $\left\langle \mathbf{t}^{(0)}, \mathbf{t}^{(1)} \right\rangle^{\perp}$ is selected as $\mathbf{t}^{(2)}$.

The above procedure is repeated many times to find a third data sample vector $\mathbf{t}^{(3)}$, a fourth data sample vector $\mathbf{t}^{(4)}$, etc., until a certain stopping rule is satisfied. The stopping rule is determined by the number of data sample vectors required to generate, p, which is estimated by the VD. Using p as a stopping criterion, ATGP can be implemented in the following steps.

Algorithm for Automatic Target Generation Process-EEA (ATGP-EEA)

1. Initial condition:
   Let p be the number of endmembers to be generated and $\mathbf{t}^{(0)}$ be a randomly generated initial endmember. Set $k = 0$.
2. At $k \geq 1$ iteration, apply $P^{\perp}_{\mathbf{t}^{(0)}}$ via (2.86) to all data sample vectors $\mathbf{r}$ and find the kth data sample vector $\mathbf{t}^{(k)}$ that has the maximum OP defined by

$$\mathbf{t}^{(k)} = \arg\left\{ \max_{\mathbf{r}} \left[\left(P^{\perp}_{[\mathbf{U}^{(k-1)}\mathbf{t}^{(0)}]} \mathbf{r} \right)^{T} \left(P^{\perp}_{[\mathbf{U}^{(k-1)}\mathbf{t}^{(0)}]} \mathbf{r} \right) \right] \right\} \tag{8.6}$$

   where $\mathbf{U}^{(k-1)} = \left[\mathbf{t}^{(1)}\mathbf{t}^{(2)} \cdots \mathbf{t}^{(k-1)} \right]$ is the data matrix and $\mathbf{U}^{(k-1)} = \emptyset$ if $k - 1 = 0$.
3. Stopping rule:
   If $k - 1 < p$, let $\mathbf{U}^{(k)} = \left[\mathbf{U}^{(k-1)}\mathbf{t}^{(k)} \right] = \left[\mathbf{t}^{(1)}\mathbf{t}^{(2)} \cdots \mathbf{t}^{(k)} \right]$ be the kth data matrix, go to step 2. Otherwise, continue.
4. At this stage, ATGP is terminated. At this point, the data matrix is $\mathbf{U}^{(p-1)}$, which contains $p - 1$ data sample vectors as its column vectors, that do not include the initial vector $\mathbf{t}^{(0)}$.

When ATGP is terminated, the final set of data sample vectors produced by ATGP at step 4 is the desired set of endmembers $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p\}$ that comprises p data sample vectors, $\{\mathbf{t}^{(0)}, \mathbf{t}^{(1)}, \mathbf{t}^{(2)}, \ldots, \mathbf{t}^{(p-1)}\} = \{\mathbf{t}^{(0)}\} \cup \{\mathbf{t}^{(1)}, \mathbf{t}^{(2)}, \ldots, \mathbf{t}^{(p-1)}\} = \{\mathbf{t}^{(0)}\} \cup \mathbf{U}^{(p-1)}$, found by repeatedly using (8.6).

8.5.2 Unsupervised Nonnegativity Constrained Least-Squares-EEA (UNCLS-EEA)

The UNCLS-EEA uses the NCLS method developed by Chang and Heinz (2000) for generating a set of potential data sample vectors, which can be considered endmembers. It first randomly picks an initial data sample vector denoted by $\mathbf{t}^{(0)}$. Then, it assumes that all other data sample vectors are pure data sample vectors consisting of $\mathbf{t}^{(0)}$ with 100% abundance. Of course, this is, in general, not true. Therefore, it next finds a data sample vector that has the largest LSE from the $\mathbf{t}^{(0)}$, and selects it as a first data sample vector denoted by $\mathbf{t}^{(1)}$. Because the LSE between $\mathbf{t}^{(0)}$ and $\mathbf{t}^{(1)}$ is the largest, it can be expected that $\mathbf{t}^{(1)}$ is most distinct from $\mathbf{t}^{(0)}$. NCLS is then used to estimate the abundance fractions for $\mathbf{t}^{(0)}$ and $\mathbf{t}^{(1)}$, denoted by $\hat{\alpha}_0^{(1)}(\mathbf{r})$ and $\hat{\alpha}_1^{(1)}(\mathbf{r})$, for each data sample vector $\mathbf{r}$, respectively. Here, $\mathbf{r}$ is included in the estimated abundance fractions $\hat{\alpha}_0^{(1)}(\mathbf{r})$ and $\hat{\alpha}_1^{(1)}(\mathbf{r})$ to emphasize that $\hat{\alpha}_0^{(1)}(\mathbf{r})$ and $\hat{\alpha}_1^{(1)}(\mathbf{r})$ are the functions of $\mathbf{r}$ and vary with $\mathbf{r}$. The superscript indicates the number of iterations already executed. Now, we find an optimal constrained linear mixture of $\mathbf{t}_0$ and $\mathbf{t}_1$, $\hat{\alpha}_0^{(1)}(\mathbf{r})\mathbf{t}^{(0)} + \hat{\alpha}_1^{(1)}(\mathbf{r})\mathbf{t}^{(1)}$, to approximate the $\mathbf{r}$. Once again, it calculates the LSE between $\mathbf{r}$ and its estimated linear mixture $\hat{\alpha}_0^{(1)}(\mathbf{r})\mathbf{t}^{(0)} + \hat{\alpha}_1^{(1)}(\mathbf{r})\mathbf{t}^{(1)}$ for all data sample vectors $\mathbf{r}$. A pixel that yields the largest LSE from its estimated linear mixture will be selected as a second data sample vector $\mathbf{t}^{(2)}$. As expected, such a selected data sample vector has the largest OP to the space linearly spanned by $\mathbf{t}^{(0)}$ and $\mathbf{t}^{(1)}$. The same procedure of using the NCLS algorithm is repeated until the number of data sample vectors reaches the desired number of endmembers, p, which is preset in advance. The above-outlined procedure is called unsupervised NCLS (UNCLS)-EEA, which can be summarized as follows.

Algorithm for UNCLS-EEA

1. Initial condition:
   Let p be the number of endmembers to be generated and $\mathbf{t}_0$ be a randomly generated initial endmember. Set $k = 0$.
2. Let $k \leftarrow k + 1$ and find $\mathbf{t}^{(k)} = \arg\{\max_{\mathbf{r}}[\mathrm{LSE}^{(k)}(\mathbf{r})]\}$, where the kth least-squares error $\mathrm{LSE}^{(k)}(\mathbf{r})$ is defined by

$$\mathrm{LSE}^{(k)}(\mathbf{r}) = \left(\mathbf{r} - \left[\sum\nolimits_{i=0}^{k} \hat{\alpha}_i^{(k)}(\mathbf{r})\mathbf{t}^{(i)}\right]\right)^T \left(\mathbf{r} - \left[\sum\nolimits_{i=0}^{k} \hat{\alpha}_i^{(k)}(\mathbf{r})\mathbf{t}^{(i)}\right]\right) \tag{8.7}$$

3. Apply the NCLS method with the endmember matrix $\mathbf{M}^{(k)} = \left[\mathbf{t}^{(0)}\mathbf{t}^{(1)} \cdots \mathbf{t}^{(k-1)}\right]$ to estimate the abundance fraction of $\mathbf{t}^{(0)}, \mathbf{t}^{(1)}, \ldots, \mathbf{t}^{(k-1)}$, $\hat{\alpha}_0^{(k)}(\mathbf{r}), \hat{\alpha}_1^{(k)}(\mathbf{r}), \ldots, \hat{\alpha}_{k-1}^{(k)}(\mathbf{r})$. If $k = p$, the algorithm is terminated; otherwise, go to step 2.

When the UNCLS algorithm is terminated at step 3, the final generated set $\{\mathbf{t}^{(0)}, \mathbf{t}^{(1)}, \mathbf{t}^{(2)}, \ldots, \mathbf{t}^{(p-1)}\} = \{\mathbf{t}^{(0)}\} \cup \{\mathbf{t}^{(1)}, \mathbf{t}^{(2)}, \ldots, \mathbf{t}^{(p-1)}\}$ is the desired endmembers $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p\}$.

8.5.3 Unsupervised Fully Constrained Least-Squares-EEA (UFCLS-EEA)

The UFCLS-EEA presented in the following is identical to UNCLS-EEA described in Section 8.5.2 with the exception that NCLS used in UNCLS-EEA is replaced by the FCLS method developed by Heinz and Chang (2001).

Algorithm for Unsupervised FCLS (UFCLS)-EEA

1. Initial condition:
   Let p be the number of endmembers to be generated and $\mathbf{t}^{(0)}$ be a randomly generated initial endmember. Set $k = 0$.
2. Let $k \leftarrow k + 1$ and find $\mathbf{t}^{(k)} = \arg\{\max_{\mathbf{r}}[\mathrm{LSE}^{(k)}(\mathbf{r})]\}$, where the kth least-squares error $\mathrm{LSE}^{(k)}(\mathbf{r})$ is defined in (8.5).
3. Apply FCLS with the endmember matrix $\mathbf{M}^{(k)} = \left[\mathbf{t}^{(0)}\mathbf{t}^{(1)} \cdots \mathbf{t}^{(k-1)}\right]$ to estimate the abundance fraction of $\mathbf{t}^{(0)}, \mathbf{t}^{(1)}, \ldots, \mathbf{t}^{(k-1)}$, $\hat{\alpha}_0^{(n)}(\mathbf{r}), \hat{\alpha}_1^{(n)}(\mathbf{r}), \ldots, \hat{\alpha}_{k-1}^{(n)}(\mathbf{r})$. If $k = p$, the algorithm is terminated; otherwise, go to step 2.

8.5.4 Iterative Error Analysis-EEA (IEA-EEA)

The IEA was originally proposed by Neville et al. (1999) for endmember extraction. In analogy with UFCLS-EEA, it also makes use of constrained linear spectral unmixing to search for possible endmembers. In addition, it does not produce all endmembers simultaneously as an SM-EEA does. It calculates the sample mean and uses it to initialize the algorithm. Then, it repeatedly performs constrained linear spectral unmixing procedures for producing a sequence of data sample vectors in succession, which are considered endmembers by IEA. Interestingly, there are major differences between IEA and UFCLS-EEA. One difference is the initial data sample vector generated by the algorithms. While the IEA calculates the sample mean vector for initialization, UFCLS-EEA finds a pixel with the largest vector length to be used as its initial pixel to start the algorithm. Another difference is that UFCLS-EEA generates a new data sample vector that has the largest least-squares error in terms of a fully constrained least-squares linear mixture described by (8.5). On the contrary, IEA uses an error image resulting from linear constrained linear unmixing after each spectral unmixing and finds the mean of the pixels with the largest least-squares error that are farthest from the previously selected endmembers. As a result, the endmembers sought by UFCLS-EEA are actually pixels, as opposed to signatures found by IEA that produces sample means as endmembers. In the latter case, the IEA-generated signatures are not necessarily image pixels. A third difference is that UFCLS-EEA does not need any prior knowledge, except the knowledge of the number of data sample vectors, p or a prescribed error threshold to terminate the algorithm. As for IEA, three parameters are required to be determined *a priori*, p, the desired number of endmembers, ${N_{R(\theta)}}^{(k)}$, the number of pixels in $R^{(k)}(\theta)$, where $R^{(k)}(\theta)$ is the set of data sample vectors with the largest errors in an error data set $E^{(k)}$ after the kth spectral unmixing and θ is a spectral angle to be used to find data sample vectors that will be averaged to generate an endmember signature. Since no specific constrained spectral unmixing method was mentioned by Neville et al. (1999), the FCLS developed by Heinz and Chang (2001) will be used in IEA for our version of interpreting IEA.

Algorithm for IEA-EEA

1. Set values for three parameters p, R, and θ.
2. Initialization:
   Generate randomly an initial endmember, denoted by $\mathbf{t}^{(0)}$.
3. Perform constrained linear spectral unmixing of $\mathbf{t}^{(0)}$ to find an error data set $E^{(0)}$.
4. For $k \geq 0$, find a set of data sample vectors in $R^{(i)}$ that are within the spectral angle θ and farthest from the obtained kth error data set $E^{(i)}$ in terms of the Euclidean distance (i.e., vector length). Finally, calculate the average of data sample vectors in $R^{(k)}(\theta)$ and use it as the $(k+1)$ st endmember, denoted by $\mathbf{t}^{(k+1)}$.
5. If the used stopping rule is satisfied, the algorithm is terminated. Otherwise, perform constrained linear spectral unmixing on the kth endmember set, $E^{(k)} = \{\mathbf{t}^{(0)}, \mathbf{t}^{(1)}, \ldots, \mathbf{t}^{(k-1)}\}$, and find its error data set $E^{(k)}$. Let $k \leftarrow k+1$, and go to step 4. It should be noted that the stopping rule used in this step can be implemented in two ways. One way is to predetermine the number of endmember, p, in advance. Another way is to predetermine the unmixing error.

As noted, since the endmembers generated by the IEA-EEA are the averaged value of a set of data sample vectors in $R^{(k)}(\theta)$, they are not real data sample vectors in the data. In order to make a fair comparison with other EEAs, we set ${N_R}^{(k)} = 1$ and $\theta = 0$ for our experiments. In this case, the IEA-generated endmembers are actually real data sample vectors in the data.

8.6 High-Order Statistics-Based SQ-EEAS

In Section 7.3, a second-order statistics SM-EEA, SPCA-EEA, is derived to find a set of endmembers that yield the least statistical correlation. However, there are no HOS SM-EEAs that are similar to SPCA-EEA developed in Chapter 7. The reason for this is that no analytic form can be derived for HOS-EEAs in the same way as SPCA-EEA that solves a characteristic polynomial equation to find all eigenvalues simultaneously. In this case, instead of solving a known equation such as the characteristic polynomial equation, HOS-based EEAs must appeal for an algorithm that allows one to find projection vectors similar to eigenvectors found by SPCA-EEA through eigenvalues and each of such projection vectors can only be found one at a time. Then, each projection vector produces an HOS component from which an endmember can be extracted. An EEA design, based on this approach, is called an HOS-based SQ-EEA.

More specifically, we assume that the ith HOS component, denoted by $\mathrm{C}_i^{\mathrm{HOS}}$, can be described by a random variable ζ_i with values taken by the gray level value of the nth pixel in the component $\mathrm{C}_i^{\mathrm{HOS}}$, denoted by z_n^i. Therefore, criteria used to generate various HOS components in Section 6.3 are also used here to generate HOS components for endmember extraction as follows. However, it should be noted that since only HOS is of interest, a process, called sphering discussed in Section 6.3, should be first applied as a preprocessing to remove second-order statistics prior to finding HOS components.

8.6.1 Third-Order Statistics-Based SQ-EEA

The first HOS is the third-order statistics, also known as skewness, which measures asymmetry of a probability distribution. The third-order statistics component is generated by a projection vector $\mathbf{w}^*$, which solves the following optimization:

$$\begin{aligned}&\max_{\mathbf{w}}\left\{(1/N)\textstyle\sum_{i=1}^{N} z_i^3\right\}\\ &=\max_{\mathbf{w}}\left\{(1/N)\textstyle\sum_{i=1}^{N} \mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\right\}\end{aligned}\quad \text{subject to } \mathbf{w}^T\mathbf{w}=1 \tag{8.8}$$

which is equivalent to solving the following equation:

$$\left(E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\tilde{\mathbf{r}}_i^T\right]-\lambda'\mathbf{I}\right)\mathbf{w}=0 \tag{8.9}$$

by finding the eigenvalue λ' of the matrix $E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\tilde{\mathbf{r}}_i^T\right]$ and its corresponding eigenvector $\mathbf{w}^*$.

8.6.2 Fourth-Order Statistics-Based SQ-EEA

The second HOS is the fourth-order statistics, also known as kurtosis, that measures the flatness of a probability distribution. Similarly to (8.8), the fourth-order statistics component is generated by a projection vector $\mathbf{w}^*$ that solves the following optimization:

$$\begin{aligned}&\max_{\mathbf{w}}\left\{(1/N)\textstyle\sum_{i=1}^{N} z_i^4\right\}\\ &=\max_{\mathbf{w}}\left\{(1/N)\textstyle\sum_{i=1}^{N} \mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\right\}\end{aligned}\quad \text{subject to } \mathbf{w}^T\mathbf{w}=1 \tag{8.10}$$

which is equivalent to solving the following equation:

$$\left(E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\mathbf{w}^T\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\right]-\lambda'\mathbf{I}\right)\mathbf{w}=0 \tag{8.11}$$

by finding the eigenvalue λ' of the matrix $E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}\tilde{\mathbf{r}}_i^T\right]$ and its corresponding eigenvector $\mathbf{w}^*$.

8.6.3 Criterion for kth Moment-Based SQ-EEA

In analogy with (8.9) and (8.11), HOS can be extended to the *k*th-order statistics: *k*th moment by solving the following eigenproblem:

$$\left(E\left[\tilde{\mathbf{r}}_i\left(\tilde{\mathbf{r}}_i^T\mathbf{w}\right)^{k-2}\tilde{\mathbf{r}}_i^T\right]-\lambda'\mathbf{I}\right)\mathbf{w}=0 \tag{8.12}$$

8.6.4 Algorithm for Finding Projection Vectors

It should be noted that a single projection vector $\mathbf{w}^*$, which solves (8.9) for skewness, (8.11) for kurtosis, or (8.12) for the *k*th moment, represents only one component. In order to continuously generate new components, a sequence of projections must be performed. In this case, when a projector vector $\mathbf{w}^*$ is found, the de-correlated data $\tilde{\mathbf{X}}$ are mapped into the linear subspace $\langle\mathbf{w}^*\rangle^{\perp}$ orthogonal to $\langle\mathbf{w}^*\rangle$ that is the space spanned linearly by $\mathbf{w}^*$. The next projection vector $\mathbf{w}^*$ is then found by solving (8.9), (8.11), or (8.12) in the space $\langle\mathbf{w}^*\rangle^{\perp}$. The same procedure is then continued until a stop criterion, that is, the predetermined number of projections required to be generated, is satisfied. An algorithm for finding a sequence of projection vectors is called the projection vector generation algorithm (PVGA) and can be described as follows.

Projection Vector Generation Algorithm (PVGA)

1. Initially, sphere the original data set $\mathbf{X}$ via (6.36)–(6.38). The resulting data set is denoted by $\tilde{\mathbf{X}}$.
2. Find the first projection vector $\mathbf{w}_1^*$ by solving (8.9) or (8.11) or (8.12) depending on which criterion is used, skewness or kurtosis, or the *k*th moment.
3. Using the obtained $\mathbf{w}_1^*$, generate the first projection image $\tilde{\mathbf{Z}}^1=\left(\mathbf{w}_1^*\right)^T\tilde{\mathbf{X}}=\{\tilde{\mathbf{z}}_i^1|\tilde{\mathbf{z}}_i^1=\left(\mathbf{w}_1^*\right)^T\tilde{\mathbf{r}}_i\}$, which can be used to detect the first endmember.
4. Apply the OSP specified by $P_{\mathbf{w}_1}^{\perp}=\mathbf{I}-\mathbf{w}_1(\mathbf{w}_1^T\mathbf{w}_1)^{-1}\mathbf{w}_1^T$ to the data set $\tilde{\mathbf{X}}$ to produce the first OSP-projected data set denoted by $\tilde{\mathbf{X}}^1$, $\tilde{\mathbf{X}}^1=P_{\mathbf{w}_1}^{\perp}\tilde{\mathbf{X}}$.
5. Use the data set $\tilde{\mathbf{X}}^1$ and find the second projection vector $\mathbf{w}_2^*$ by solving (8.9), (8.11), or (8.12) that depends on which criterion is used, skewness or kurtosis, or the *k*th moment.
6. Apply $P_{\mathbf{w}_2}^{\perp}=\mathbf{I}-\mathbf{w}_2(\mathbf{w}_2^T\mathbf{w}_2)^{-1}\mathbf{w}_2^T$ to the data set $\tilde{\mathbf{X}}^1$ to produce the second OSP-projected data set denoted by $\tilde{\mathbf{X}}^2$, $\tilde{\mathbf{X}}^2=P_{\mathbf{w}_2}^{\perp}\tilde{\mathbf{X}}^1$, which can be used to produce the third projection vector $\mathbf{w}_3^*$ by solving (8.9), (8.11), or (8.12). Or, equivalently, we define a matrix projection matrix $\mathbf{W}^2=[\mathbf{w}_1\mathbf{w}_2]$ and apply $P_{\mathbf{W}^2}^{\perp}=\mathbf{I}-\mathbf{W}^2\left(\left(\mathbf{W}^2\right)^T\mathbf{W}^2\right)^{-1}(\mathbf{W}^2)^T$ to the original sphered data set $\tilde{\mathbf{X}}$ to obtain $\tilde{\mathbf{X}}^2=P_{\mathbf{W}^2}^{\perp}\tilde{\mathbf{X}}$.
7. Repeat the procedure of steps 5 and 6 many times to produce $\mathbf{w}_3^*,\ldots,\mathbf{w}_k^*$ until a stopping criterion is met. It should be noted that a stopping criterion can either be a predetermined number of projection vectors required to be generated or be a predetermined threshold for the difference between two consecutive projection vectors.

It should be noted that the implementation of step 2 in PVGA is not trivial. In order to solve (8.9), (8.11), or (8.12) for the optimal projection vector $\mathbf{w}_1^*$, the following iterative procedure is proposed to execute step 2, where only the criterion (8.9) for skewness is used for illustration. Similarly, the same implementation can also be used for kurtosis and the kth moment with (8.9) replaced by (8.11) and (8.12), respectively.

Implementation of step 2 to execute (8.9) for skewness:

2(a) Initialize a random projector $\mathbf{w}_1^{(0)}$ and set $k = 0$

2(b) Calculate the matrix $E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}_1^{(k)}\tilde{\mathbf{r}}_i^T\right]$ and find an eigenvector $\mathbf{v}_1^{(k)}$ corresponding to the largest magnitude of eigenvalues of the matrix $E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\mathbf{w}_1^{(k)}\tilde{\mathbf{r}}_i^T\right]$.

2(c) If the Euclidean distance $\left\|\mathbf{w}_1^{(k)} - \mathbf{v}_1^{(k)}\right\| > \varepsilon$ and $\left\|\mathbf{w}_1^{(k)} + \mathbf{v}_1^{(k)}\right\| > \varepsilon$, then let $\mathbf{w}_1^{(k+1)} = \mathbf{v}_1^{(k)}$ and $k \leftarrow k + 1$, go to step 2(b). Otherwise, $\mathbf{w}_1^{(k)}$ is the desired projector $\mathbf{w}_1^{(*)}$. Let $\mathbf{w}_1^* = \mathbf{w}_1^{(k)}$ and return to step 3 in the PVGA.

8.6.5 *ICA-Based SQ-EEA*

This section presents an ICA-based EEA, called ICA-EEA. The idea is to select the first p prioritized ICs for endmember extraction and to further use the same selected p prioritized ICs for abundance quantification for all image pixels. In other words, each IC represents a specific class of data sample vectors extracted from the image data. Thus, it can be used to serve as an abundance fraction map for this particular class. Most EEAs extract all desired endmembers in a single map to show their spatial presence in the image in a similar way that a classification algorithm produces a class map. The following proposed ICA-based endmember extraction algorithm performs otherwise. It uses the FastICA-generated ICs that separate all extracted endmember pixels in individual components so that no two endmember pixels that represent two different classes will be present in the same IC component.

ICA-Based Endmember Extraction Algorithm (ICA-EEA)

1. Let p be the number of ICs needed to be generated.
2. Implement either HOS-ICPA or ID-ICPA to find first p prioritized ICs in accordance with their priority scores.
3. For each of the selected p FastICA-generated IC images, find a pixel with the maximal absolute value, which is referred to as the endmember pixel. The spectral signature of such a found pixel is then selected as an endmember.
4. The p endmember pixels produced in step 3, denoted by $\{\mathbf{e}_j\}_{j=1}^p$, are the final set of endmember pixels whose spectral signatures are our desirable endmembers.

It should be noted that there may have been more than one endmember pixel extracted in a single IC, in which case all the extracted endmember pixels are considered to be in the same class, which will be shown in our experiments.

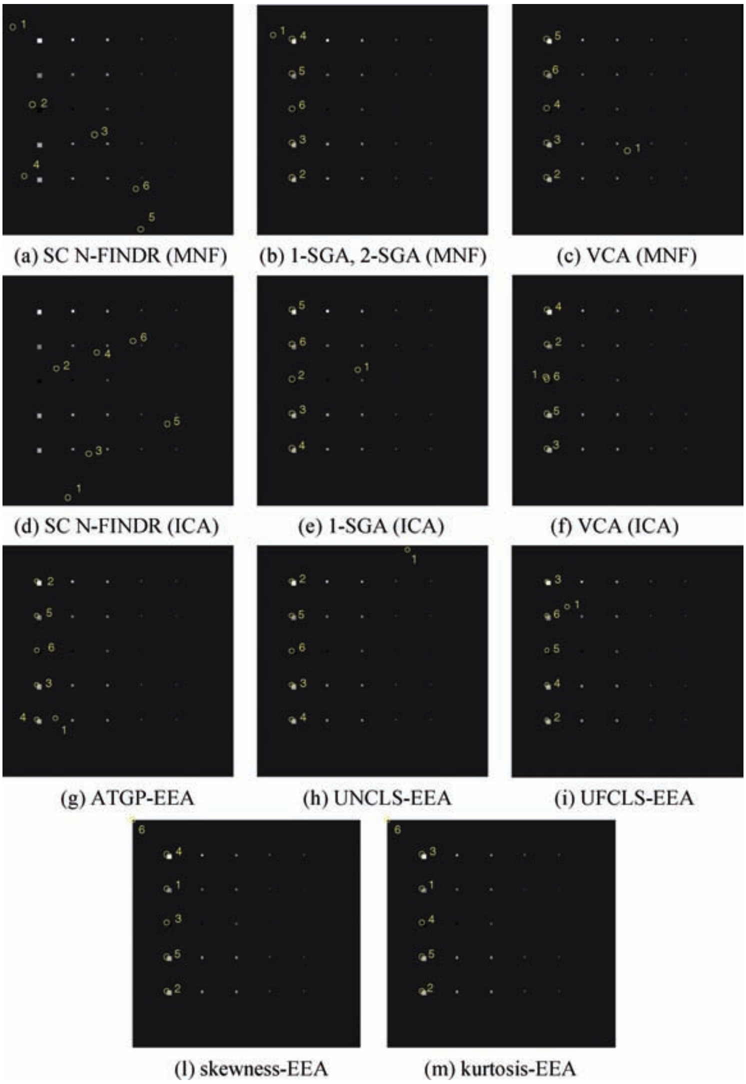
8.7 Experiments

The experiments conducted in this section follow the same experiments conducted in Section 7.5 so that a comparative analysis on performance evaluation between SM-EEAs in Chapter 7 and SQ-EEAs in this chapter can be performed by comparing their respective experimental results, where the same six synthetic image-based scenarios, Cuprite image and HYDICE image are used for experiments. Four categories of SQ-EEAs are selected for performance evaluation, (1) OP-based EEA, VCA, (2) simplex-based EEA, SGA, SC N-FINDR, (3) second-order statistics least-squares error-based EEAs, ATGP-EEA, UNCLS-EEA, UFCLS-EEA, and (4) high-order statistics-based EEAs, HOS-EEAs (skewness-EEA, kurtosis-EEA), ICA-EEA. So, a total of nine SQ-EEAs, VCA, SC N-FINDR, SGA, ATGP-EEA, UNCLS-EEA, UFCLS-EEA, skewness-EEA, kurtosis-EEA, and ICA-EEA are actually studied in this section for comparative analysis. Several comments on the selection of these eight SQ-EEAs are worth noting.

1. SC N-FINDR is derived as an SQ-EEA version of N-FINDR in Chapter 7.
2. SGA can also be considered as another SQ-EEA version of N-FINDR in Chapter 5, which has two versions, 1-SGA and 2-SGA, depending on one or two endmembers randomly generated for initialization.
3. VCA can be considered as an SQ-EEA of PPI in Chapter 7.
4. As will also be shown in Chapter 11, PPI, VCA, and ATGP-EEA are actually closely related through OP. With this interpretation, VCA can be considered as an SQ-EEA version of PPI and a random version of ATGP-EEA.
5. UFCLS-EEA can be considered as an SQ-EEA version of FCLS-EEA. Since IEA-EEA is very similar to UFCLS-EEA, only UFCLS-EEA is selected for evaluation.
6. In comparison with UFCLS-EEA, which imposes full abundance constraints on endmembers, it is interesting to see endmember extraction performance if abundance constraints are not fully implemented, in which case ATGP-EEA is fully unconstrained and UNCLS-EEA is partially constrained by imposing only abundance nonnegativity constraints. So, these three least-squares-based EEAs cover three possible scenarios: unconstrained, partially constrained, and fully constrained cases.
7. Since no significant difference can be gained by statistics higher than 4, only the third- and fourth-order statistics-based EEAs, that is, skewness-EEA, kurtosis-EEA, are considered to represent HOS-EEAs.
8. Finally, for those EEAs such as SC N-FINDR, SGA, and VCA, which require dimensionality reduction, four different transforms, SVD, PCA, MNF, and ICA, can be used to perform DR. Since SVD, PCA, and MNF are second-order statistics-based transforms and perform similarly, only experiments using MNF are presented.

8.7.1 Synthetic Image Experiments

The same six scenarios, TI1, TI2, and TI3 for target implantation (TI) and TE1, TE2, and TE3 for target embeddedness (TE) studied in Section 7.5 were also used for experiments to conduct comparative studies where eight SQ-EEAs, SC N-FINDR, 1-SGA/2-SGA, VCA, ATGP-EEA, UFCLS-EEA, UNCLS-EEA, skewness-EEA, and kurtosis-EEA, were evaluated for performance analysis. For the three SQ-EEAs, SC N-FINDR, SGA, and VCA, MNF and ICA were used to perform DR because MNF is more effective than PCA, in general, in terms of second-order statistics and ICA is the most widely used HOS-based DR transform. It is also noted that SC N-FINDR was used as an SQ N-FINDR due to its simple implementation. Since the number of endmembers to be extracted for six scenarios was estimated to be either $p = 5$ or 6, the dimensionality to be retained after DR was set to $q = 6$. Furthermore, since SQ-EEAs extracted endmembers one after another in a sequential order, numerals were used to indicate the orders that the pixels were extracted as endmembers and the endmembers



(a) SC N-FINDR (MNF) (b) 1-SGA, 2-SGA (MNF) (c) VCA (MNF)

(d) SC N-FINDR (ICA) (e) 1-SGA (ICA) (f) VCA (ICA)

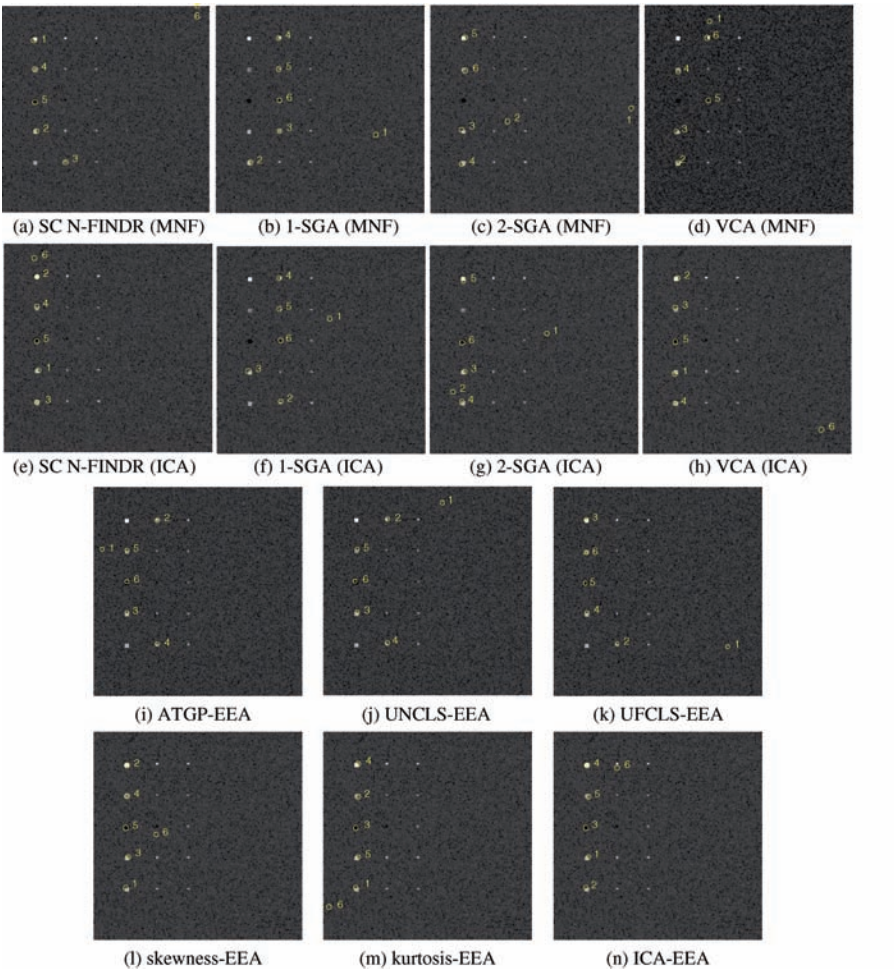
(g) ATGP-EEA (h) UNCLS-EEA (i) UFCLS-EEA

(l) skewness-EEA (m) kurtosis-EEA

Figure 8.2 Results for scenario TI1 with six endmembers extracted by SC N-FINDR, 1-SGA, 2-SGA, VCA, ATGP-EEA, UFCLS-EEA, UNCLS-EEA, skewness-EEA, and kurtosis-EEA using one set of random initial endmembers.

extracted by SQ-EEAs using $p = 5$ were part of endmembers extracted by SQ-EEAs using $p = 6$ by setting $q = 6$. Figures 8.2–8.7 show that with six endmembers extracted by SC N-FINDR, 1-SGA/2-SGA, VCA, ATGP-EEA, UFCLS-EEA, UNCLS-EEA, skewness-EEA, and kurtosis-EEA using one set of random initial endmembers.

Among six scenarios is TI2, which represents the most realistic one for endmember extraction because pure pixels are inserted into the noisy image background. In this particular scenario, all the eight SQ-EEAs except 2-SGA were able to extract five endmembers if $p = 6$. However, when $p = 5$,



(a) SC N-FINDR (MNF) (b) 1-SGA (MNF) (c) 2-SGA (MNF) (d) VCA (MNF)

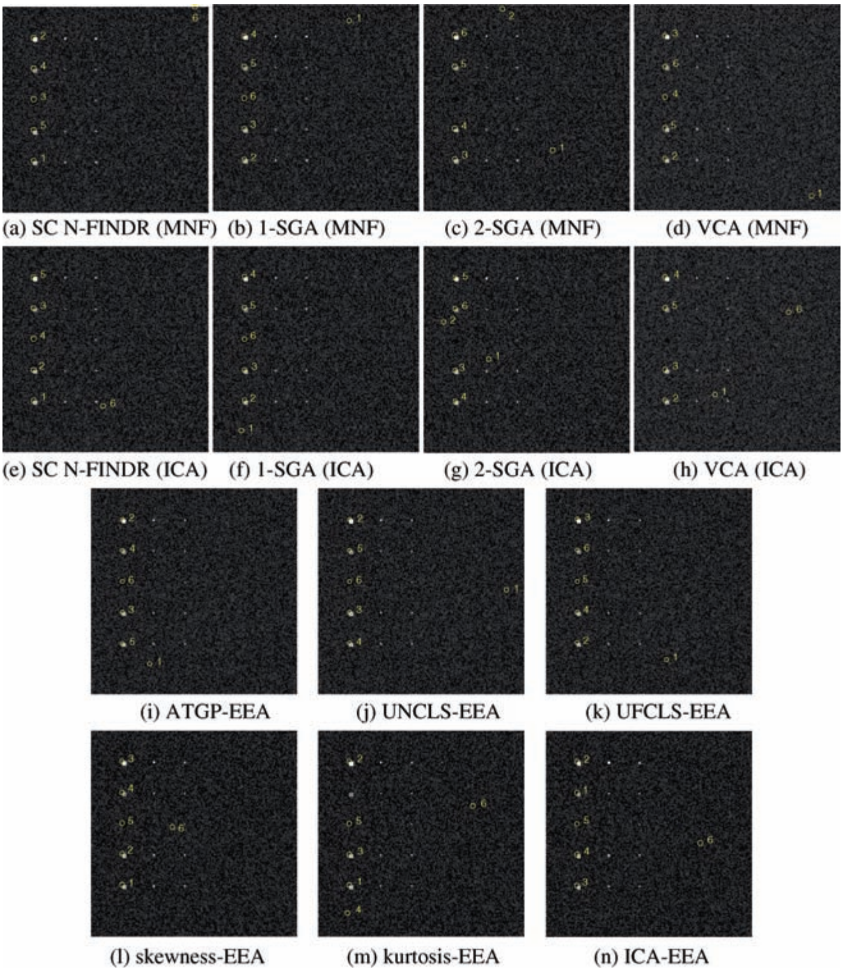
(e) SC N-FINDR (ICA) (f) 1-SGA (ICA) (g) 2-SGA (ICA) (h) VCA (ICA)

(i) ATGP-EEA (j) UNCLS-EEA (k) UFCLS-EEA

(l) skewness-EEA (m) kurtosis-EEA (n) ICA-EEA

Figure 8.3 Results for scenario TI2 with six endmembers extracted by SC N-FINDR, 1-SGA, 2-SGA, VCA, ATGP-EEA, UFCLS-EEA, UNCLS-EEA, skewness-EEA, kurtosis-EEA, and ICA-EEA using one set of random initial endmembers.

only HOS-based EEAs and those SQ-EEAs using ICA except 1-SGA to perform DR could extract all five endmembers in the first five pixels. This implies that HOS characterizes endmembers more effectively than second-order statistics. The same conclusions could also be drawn for TI1, TI3, TE2, and TE3. An interesting observation is scenarios of TI1 and TE1 where no noise is present in the image. In this case, all pixels, including background pixels, are clean and considered to be pure. As a result, SC N-FINDR had difficulty with determining which one was really a pure signature as an endmember. As a matter of fact, all SQ-EEAs using maximal simplex or convex hull volume as a criterion such as SGA or VCA had the same problem compared to other criteria, which did not have such a difficulty with finding endmembers as target of interest.



(a) SC N-FINDR (MNF) (b) 1-SGA (MNF) (c) 2-SGA (MNF) (d) VCA (MNF)

(e) SC N-FINDR (ICA) (f) 1-SGA (ICA) (g) 2-SGA (ICA) (h) VCA (ICA)

(i) ATGP-EEA (j) UNCLS-EEA (k) UFCLS-EEA

(l) skewness-EEA (m) kurtosis-EEA (n) ICA-EEA

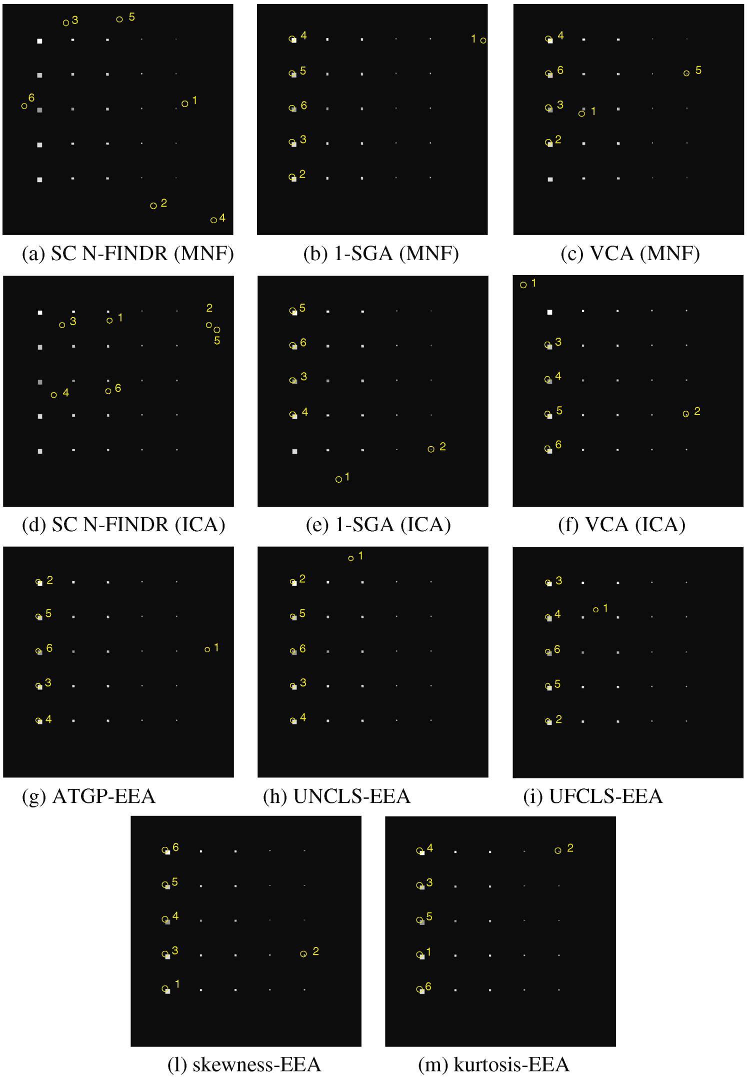
Figure 8.4 Results for scenario TI3 with six endmembers extracted by SC N-FINDR, 1-SGA, 2-SGA, VCA, ATGP-EEA, UFCLS-EEA, UNCLS-EEA, skewness-EEA, kurtosis-EEA, and ICA-EEA using one set of random initial endmembers.

8.7.2 Real Hyperspectral Image Experiments

The same two real hyperspectral image data sets, Cupritre data in Figures 1.12(a) and (b) and the HYDICE image scene in Figures 1.15(a) and (b), were also used for experiments here.

8.7.2.1 Cuprite Data

The value of p estimated by VD for the reflectance data of the Cuprite scene in Figures 1.12 (a) and (b) is 22, which has been used in Chapter 7 to evaluate SM-EEAs. However, based on SSE its estimate is 28. Since SQ-EEAs generate endmembers sequentially, in this case we use a large value of p to generate endmembers so that the first smaller number of endmembers



(a) SC N-FINDR (MNF) (b) 1-SGA (MNF) (c) VCA (MNF)

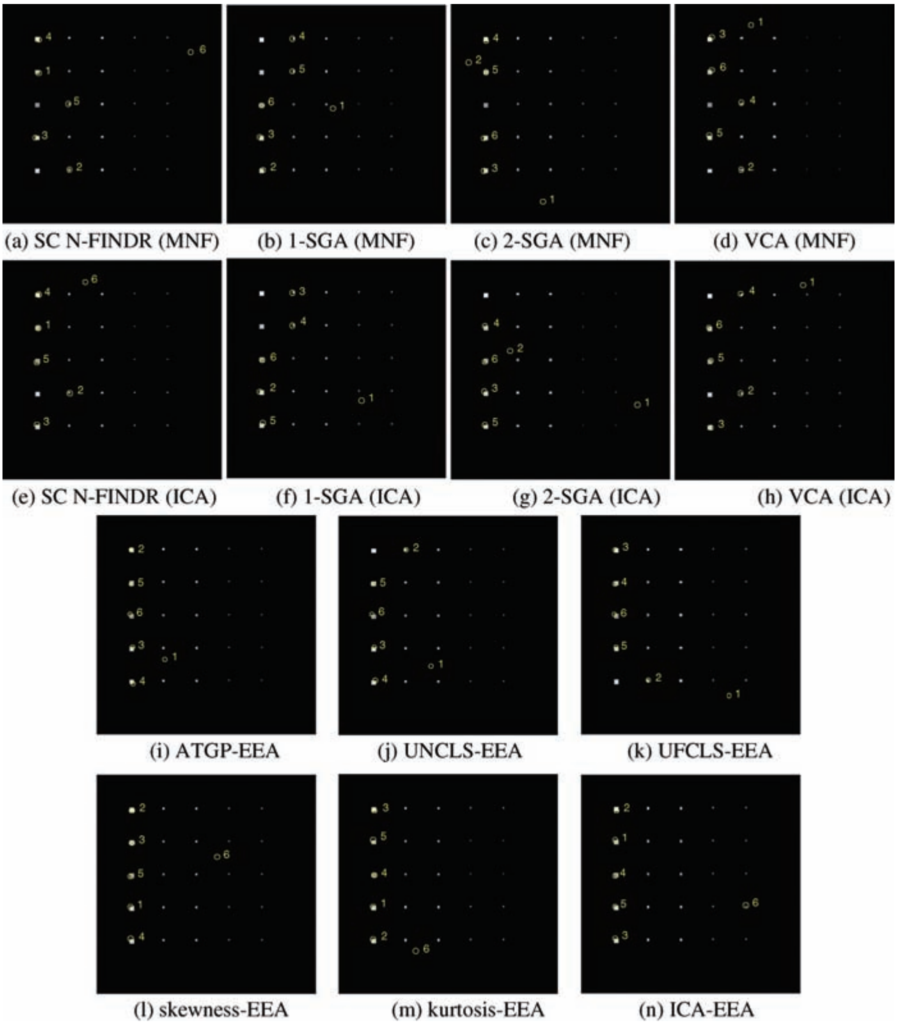
(d) SC N-FINDR (ICA) (e) 1-SGA (ICA) (f) VCA (ICA)

(g) ATGP-EEA (h) UNCLS-EEA (i) UFCLS-EEA

(l) skewness-EEA (m) kurtosis-EEA

Figure 8.5 Results for scenario TE1 with six endmembers extracted by SC N-FINDR, 1-SGA, VCA, ATGP-EEA, UFCLS-EEA, UNCLS-EEA, skewness-EEA, and kurtosis-EEA using one set of random initial endmembers.

would be those generated by a smaller value of p. Figure 8.8 shows $n_{\text{SSE}} = 28$ ($n_{\text{VD}} = 22$) endmembers extracted by eight SQ-EEAs, including two veriosns of SGA, 1-SGA and 2-SGA, where endmember pixels extracted by algorithms are marked by yellow circles, the pixels marked by upper cases of "A, B, C, K, M" are the five ground truth mineral pixels, and the pixels marked by yellow triangles with lower cases of "a, b, c, k, m" are identified by SQ-EEAs corresponding to the five true mineral signatures and are listed in the parenthesis underneath each figure.



(a) SC N-FINDR (MNF) (b) 1-SGA (MNF) (c) 2-SGA (MNF) (d) VCA (MNF)

(e) SC N-FINDR (ICA) (f) 1-SGA (ICA) (g) 2-SGA (ICA) (h) VCA (ICA)

(i) ATGP-EEA (j) UNCLS-EEA (k) UFCLS-EEA

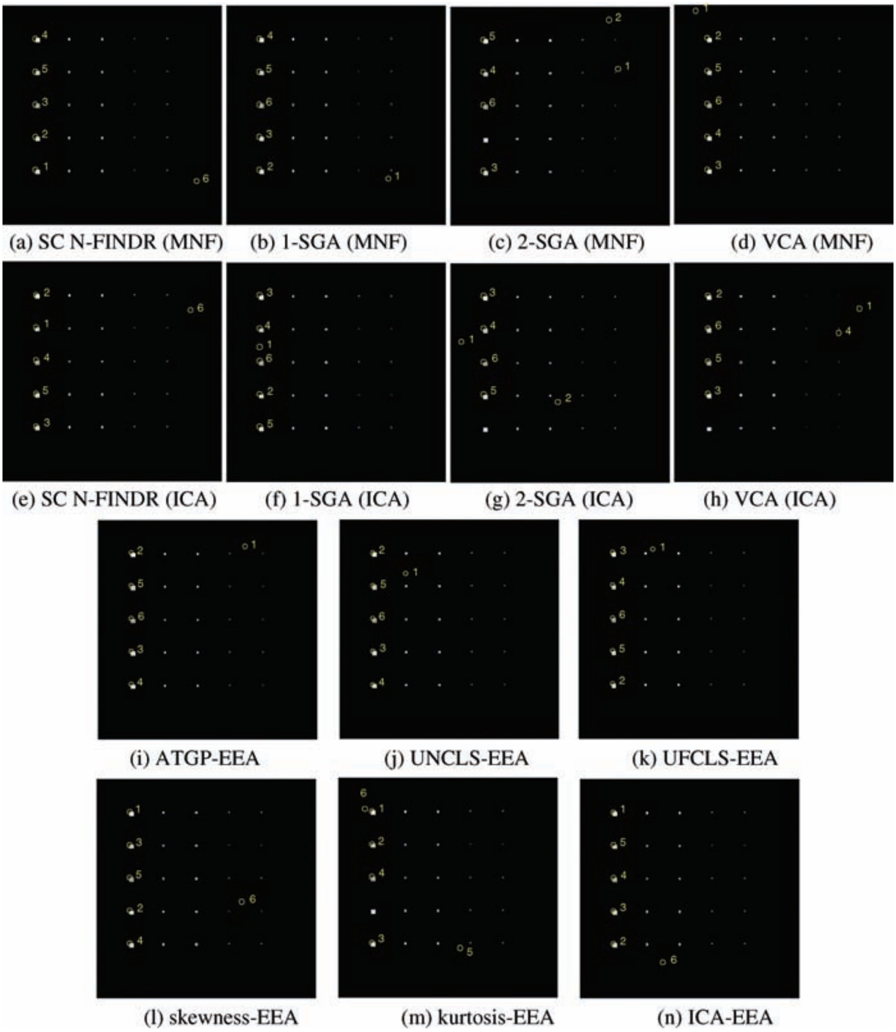
(l) skewness-EEA (m) kurtosis-EEA (n) ICA-EEA

Figure 8.6 Results for scenario TE2 with six endmembers extracted by SC N-FINDR, 1-SGA, 2-SGA, VCA, ATGP-EEA, UFCLS-EEA, UNCLS-EEA, skewness-EEA, kurtosis-EEA, and ICA-EEA using one set of random initial endmembers.

Comparing the results in Figure 8.8 extracted by SQ-EEAs with those in Figure 7.25 extarcted by SM-EEAs, we find that the results were similar and close if an SQ-EEA was modified from its SM-EEA counterpart such as SC N-FINDR. As expected, using HOS-based SQ-EEAs or ICAs to perform DR did improve the performance of endmember extarction. Interestingly, ATGP- and LSE-based SQ-EEAs were shown to perform better than convex geometry-based SQ-EEAs.

8.7.2.2 HYDICE Data

Following the same experiment as conducted for the cuprite data, Figure 8.9 shows 10 endmembers extracted by eight SQ-EEAs including two versions of SGA, 1-SGA and 2-SGA, where the



(a) SC N-FINDR (MNF) (b) 1-SGA (MNF) (c) 2-SGA (MNF) (d) VCA (MNF)

(e) SC N-FINDR (ICA) (f) 1-SGA (ICA) (g) 2-SGA (ICA) (h) VCA (ICA)

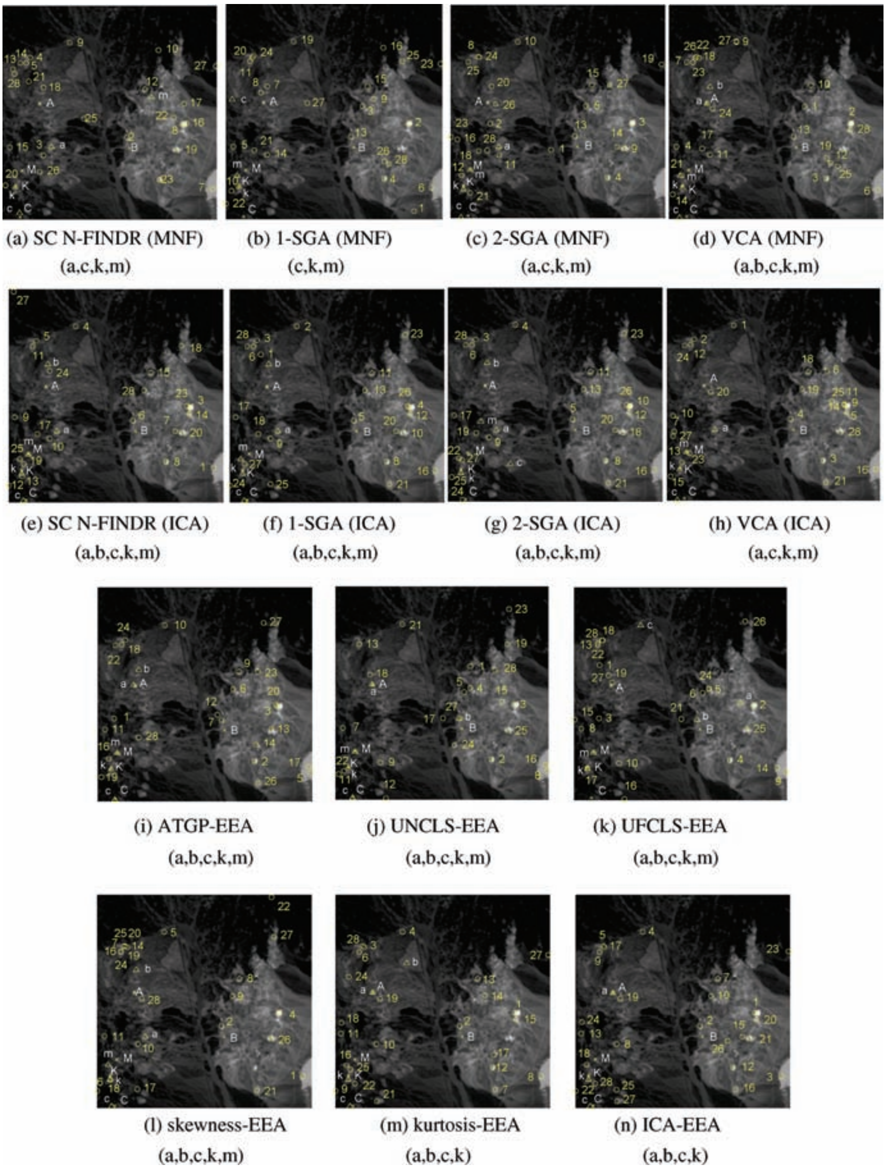
(i) ATGP-EEA (j) UNCLS-EEA (k) UFCLS-EEA

(l) skewness-EEA (m) kurtosis-EEA (n) ICA-EEA

Figure 8.7 Results for scenario TE3 with six endmembers extracted by SC N-FINDR, 1-SGA, 2-SGA, VCA, ATGP-EEA, UFCLS-EEA, UNCLS-EEA, skewness-EEA, kurtosis-EEA, and ICA-EEA using one set of random initial endmembers.

value of p was chosen to be a larger vaule of $n_{\text{SSE}} = 10$ and $n_{\text{VD}} = 9$. In the figure, the endmember pixels extracted by algorithms are marked by yellow circles and the pixels marked by yellow triangles are identified by SQ-EEAs corresponding to the five panel signatures and listed in the parenthesis underneath each figure.

Comparing Figure 8.9 with Figure 7.27, we observe that SQ-EEAs performed at least equally as SM-EEAs, including their counterpart, N-FINDR, and FCLS-EEA. Also, analogous to the Cuprite experiments, using HOS-based SQ-EEAs or ICAs to perform DR did improve the performance of endmember extarction.



(a) SC N-FINDR (MNF) (a,c,k,m)

(b) 1-SGA (MNF) (c,k,m)

(c) 2-SGA (MNF) (a,c,k,m)

(d) VCA (MNF) (a,b,c,k,m)

(e) SC N-FINDR (ICA) (a,b,c,k,m)

(f) 1-SGA (ICA) (a,b,c,k,m)

(g) 2-SGA (ICA) (a,b,c,k,m)

(h) VCA (ICA) (a,c,k,m)

(i) ATGP-EEA (a,b,c,k,m)

(j) UNCLS-EEA (a,b,c,k,m)

(k) UFCLS-EEA (a,b,c,k,m)

(l) skewness-EEA (a,b,c,k,m)

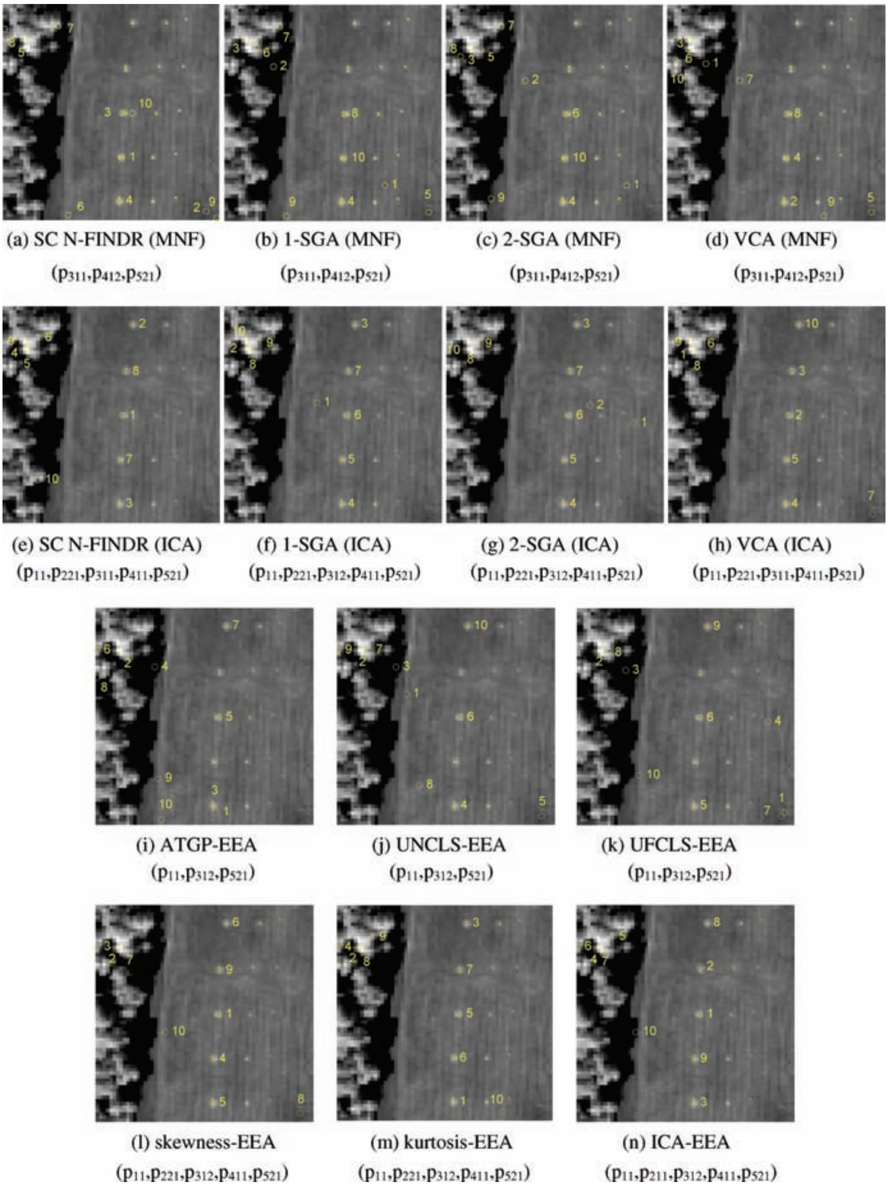
(m) kurtosis-EEA (a,b,c,k)

(n) ICA-EEA (a,b,c,k)

Figure 8.8 Results for cuprite reflectance with $n_{\text{SSE}} = 28$ ($n_{\text{VD}} = 22$) endmember extracted by SC N-FINDR, 1-SGA, 2-SGA, VCA, UFCLS-EEA, ATGP-EEA, skewness-EEA, kurtosis-EEA, and ICA-EEA using one set of random initial endmembers.

8.8 Conclusions

SM-EEAs considered in Chapter 7 extract endmembers simultaneously for a given number of endmembers, p. One major difficulty of implementing SM-EEAs is finding all the endmembers at once, which results in high computational complexity. Another difficulty is that once the value of p is changed an SM-EEA must be reimplemented. In other words, all the previously generated



(a) SC N-FINDR (MNF) $(p_{311}, p_{412}, p_{521})$

(b) 1-SGA (MNF) $(p_{311}, p_{412}, p_{521})$

(c) 2-SGA (MNF) $(p_{311}, p_{412}, p_{521})$

(d) VCA (MNF) $(p_{311}, p_{412}, p_{521})$

(e) SC N-FINDR (ICA) $(p_{11}, p_{221}, p_{311}, p_{411}, p_{521})$

(f) 1-SGA (ICA) $(p_{11}, p_{221}, p_{312}, p_{411}, p_{521})$

(g) 2-SGA (ICA) $(p_{11}, p_{221}, p_{312}, p_{411}, p_{521})$

(h) VCA (ICA) $(p_{11}, p_{221}, p_{311}, p_{411}, p_{521})$

(i) ATGP-EEA $(p_{11}, p_{312}, p_{521})$

(j) UNCLS-EEA $(p_{11}, p_{312}, p_{521})$

(k) UFCLS-EEA $(p_{11}, p_{312}, p_{521})$

(l) skewness-EEA $(p_{11}, p_{221}, p_{312}, p_{411}, p_{521})$

(m) kurtosis-EEA $(p_{11}, p_{221}, p_{312}, p_{411}, p_{521})$

(n) ICA-EEA $(p_{11}, p_{211}, p_{312}, p_{411}, p_{521})$

Figure 8.9 Results for the HYDICE scene with $n_{\text{SSE}} = 10$ $(n_{\text{VD}} = 9)$ endmember extracted by SC N-FINDR, 1-SGA, 2-SGA, VCA, UFCLS-EEA, ATGP-EEA, skewness-EEA, kurtosis-EEA, and ICA-EEA using one set of random initial endmembers.

endmembers by an SM-EEA cannot be used and new endmembers must be regenerated. Such a circumstance arises when p is not known precisely, and must be tested on a trial-and-error basis. This chapter has addressed these two issues by developing SQ-EEAs that implement SM-EEAs in a sequential manner. More specifically, an SQ-EEA finds one endmember at a time and one after another sequentially rather than all endmembers together simultaneously as an SM-EEA does.

Table 8.1 Summary of design criteria of SQ-EEAs and their advantages and disadvantages

SQ-EEAs	Design criteria	Advantages	Disadvantages
SC N-FINDR	Maximum simplex volume	Less computation	Precise knowledge about p
SGA	Maximum simplex volume	Less computation	Precise knowledge about p
VCA	Orthogonal projection	Less computation	1. Precise knowledge about p 2. Random initial conditions
ATGP-EEA	Unconstrained OSP	1. Fast computation 2. Effective in finding spectrally distinct targets	1. Precise knowledge about p 2. Computational complexity 3. Presence of pure signatures
UNCLS-EEA	ANC-constrained OSP	Effective in finding spectrally distinct targets	1. Precise knowledge about p 2. Computational complexity
UFCLS-EEA	Fully abundance-constrained OSP	Effective in finding spectrally distinct targets	1. Precise knowledge about p 2. Computational complexity 3. Ill-rank of endmember matrix
Skewness-EEA	Third-order statistics	Effective in finding small targets	1. Determination of number of components, p 2. Random initial conditions
Kurtosis-EEA	Fourth-order statistics	Effective in finding small targets	1. Determination of number of components, p 2. Random initial conditions
ICA-EEA	Statistical independency	Effective in finding small targets	1. Determination of number of components, p 2. Random initial conditions

Most importantly, an SQ-EEA adapts its ability to numbers of endmembers, p, which can also be adaptive. It uses previously generated endmembers as part of future endmembers as the number of endmembers, p, grows. As a result, computational complexity can greatly be reduced. In addition, most SQ-EEAs do not require dimensionality reduction as SM-EEAs do. Finally, Table 8.1 summarizes design criteria of various SQ-EEAs presented in this chapter and their advantages and disadvantages.

9

Initialization-Driven Endmember Extraction Algorithms (ID-EEAs)

One major issue arising in all the endmember extraction algorithms (EEAs) developed in Chapters 7 and 8 is the use of randomly generated initial endmembers to initialize algorithms. Accordingly the final set of selected endmembers are generally not the same and the results are not repeatable. Such inconsistency certainly causes discrepancies during data analysis. Another issue is that EEAs are also sensitive to how initial endmembers are chosen. Unfortunately, very little effort has been devoted to deal with these problems until recent works by Chang and Plaza (2006) and Plaza and Chang (2006). Interestingly, a good initial condition can not only reduce computing time significantly but also produce consistent final results. An appropriately selected set of initial endmembers can make tremendous improvement in performance and computational complexity on the endmember extraction process. To address such initialization issues, this chapter introduces initialization-driven (ID) EEAs (ID-EEAs) where their initial conditions can be selected by a custom-designed process. Two procedures are developed for such a selection of initial endmembers. One procedure is called the initial endmember-driven (IED) initialization that is generally used by SQ-EEAs to produce the first initial endmember. The other procedure introduces a new concept of endmember initialization algorithm (EIA) that is particularly designed to generate an appropriate set of initial endmembers for SM-EEAs. Furthermore, an EEA implemented in conjunction with EIA-generated initial endmembers can also significantly reduce the number of endmember replacements as well as computing time during the course of searching new endmembers. As a surprising finding, many of the EIA-generated initial endmembers turn out to be final desired endmembers.

9.1 Introduction

As demonstrated by the experiments in Chapters 7 and 8, one common issue in implementation of SM-EEA and SQ-EEA is their use of randomly selected data sample vectors as initial endmembers to initialize an EEA. As a consequence, the final set of selected endmembers by an EEA varies. More specifically, two different sets of random initial endmembers may produce different final sets of endmembers. Such inconsistency results from the nature of randomness caused by the use of random initial endmembers. Interestingly, very little attention was paid to this issue in the past until a recent work reported in Plaza and Chang (2006). Although Berman et al. (2004) also realized the problem of starting points (i.e., initial condition) when their iterated constrained

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

endmember (ICE) was developed, they did not specifically address this issue. Nevertheless, the ICE mitigated this problem by using PPI-generated endmembers as candidate points for their algorithm initialization in which case PPI can be considered as an EIA.

It is known that the ultimate goal of an EEA is to find pure spectrally distinct signatures present in the data. Despite that much effort has been devoted to the design and development of EEAs, it seems that very little has been done in addressing the issue of algorithm initialization which in fact has a significant impact on the final endmembers selected by an EEA. Such an initialization issue is also encountered in vector quantization where an algorithm may be trapped locally by an inappropriate selection of initial code words (Gersho and Gray, 1992) as well as in pattern classification where an unsupervised clustering process, called ISODATA (Duda and Hart, 1973) and also known as the *C*-means/*K*-means clustering algorithm, makes use of randomly generated initial data samples as its initial cluster means.

Generally, three primary factors play a key role in effectiveness of an EEA. One is how to adopt an appropriate criterion to design an EEA. As noted in Chapters 7 and 8, most effective ones are orthogonal projection (OP) used by PPI and VCA, and maximal/minimal simplex volume (MSV) used by N-FINDR and simplex growing algorithm (SGA). Another is to preset an appropriate number of endmembers, p, for an EEA to generate. In this case, determination of p becomes crucial and critical. If p is set too small, the extracted endmembers may not represent the data well. On the other hand, if p is set too large, some endmembers may not be pure signatures and can be either mixed or interfering signatures. This issue is applied to both SM-EEAs and SQ-EEAs. Once the value of p is determined, a third relevant issue is initialization of an EEA, especially, how to select a set of appropriate p initial endmembers to speed up the searching process while avoiding the process being trapped in local optima. From an algorithm design point of view, a good and effective algorithm should not depend on initial conditions which can only affect the algorithm convergence rate but cannot alter the final results if the process is run indefinitely. Unfortunately, this is generally not true for many algorithms that are implemented in a finite number of runs. For example, Newton's method will never converge if an initial condition is selected incorrectly (Hamming, 1989). As will also be demonstrated by experiments, an EEA also suffers from similar problems. As noted in Chapter 7, an SM-EEA requires an exhaustive search which is very computationally expensive. To circumvent this issue many EEAs developed in the literature are actually not SM-EEAs because their implemented searching processes are neither simultaneous nor exhaustive, but rather focused on sequential searches or narrowed their searches in selective feasible regions. As a consequence, an initial set of endmembers selected for algorithm initialization may ultimately determine the final result of endmembers. Accordingly, how to select a set of desired initial endmembers becomes a crucial step in design of EEAs.

This chapter introduces a new type of EEAs, called initialization-driven EEAs (ID-EEAs), which use specific initial conditions for algorithm initialization. Depending on how initial conditions are generated, ID-EEAs can be further categorized into initial endmember-driven EEAs (IED-EEAs) which are essentially designed for those EEAs that only require the first initial endmember to initialize the algorithms and endmember initialization algorithm (EIA)-driven EEAs (EIAD-EEAs) which are primarily designed for those EEAs that generally require a complete set of p initial endmembers for algorithm initialization.

9.2 Initialization Issues

From a viewpoint of algorithm design, three major issues determine the performance of an algorithm: initial conditions, stopping criteria, and learning rules. On some occasions, stopping criteria that set thresholds to terminate an algorithm are also closely related to initial conditions. Over the past few years, EEAs have been designed for finding endmembers which are mainly focused on the third issue, that is, learning rules. However, little work has been devoted to deal with algorithm initialization issues, which can be as important as learning rules as described in the following section.

9.2.1 Initial Conditions to Terminate an EEA

There are generally two initial conditions that can be used to terminate an EEA. One is to preset an error threshold, ε, to terminate an algorithm. Since the selection of an appropriate ε is usually data dependent, it is generally difficult to do so without prior knowledge about the data. If the value of ε is too small, the algorithm may take a long time to converge and may also run into a stability problem with fluctuating results. Besides, it may generate more endmembers than are actually needed. On the contrary, if the value of ε is too large, the algorithm may terminate earlier than it should. In this case, the set of generated endmembers may be insufficient and some desired endmembers will have to be left out. As an alternative, we may preset the number of endmembers needed to be generated, p. In this case, an alternative approach to selection of an error threshold ε is to determine an appropriate value for p, that is, how many endmembers are required for an EEA to generate. On one hand, if the value of p is too small, not all desired endmembers will be extracted. On other hand, if the value of p is too large, some extracted endmembers may turn out to be unwanted signatures which are not pure. Therefore, the same dilemma encountered in the selection of an appropriate value for ε also arises in the selection of an appropriate value for p. However, using a recently developed new concept of virtual dimensionality (VD) introduced in Chapter 5 finding an appropriate value of p seems more feasible and doable by setting $\mathrm{VD} = p$ because the VD provides a good estimate of the number of spectrally distinct signatures in the data and can be used to resolve the issue in determining the value of p. Therefore, this chapter assumes that the value of p is estimated by VD, and this VD-estimated p will be used to determine how many endmembers are required for an EEA to generate.

9.2.2 Selection of an Initial Set of Endmembers for an EEA

After the number of endmembers, p, is determined for an EEA, a follow-up step is to find an adequate set of initial endmembers $\left\{\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}\right\}$ to initialize the EEA. It is interesting to note that except the works reported in Chang and Plaza (2006), Plaza and Chang (2006), and Chang et al. (2006), very little has been done in the literature regarding how to select an initial set for an EEA. In fact, an appropriate selection of initial endmembers can be very beneficial. On some occasions, it is critical to produce correct final results, and in the mean time it also speeds up the endmember searching process. This indicates that finding appropriate initial conditions becomes necessary and highly desirable for algorithm design, for example, vector quantization (Katsavounides et al., 1994). Without a specific set of initial conditions a general practice to implement an EEA often starts with any set of initial endmembers, most likely, randomly generated. If an EEA conducts an exhaustive search for p endmembers, then the final results should not depend on the initial endmembers that are selected for initialization because all possible p combinations will be eventually exhausted for the endmember search. The only issue is its computing time which is largely determined by the number of searches, which is $\binom{N}{\mathrm{p}} = \frac{N!}{p!(N-p)!}$. Unfortunately, an exhaustive search generally suffers from several drawbacks. First, it is computationally very expensive, in particular, for hyperspectral imagery with large volumes of data sample vectors, N. Second, it has to exhaust all p combinations before finding an optimal set of endmembers even though such set may likely to be found in a very early stage. Third and most importantly, it is not feasible in practical applications. So, an efficient and effective EEA should not perform a fully exhaustive search, but rather a search for endmembers in certain feasible regions or should use an iterative search by taking advantage of previous search results as a base to improve the next search. However, the success of such approaches is heavily determined by initial conditions to be used for the initial search. If the initial conditions are poorly chosen, the algorithm may be trapped in local optima or may not even converge. Therefore, for an EEA to be efficient as well as effective, the selected initial endmembers must be representative and cannot be arbitrary. Interestingly, this issue has not received much attention in the past. As will be demonstrated by the experiments, EEAs are indeed sensitive to the initial endmembers and a judicious selection of initial endmembers is crucial for an EEA to be successful in producing the final results of endmembers.

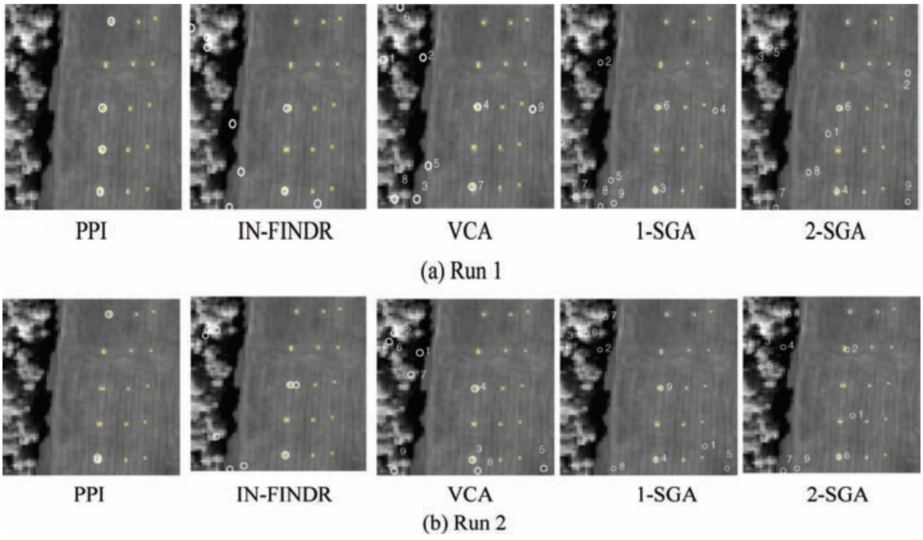
9.2.3 Issues of Random Initial Conditions Demonstrated by Experiments

To illustrate the issue caused by random initial conditions in data analysis, this section presents real image experiments to show how difficult interpretation can be when inconsistent results are produced by different sets of random initial conditions. Four major EEAs, PPI, and IN-FINDR in Chapter 7 and VCA and SGA in Chapter 8, all of which use random initial conditions, are used for illustration.

9.2.3.1 HYDICE Experiments

First, we demonstrate inconsistent results of endmembers extracted by PPI, N-FINDR, and VCA using two different sets of randomly generated initial endmember pixels in two separate runs where the numbers of endmembers are chosen to be 9 and 10 because of $\mathrm{VD}_{\mathrm{HFC}}^{\mathrm{NP}} = 9$ with $P_{\mathrm{F}} \leq 10^{-3}$ and $\mathrm{VD}_{\mathrm{SSE}} = 10$. Figures 9.1 and 9.2 show the results of endmember pixels extracted and marked with circles by PPI using 1000 skewers, IN-FINDR, VCA, and SGA in two different runs for $p = 9$ and 10, respectively, where MNF was used for DR.

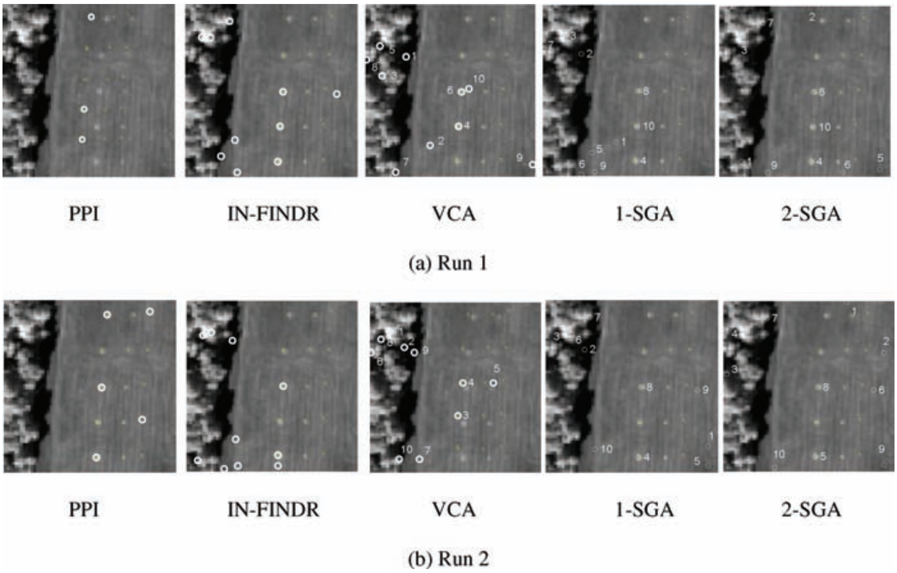
For $p = 9$, PPI produced four panel pixels p_{11}, p_{311}, p_{412}, and p_{521} in one run and another four panel pixels p_{11}, p_{312}, p_{412}, and p_{521} with one panel pixel different which is p_{312}. All the four panel pixels represented four distinct panel signatures, $\mathbf{p}_1$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$, as endmembers,



(a) Run 1

(b) Run 2

Figure 9.1 Endmember pixels extracted by PPI, IN-FINDR, VCA, and SGA with MNF in two different runs for $p = 9$.



PPI IN-FINDR VCA 1-SGA 2-SGA

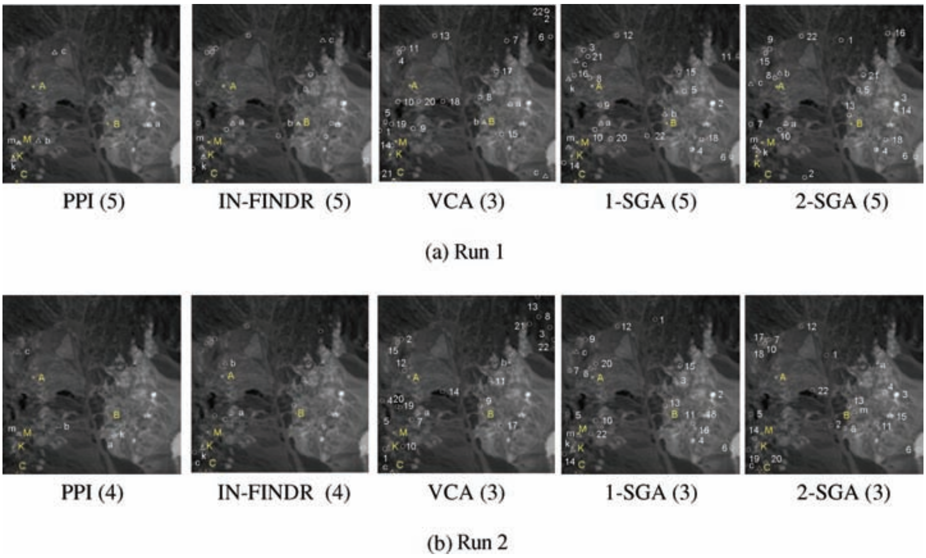
(a) Run 1

PPI IN-FINDR VCA 1-SGA 2-SGA

(b) Run 2

Figure 9.2 Endmember pixels extracted by PPI, IN-FINDR, VCA, and SGA with MNF in two different runs for $p = 10$.

but their PPI counts were different. On the other hand, for $p = 10$, PPI only extracted three panel pixels, p_{11}, p_{412}, and p_{521}, in one run and four panel pixels, p_{11}, p_{411}, p_{412}, and p_{521}, in another run. All the extracted panel pixels represented three distinct panel signatures, $\mathbf{p}_1$, $\mathbf{p}_4$, and $\mathbf{p}_5$, as endmembers which were one short of four endmembers with $p = 9$. These interesting experiments demonstrated one important observation. That is, a larger $p = 10$ did not necessarily guarantee to perform better than a smaller $p = 9$. This is mainly due to two major reasons. One is that unlike an SQ-EEA, which can take advantage of endmembers extracted for a smaller value of p as part of endmembers generated for a larger value of p, PPI is an SM-EEA and must regenerate all endmembers for different values of p. The other reason is that PPI produced different PPI counts for endmembers in different runs. In this case, the panel pixel p_{311} which appeared in one run may also disappear in another run due to its low PPI count. Similarly, among all the pixels extracted by IN-FINDR in the case of $p = 9$ only two were real endmember pixels, p_{311} and p_{521}, representing two distinct panel signatures $\mathbf{p}_3$ and $\mathbf{p}_{7.}$ However, among all the pixels extracted by IN-FINDR for $p = 10$ three panel pixels, p_{311}, p_{412}, and p_{521}, were extracted in one run and two panel pixels, p_{311} and p_{521}, were extracted in another run. Figures 9.1 and 9.2 show that VCA suffered from the same dilemma encountered in PPI and IN-FINDR where only two panel pixels which might be different are extracted by all the cases. Since the VCA is an SQ-EEA, the pixels in Figures 9.1 and 9.2 labeled by numbers indicate the order in which these pixels were extracted in sequence by VCA. It is also interesting to note that the VCA extracted a different set of distinct panel signatures in two different runs, $\{\mathbf{p}_3,\mathbf{p}_5\}$ and $\{\mathbf{p}_3,\mathbf{p}_4\}$, for $p = 9$, while the same two distinct panel signatures, $\mathbf{p}_3$ and $\mathbf{p}_4$, were extracted for $p = 10$. Finally, SGA performed comparably to IN-FINDR and VCA for $p = 9$ but slightly better than VCA for $p = 10$ where SGA extracted three panel pixels p_{311}, p_{412} and p_{521} in one run compared to only two panel pixels p_{311}, p_{412} extracted by VCA in both runs. These experiments demonstrate that the value of p actually has a large compact on endmember extraction.



PPI (5) IN-FINDR (5) VCA (3) 1-SGA (5) 2-SGA (5)

(a) Run 1

PPI (4) IN-FINDR (4) VCA (3) 1-SGA (3) 2-SGA (3)

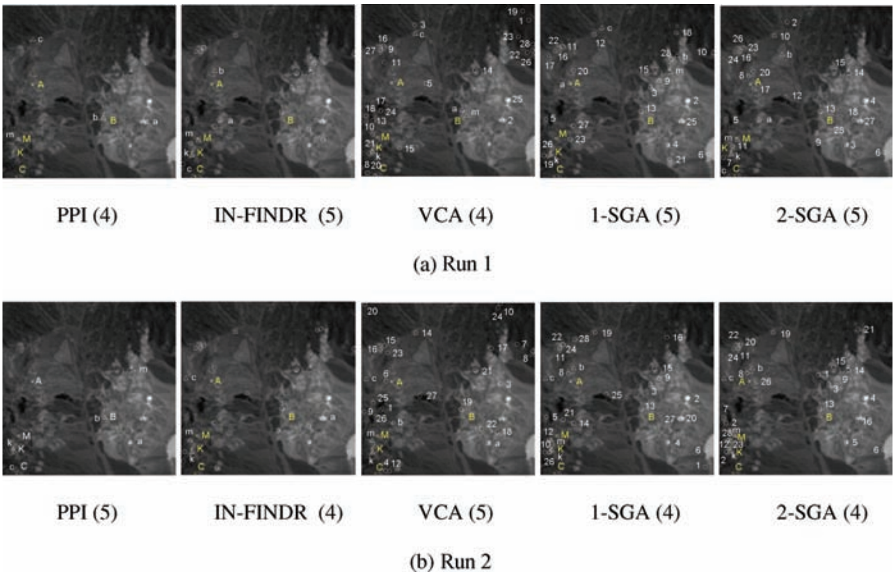
(b) Run 2

Figure 9.3 Endmember pixels extracted by PPI, IN-FINDR, VCA, and SGA with MNF in two different runs for $p = 22$.

9.2.3.2 AVIRIS Experiments

For the following AVIRIS experiments, p was chosen to be 22 and 28 for comparative analysis. Experiments similar to those conducted for the HYDICE image scene in Section 9.2.3.1 were also performed for the Cuprite image scene in Figure 1.11(b). Figures 9.3 and 9.4 demonstrate that the same dilemma encountered in the HYDICE experiments also occurred in the AVIRIS experiments for the PPI, SM N-FINDR, and VCA with MNF used for DR where their final results produced by two different runs were inconsistent for both $p = 22$ and 29. In these figures, the pixels marked by open circles were extracted by EEAs and the pixels marked by the lowercases of “a, b, c, k, m” with triangles were the desired endmember pixels corresponding to the five ground truth mineral endmembers provided in Figure 1.11(b) and marked by the uppercases of “A, B, C, K, M” with yellow crosses “x” in the sense of spectral similarity measured by the spectral angle mapper (SAM) (Chang, 2003a). Therefore, only spatial locations of the extracted endmember pixels were identified and shown in the figures. Additionally, the numerals in open parentheses underneath the figures indicate the numbers of extracted endmember pixels that were identified in correspondence with ground truth mineral pixels by the SAM. The pixels in Figure 9.4 labeled by the numbers indicate the order in which these pixels were extracted in sequence by VCA. It should also be noted that the PPI implemented here used 500 skewers for computational convenience.

As shown in Figures 9.3 and 9.4, PPI and IN-FINDR extracted different numbers of ground-truth-corresponding endmember pixels, 5 and 4, in two different runs for $p = 22$ and 29. It was also true for the VCA with $p = 29$. Moreover, these extracted ground-truth-corresponding endmember pixels were generally not the same. This is mainly due to the fact that there were other pixels whose spectral signatures were also similar and very close to ground truth mineral signatures. As a result, a different run was very much likely to extract different pixels corresponding to ground truth pixels. This evidence was demonstrated in Figure 1.11(a) where the two sets of three endmember pixels corresponding to alunite (A), buddingtonite (B), and calcite (C) produced by the VCA in Figure 1.11(a) for $p = 22$ in two separate runs were different.



PPI (4) IN-FINDR (5) VCA (4) 1-SGA (5) 2-SGA (5)

(a) Run 1

PPI (5) IN-FINDR (4) VCA (5) 1-SGA (4) 2-SGA (4)

(b) Run 2

Figure 9.4 Endmember pixels extracted by PPI, IN-FINDR, VCA, and SGA with MNF in two different runs for $p = 28$.

9.3 Initialization-Driven EEAs

Thus far, we have described two types of EEAs, SM-EEAs in Chapter 7 and SQ-EEAs in Chapter 8, both of which make use of initial endmembers randomly selected from the data. As demonstrated in experiments, the use of random initial endmembers results in inconsistent final selections of endmembers. An ID-EEA for PPI was first developed by Chang and Plaza (2006) to cope with this particular issue and more general issues were later investigated in Plaza and Chang (2006). Due to the natural difference in implementation of an SM-EEA and an SQ-EEA, how to find a specific set of initial endmembers for an SM-EEA and an SQ-EEA is also different. For a given number of endmembers, p, an initial set of endmembers for an SM-EEA needs p endmembers altogether to initialize an SM-EEA. On the other hand, an SQ-EEA produces one endmember at a time sequentially. In this case, an initial set of endmembers used to initialize an SQ-EEA is generally a singleton set which consists of only one endmember instead of p endmembers. As a result, finding an appropriate set of initial endmembers for an SQ-EEA is reduced to looking for a specific data sample that not only can speed up the endmember-searching process but also can produce consistent final results. So, an EEA which only requires the first endmember to be generated for its initial condition is called IED-EEA. When an EEA is an SQ-EEA using a specific data sample as an initial endmember, it is called IED-SQ-EEA. On the other hand, an SM-EEA requires a set of p initial endmembers to be specified. In this case, it needs an algorithm to produce a good set of initial endmembers which can speed up the search process implemented in an EEA to converge rapidly to the desired final results. An algorithm designed for this purpose is called EIA. An EEA makes use of an EIA to produce its initial condition called an EIA-driven (EIAD)-EEA. In particular, when an EEA is an SM-EEA using an EIA as an initialization algorithm to produce an initial set of p-specific endmembers, it is called an EIAD-SM-EEA. Interestingly, an SQ-EEA can also be used to serve as an EIA. This is because an SQ-EEA is not necessarily optimal but provides very close final results. Consequently, an SM-EEA using an SQ-EEA as an EIA may be the best way to produce a final set of desired endmembers in terms of computational complexity and consistency in final results.

9.3.1 Initial Endmember-Driven EEAs

In Chapter 8, six SQ-EEAs are developed, SC N-FINDR, SGA, VCA, UFCLS/IEA-EEA, ATGP-EEA, and HOS-EEAs, each of which represents a specific category associated with one particular criterion. These SQ-EEAs can be implemented as IED-EEAs by specifying their first initial endmember in two ways.

9.3.1.1 Finding Maximum Length of Data Sample Vectors

The idea of finding a data sample vector that yields the maximum length among all data sample vectors came from the automatic target detection and classification algorithm developed by Ren and Chang (2003), that is,

$$\mathbf{t}_0 = \arg\left\{\max_{\mathbf{r}}\mathbf{r}^T\mathbf{r}\right\}, \tag{9.1}$$

where $\mathbf{r}$ runs over all data sample vectors. Since there is no prior knowledge regarding what signatures of interest we are looking for, a best hope is to search for a data sample vector with the most spectrally distinct signature which may be more likely to be an endmember. Equation (9.1) provides such a signature $\mathbf{t}_0$ that can be used to replace the randomly generated initial endmember by EEAs. Using (9.1) to specify their initial endmembers, EEAs of this type include ATGP-EEA, UNCLS, UFCLS-EEA, VCA, and HOS-EEAs, which result in IED-ATGP-EEA, IED-UNCLS, IED-UFCLS, IED-VCA, and IED-HOS-EEAs. However, there is a different way for IED-VCA and IED-HOS-EEAs to implement (9.1). In step 4 of the VCA algorithm described in Section 8.4, a Gaussian random vector is used to generate an initial vector to produce an initial endmember every time after an endmember is generated. As also shown, the VCA can also be implemented using a uniform random variable to replace the Gaussian random vector. To implement VCA as IED-VCA, the random variable used by VCA must be replaced by specifically chosen initial endmembers as described in the following.

IED-VCA (step 4 implemented in VCA is replaced by the following step):

1. At the nth iteration, the randomly generated Gaussian random vector, $\mathbf{w}^n$, used in (8.1) is replaced by finding the maximum of $\mathbf{r}^T\mathbf{r}$ with $\mathbf{r} \in < \mathbf{A}^{(n-1)} >^{\perp}$, that is, $\mathbf{w}^{(n)} = \arg\left\{\max_{\mathbf{r}\in<\mathbf{A}^{(n-1)}>^{\perp}}\mathbf{r}^T\mathbf{r}\right\}$.

As noted in the above IED-VCA, it requires a new random initial projection vector every time a new search is initiated for a new endmember. Compared to IED-VCA, IED-ATGP only needs one initial random projection vector for initialization. Once IED-ATGP is initialized, the algorithm automatically carries out the rest of orthogonal projections without any more random projection vectors. This difference sheds light on how to resolve the issue in the use of random projection vectors frequently occurred in projection pursuit (PP)-based algorithms including VCA in Section 8.4, high-order statistics-based EEA (HOS-EEA) in Section 8.6, and independent component analysis-based EEA (ICA-EEA) in Section 8.6.5. Using exactly the same ideas of extending VCA to IED-VCA in Section 9.3.1.1, HOS-EEA and ICA-EEA can also be extended in the same fashion. In other words, HOS-EEA and ICA-EEA can be extended to IED-HOS-EEA and IED-ICA-EEA by specifying the brightest pixel with the highest intensity as their initial projection vector every time they search for a new endmember.

Unlike VCA and HOS-EEA, which only need to specify one initial endmember, the SGA described in Section 8.3 requires two initial endmembers for initialization since the minimum number of forming a simplex is two vertices. In this case, a best pair of two initial endmembers that can be used to form a two-dimensional simplex is made up of two data samples ($\mathbf{e}^{(0)}$,$\mathbf{e}^{(1)}$) that yield the maximum Euclidean distance among all possible pairs of data sample vectors in a reduced one-dimensional data space obtained by a DR transform. This is because such a pair represents the maximum volume of two-dimensional simplexes. Using this pair of data samples ($\mathbf{e}^{(0)}$,$\mathbf{e}^{(1)}$) to replace the random initial endmember used in step 1 of SGA, we can obtain the following IED-SGA.

IED-2-SGA (step 1 in 1-SGA is replaced by the following step):

1. Initialization
   a. Let p be the number of endmembers to be generated.
   b. Find ($\mathbf{e}^{(0)}$,$\mathbf{e}^{(1)}$) which produce the maximum distance in a one-dimensional data space obtained by any DR transform and set $k = 2$.

It should be noted that the above IED-SGA is slightly different from the SGA developed in Chang et al. (2006), which uses the first endmember selection process described below.

First Endmember Selection Process for 1-SGA.

1. Randomly generate a target sample vector, denoted by $\mathbf{t}$.
2. Find a pixel $\mathbf{e}^{(0)}$ that yields the maximum of absolute determinant of the matrix, $\left|\det\begin{bmatrix}1 & 1\\ \mathbf{t} & \mathbf{r}\end{bmatrix}\right|$ over all sample vectors $\mathbf{r}$, that is, $\mathbf{e}^{(0)} = \arg\left\{\max_{\mathbf{r}}\left[\left|\det\begin{bmatrix}1 & 1\\ \mathbf{t} & \mathbf{r}\end{bmatrix}\right|\right]\right\}$ where PCA or MNF is required to reduce the original data dimensionality L to dimension 2 to find the maximum.

The generation of the above first endmember pixel $\mathbf{e}^{(0)}$ is determined by the randomly generated target sample vector $\mathbf{t}$. A different target sample vector $\mathbf{t}$ may result in a different $\mathbf{e}^{(0)}$. Interestingly, such a generated $\mathbf{e}^{(0)}$ is always a sample vector that has either a maximum or a minimum value in the first component after dimensionality reduction transform. Therefore, the target sample vector $\mathbf{t}$ has no effect on the final set of endmembers. Furthermore, it also shows that the $\mathbf{e}^{(0)}$ which is generated by the above first endmember selection process and the $\mathbf{e}^{(1)}$ which has the maximum distance from the $\mathbf{e}^{(0)}$ turn out to be exactly the same pair ($\mathbf{e}^{(0)}$, $\mathbf{e}^{(1)}$) obtained by step 1(b) in IED-2-SGA. This indicates that the above IED-2-SGA is essentially the same SGA as developed in Chang et al. (2006). It is also interesting to note that the computational complexity of producing the first two endmembers by IED-1-SGA is $N \times (N-1)$ compared to $\binom{N}{2} = \frac{N \times (N-1)}{2!}$, which is the computational complexity of producing the first two endmembers by IED-2-SGA where N is the total number of data sample vectors. This also implies that IED-2-SGA only conducts half of searches required by IED-1-SGA. Consequently, IED-1-SGA and IED-2-SGA generally perform differently since initial conditions determine final results of extracted endmembers.

9.3.1.2 Finding Sample Mean of Data Sample Vectors

The IEA discussed in Section 8.5.4 is not really the one developed by Neville et al. (1999) since the initial endmember used by Neville et al.'s IEA was not selected randomly. As a matter of fact, Neville et al.'s IEA calculates the data sample mean as an initial target vector, $\mathbf{e}^{(0)}$, and find a set of pixels in $R^{(0)}$ that are within the spectral angle θ and farthest from the obtained mean, $\mathbf{t}_0$, denoted by $R^{(0)}(\theta)$. Then they further calculate the average of pixels in $R^{(0)}(\theta)$ and use it as the first endmember, denoted by $\mathbf{e}^{(1)}$. So, according to its specific selection of an initial endmember, Neville et al.'s IEA is actually IED-IEA for consistency in notation. Since the sample mean is basically a mixture of all data sample vectors, it is expected that it cannot be an endmember in which case using the sample mean may not be a good candidate to be selected as an initial endmember.

9.3.2 Endmember Initialization Algorithm for SM-EEAs

As noted previously, an SQ-EEA may not be an optimal EEA as an SM-EEA is. So, its final endmembers produced may not be all desired endmembers. But according to experiments, many of SQ-EEA-generated endmembers eventually turn out to be the desired true endmembers. By taking advantage of this fact, an SQ-EEA can be also used as an effective EIA which provides a better set of initial endmembers that help an SM-EEA converge to final results rapidly. Nevertheless, the issue arising in inconsistency caused by randomness remains unsolved if an SQ-EEA also uses random initial endmembers.

In Section 9.3.1, an SQ-EEA is extended to an IED-SQ-EEA by specifying a particularly selected data sample as an initial endmember to initialize an SQ-EEA to avoid inconsistency resulting from the use of random initial endmembers. This is because an SQ-EEA finds one endmember at a time and only one sample is needed to initialize an SQ-EEA. However, the same strategy is not applicable to an SM-EEA due to the fact that an SM-EEA needs a set of p initial endmembers to initialize an SM-EEA instead of a single specific initial endmember only required by an SQ-EEA. In this case, we need to develop an algorithm, EIA, that allows us to produce a good set of candidates used for initial endmembers. Interestingly, an SQ-EEA, no matter whether it is an SQ-EEA using a random initial endmember or an IED-SQ-EEA, can also be used to serve as the purpose of EIA. Since an SQ-EEA may not be necessarily an optimal EEA, its extracted endmembers may not be all desired endmembers. But, according to experiments, many of its extracted endmembers turn out to be desired endmembers. Using a set of an SQ-EEA-extracted endmembers as an initial endmember to initialize an SM-EEA yields fast convergence to a final set of desired endmembers. In this section, we describe three types of algorithms, SQ-EEAs including IED-SQ-EEAs, Maxmin-distance algorithm (Tou and Gonzalez, 1974), and ISODATA (Duda and Hart, 1973) that can be used as EIAs.

9.3.2.1 SQ-EEAs

Despite the fact that all the SQ-EEAs developed in Chapter 8 are suboptimal EEAs, many of SQ-EEA-generated endmembers eventually turn out to be desired true endmembers. This suggests that an SQ-EEA can be used as an EIA with three benefits. One is to produce better sets of initial endmembers for an SM-EEA without random search in initialization. Another is to avoid unnecessary search for undesired endmembers so as to achieve fast convergence. The third benefit is that in many cases, the SQ-EEA-generated endmembers may actually end up as the final endmembers for an SM-EEA; this indicates that an SM-EEA, implemented in conjunction with an SQ-EEA used as an EIA, not only can produce optimal results, but also can reduce computational complexity tremendously in a significant order.

An EIA shares with an SQ-EEA some features which are not shared with an SM-EEA. First of all, when an SM-EEA is implemented, it assumes that the number of endmembers is known in advance and produces p endmembers *simultaneously*. So, for a different value of p, an SM-EEA generally produces a different set of endmembers. In other words, for any given number of endmembers, p, an SM-EEA must recalculate all the endmembers and cannot take advantage of a set of $p-1$ endmembers previously generated by the same algorithm. Also, these $p-1$ endmembers do not necessarily constitute a subset of the set of p endmembers generated in the end. On the other hand, an EIA produces a set of target pixels in *sequential* order. As a result, a set of

p EIA-generated pixels always includes the set of previously generated $p-1$ target pixels. This feature is highly desirable for an EIA because it can save tremendous computational time.

9.3.2.2 Maxmin-Distance Algorithm

In this subsection, we describe a very simple EIA called the Maxmin-distance algorithm, which has been commonly used in pattern recognition applications (Tou and Gonzalez, 1974). It can generate a reasonably good set of initial endmembers. Let the first initial endmember be obtained as the pixel vector with the maximum length, that is, $\mathbf{e}_1 = \arg\{\max_{\mathbf{r}} \mathbf{r}^T \mathbf{r}\}$. Then, for each $2 \leq j \leq p$, the jth endmember $\mathbf{e}_j$ with the largest distance to the set $S_{j-1} = \{\mathbf{e}_1, \mathbf{e}_2, \cdots, \mathbf{e}_{j-1}\}$ is defined and found by the following expression: $\mathbf{e}_j = \arg\left\{\max_{\mathbf{r}} d(\mathbf{r}, S_{j-1})\right\}$, where the distance $d(\mathbf{r}, S_{j-1})$ is defined by

$$d(\mathbf{r}, S_{j-1}) = \min_{1 \leq k \leq j-1} d(\mathbf{r}, \mathbf{e}_k) = \min\left\{d(\mathbf{r}, \mathbf{e}_1), d(\mathbf{r}, \mathbf{e}_2), \ldots, d(\mathbf{r}, \mathbf{e}_{j-1})\right\}. \tag{9.2}$$

It is worth noting that when $j = 2$, then $S_1 = \{\mathbf{e}_1\}$, in which case $\mathbf{e}_2 = \arg\{\max_{\mathbf{r}} d(\mathbf{r}, \mathbf{e}_1)\}$. It should also be noted that the distance measure used in the Maxmin-distance algorithm can be any spectral similarity measure such as the Euclidean distance, SAM, or spectral information divergence (Chang, 2000, 2003b). In this work, we rely on SAM to produce a set of initial endmembers $\left\{\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}\right\}$ using the Maxmin-distance algorithm.

9.3.2.3 ISODATA

The ISODATA (Duda and Hart, 1973), also known as the k-means or c-means method, is an unsupervised clustering algorithm which has been widely used in image classification and segmentation. Its implementation can be briefly described as follows.

ISODATA Algorithm

1. Initialization
   Determine the number of pattern classes p and randomly select p class means $\boldsymbol{\mu}_i^{(n)}$ for $1 \leq i \leq p$ and let $n = 0$.
2. At the $n \geq 0$ iteration, compute the distance of each sample pixel vector from all class means, $\boldsymbol{\mu}_i^{(n)}$ for $1 \leq i \leq p$ and assign the sample vector to the class whose mean has the shortest distance to the sample vector.
3. Compute the means of reclustered sample vectors for each class, $\hat{\boldsymbol{\mu}}_i^{(n)}$ for $1 \leq i \leq p$.
4. If there is any mean changed, that is $\hat{\boldsymbol{\mu}}_i^{(n)} \neq \boldsymbol{\mu}_i^{(n)}$ for some $1 \leq i \leq p$, let $\boldsymbol{\mu}_i^{(n)} \leftarrow \hat{\boldsymbol{\mu}}_i^{(n)}$ and $n \leftarrow n+1$. Go to step 2. Otherwise, the obtained $\left\{\boldsymbol{\mu}_i^{(n)}\right\}_{i=1}^{p}$ are the desired class means and the algorithm is terminated.

Now, if we use $\left\{\boldsymbol{\mu}_i^{(n)}\right\}_{i=1}^{p}$ obtained by ISODATA as an initial set of endmember to initialize an SM-EEA, the ISODATA becomes an EIA.

9.3.3 EIA-Driven EEAs

An EEA implemented in conjunction with an EIA to produce an appropriate set of initial endmembers is called EIA-EEA. When an EEA is an SM-EEA, it is referred to as EIA-SM-EEA. Similarly, when an EEA is an SQ-EEA, it is also referred to as EIA-SQ-EEA. As examples, using ATGP-EEA as an EIA SC N-FINDR and VCA can be extended to ATGP-SC N-FINDR and ATGP-VCA as follows.

ATGP-SC N-FINDR (step 2 implemented in the SC N-FINDR algorithm is replaced by the following step 2):

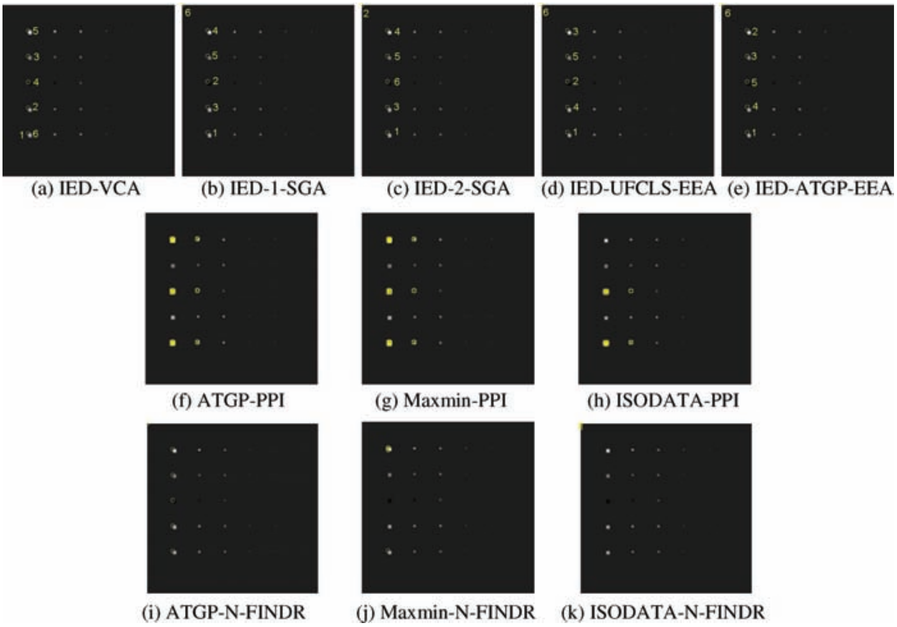
1. Initialization

   Let $\left\{\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}\right\}$ be generated by ATGP-EEA and be a set of initial vectors randomly generated from the data.

   ATGP-VCA (steps 3 and 4 implemented in VCA algorithm is replaced by the following steps 3 and 4):

2. Set the initial vector $\mathbf{e}^{(0)} = \underbrace{(0, 0, \ldots, 1)}_{p}$ and let a $p \times p$ auxiliary matrix $\mathbf{A}^{(0)}$ be $\mathbf{A}^{(0)} = \left[\mathbf{e}^{(0)}\, \mathbf{0} \cdots \mathbf{0}\right]$. Also implement ATGP-EEA to produce a set of p initial endmembers, denoted by $\left\{\mathbf{e}^{(k)}\right\}_{k=1}^{p}$.
3. At the nth iteration, the randomly generated Gaussian random vector $\mathbf{w}^k$ used in (8.1) is replaced by $\mathbf{w}^{(k-1)} = \mathbf{e}^{(k)}$, where $\left\{\mathbf{e}^{(k)}\right\}_{k=1}^{p}$ are ATGP-generated target pixels.

Figure 9.5 shows results for the TI1 scenario with six endmembers extracted by various ID-EEAs, IED-VCA, IED-1-SGA, IED-2-SGA, IED-UFCLS-EEA, IED-ATGP-EEA, ATGP-PPI, Maximin-PPI, ISODATA-PPI, ATGP-N-FINDR, Maxmin-N-FINDR and ISODATA-N-FINDR. As we can see from the figure, all IED-EEAs were able to extract all five mineral signatures except PPI using IED and N-FINDR using Maxmin and ISODATA as IED. These experiments made sense. The reason that PPI using IED did not work was because PPI required a large number of skewers to find directions of endmembers. Using IED simply did not provide enough directions. Furthermore, the experiments in Figures 9.5(j) and 9.5(k) also showed that using the maxmin and ISODATA which are popular in traditional image processing as IED did not work. This implies that not any unsupervised method can be used for the purpose of IED.


(a) IED-VCA (b) IED-1-SGA (c) IED-2-SGA (d) IED-UFCLS-EEA (e) IED-ATGP-EEA

(f) ATGP-PPI (g) Maxmin-PPI (h) ISODATA-PPI

(i) ATGP-N-FINDR (j) Maxmin-N-FINDR (k) ISODATA-N-FINDR

Figure 9.5 Results for the TI1 scenario with six endmembers extracted by various ID-EEAs.

Generally, the ATGP-EEA used in step 3 of the above ATGP-VCA can be replaced by any EIA. However, the reason that the ATGP is chosen for EIA-VCA is simply based on the fact that both ATGP and VCA are OP-based techniques. It is very much likely that an ATGP-generated initial endmember $\mathbf{t}_n$ turns out to be the same as a VCA-generated endmember $\mathbf{e}^{(k)}$ solved by (8.5). As similar examples, HOS-EEA and ICA-EEA can also be extended to EIA-HOS-EEA and EIA-ICA-EEA in the same manner in which VCA is extended to EIA-VCA described above. If the EIA used in EIA-HOS-EEA and EIA-ICA-EEA is chosen to be ATGP-EEA to produce a set of p projection vectors that can be used as initial projection vectors for each of p searches for new endmembers, the EIA-HOS-EEA and EIA-ICA-EEA are referred to as ATGP -HOS-EEA and ATGP-ICA-EEA. Figure 9.6 provides a block diagram of SQ-EEAs in Chapter 8 which can be implemented as IED-EEAs and SM-EEAs in Chapter 7 which include an EIA as their initialization algorithm to become EIA-SM-EEAs.

Two comments are noteworthy:

1. There are two ways to implement VCA as ID-VCA, one is IED-VCA and the other is EIA-VCA.
2. Although both SC N-FINDR and SGA discussed in Sections 8.2 and 8.3 are sequential versions of SM N-FINDR, SC N-FINDR requires a complete set of p endmembers for initialization. In this case, the IED approach that works for SGA cannot be applied to SC N-FINDR in which case the latter needs an algorithm that can produce a good set of p data sample vectors for its initialization.

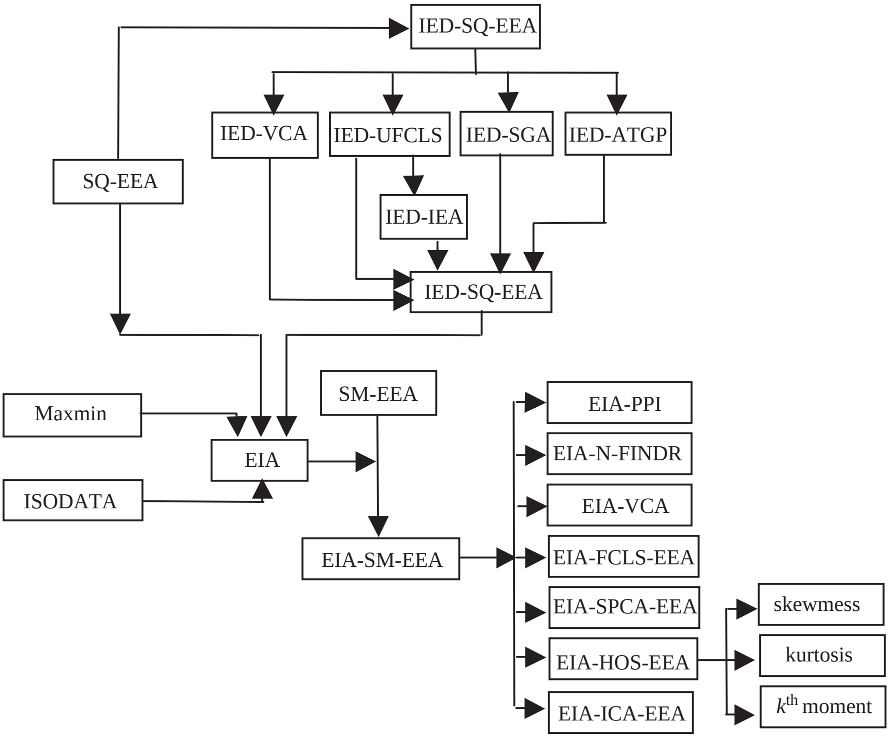


Figure 9.6 Block diagram of various ID-EEAs.

9.4 Experiments

The experiments presented in this section continue to repeat the same experiments as presented in Section 7.5 and Section 8.7 to ensure that a fair comparative analysis on performance evaluation can be conducted among SM-EEAs in Chapter 7, SQ-EEAs in Chapter 8, and ID-EEAs in this chapter where the same three sets of data, synthetic images, Cuprite image, and HYDICE image are also used for experiments. Two categories of ID-EEAs, IED-SQ-EEAS and EIA-EEAs, are evaluated for performance analysis. In the category of IED-SQ-EEAs, IED-VCA, IED-SGA, IED-UFCLS, IED-ATGP, IED-HOS-EEA, and IED-ICA-EEA are considered for evaluation. The category of EIA-EEAs is further divided into two classes, EIA-SQ-EEAs and EIA-SM-EEAs, where four EIAs, IED-ATGP (or ATGP), IED-UFCLS, Maxmin, and ISODATA, are used to initialize two major SM-EEAs, PPI and N-FINDR, and three SQ-EEAs of interest, ATGP-VCA, ATGP-HOS-EEA, and ATGP-ICA-EEA. Several remarks are noteworthy:

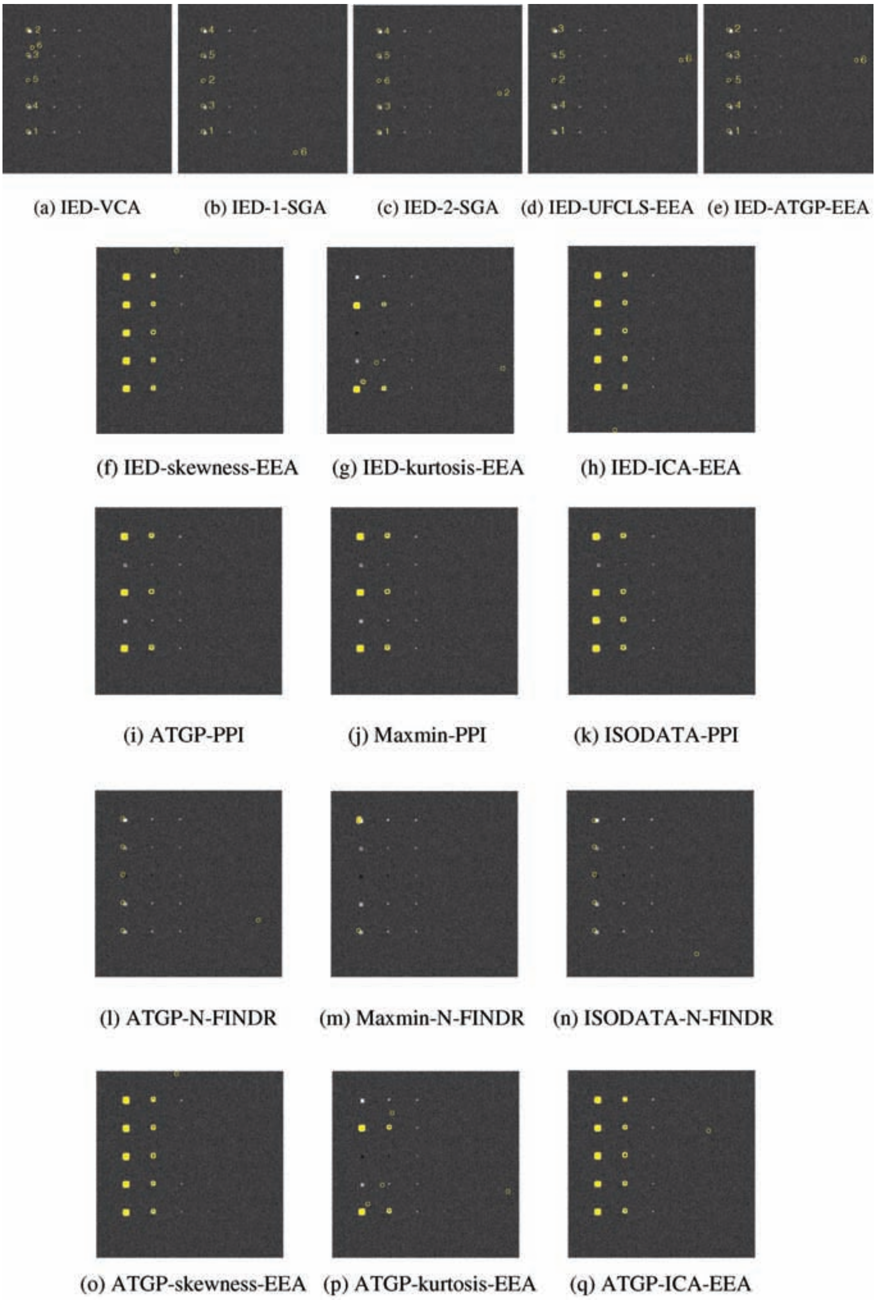
1. It should be noted that VCA, HOS-EEAs, and ICA-EEA can be considered as PP-EEAs where projection vectors specified by projection indexes that point to the direction of endmembers are generated successively. These EEAs are SQ-EEAs which start with a random initial projection vector every time they search for a new endmember. Therefore, there are two ways to initialize such PP-EEAs. One is IED-PP-EEA, which only initializes the initial projection vector one at a time. The other is EIA-PP-EAAs, which use an EIA to provide PP-EEAs with a set of p initial projection vectors where p is the number of new searches for p endmembers.
2. It is also noted that any SQ-EEA developed in Chapter 8 can also be used as an EIA for the purpose of initialization. Specifically, Maxmin and ISODATA cannot be used as EIA because they are developed as spatial domain-based clustering techniques.
3. Since ATGP-EEA, UNCLS-EEA, and UFCLS-EEA only require the first random initial endmember to be initialized, there is no EIA version for their counterparts. However, they do have EIA-FCLS-EEA because FCLS-EEA is an SM-EEA.

9.4.1 Synthetic Image Experiments

The experiments presented in this section follow the ones presented in Sections 7.5 and 8.7 so that a comparative analysis on performance evaluation among SM-EEAs in Chapter 7, SQ-EEAs in Chapter 8, and ID-EEAs in this chapter can be conducted by comparing their respective experimental results. The six scenarios TI1, TI2, TI3 and TE1, TE2, TE3 are used to evaluate various ID-EEAs with two categories: (1) IED EEAs: IED-VCA, IED-1-SGA, IED-2-SGA, IED-UFCLS-EEA, IED-ATGP-EEA, IED-skewness-EEA, IED-kurtosis-EEA, IED-ICA-EEA and (2) EIA-driven EAAs: ATGP-PPI, Maxmin-PPI, ISODATA-PPI, ATGP-N-FINDR, Maxmin-N-FINDR, ISODATA-N-FINDR, ATGP-skewness-EEA, ATGP-kurtosis-EEA, ATGP-ICA-EEA. It should be noted that there are no experiments conducted for HOS-based EEA, skewnes-EEA, kurtoiss-EEA, and ICA-EEA on TI1 and TE1 scenarios because these HOS-based EEAs require no noise to work. Since six spectral signatures are used to simulate these six scenarios, $p = 6$ is used in all the following experiments.

The results in Figure 9.6 show that when there is no noise simulated in TI1 with clean panel pixels and image background, the second-order statistics-based EEAs extracted all five panel pixels as endmembers and outperform all other EEAs including PPI and N-FINDR.

According to the results in Figure 9.7, three interesting observations are worth being mentioned. First, PPI did not work even when an EIA tried to find an appropriate set of initial endmembers, but this EIA approach did work for N-FINDR. This is mainly due to the fact that PPI required a large



(a) IED-VCA (b) IED-1-SGA (c) IED-2-SGA (d) IED-UFCLS-EEA (e) IED-ATGP-EEA

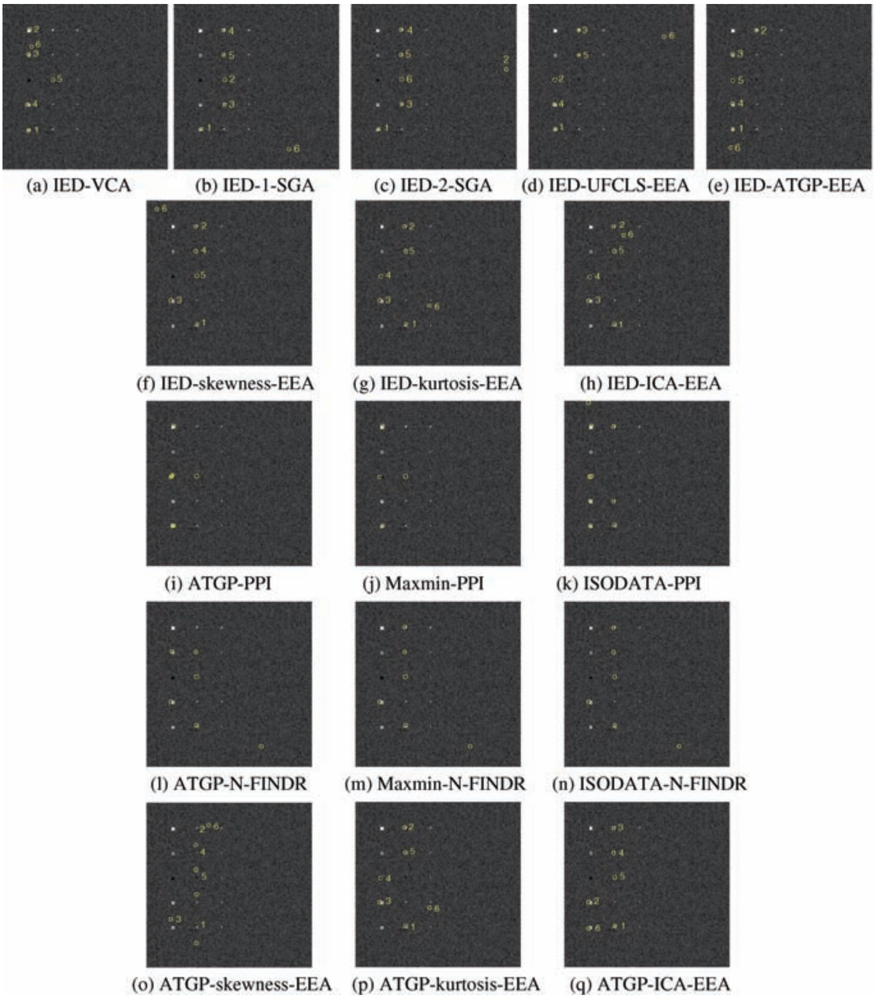
(f) IED-skewness-EEA (g) IED-kurtosis-EEA (h) IED-ICA-EEA

(i) ATGP-PPI (j) Maxmin-PPI (k) ISODATA-PPI

(l) ATGP-N-FINDR (m) Maxmin-N-FINDR (n) ISODATA-N-FINDR

(o) ATGP-skewness-EEA (p) ATGP-kurtosis-EEA (q) ATGP-ICA-EEA

Figure 9.7 Results for the TI2 scenario with six endmembers extracted by ID-EEAs.


(a) IED-VCA (b) IED-1-SGA (c) IED-2-SGA (d) IED-UFCLS-EEA (e) IED-ATGP-EEA

(f) IED-skewness-EEA (g) IED-kurtosis-EEA (h) IED-ICA-EEA

(i) ATGP-PPI (j) Maxmin-PPI (k) ISODATA-PPI

(l) ATGP-N-FINDR (m) Maxmin-N-FINDR (n) ISODATA-N-FINDR

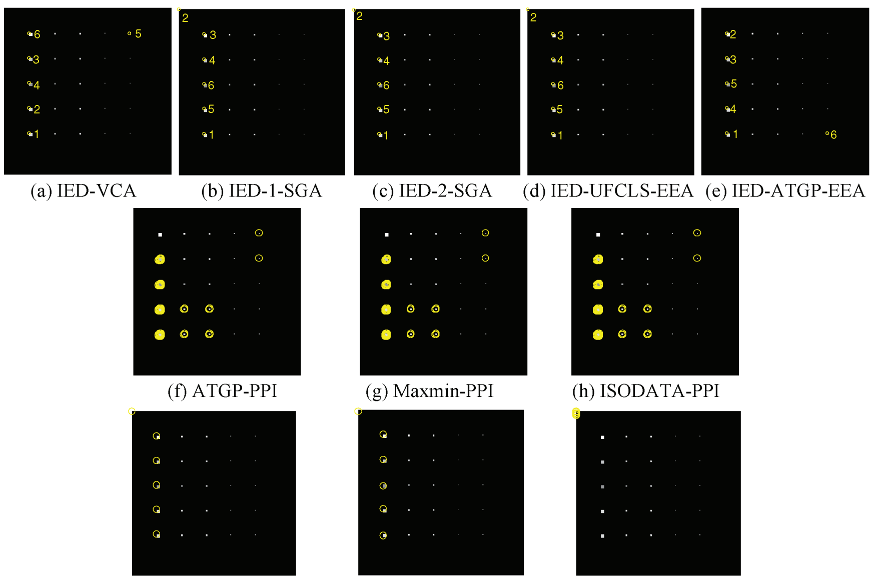
(o) ATGP-skewness-EEA (p) ATGP-kurtosis-EEA (q) ATGP-ICA-EEA

Figure 9.8 Results for the TI3 scenario with six endmembers extracted by ID-EEAs.

number of skewers to cover as many directions as it could and it did not work for small values of *p*. Second, all the second-order statistics-based EEAs performed effectively. Third, it is found that while all HOS-based EEAs worked well, only the kurtosis-based EEA failed. But this was not true for TI3 where all HOS-based EEAs did perform well.

The results in Figure 9.8 show an interesting finding, that is, except PPI and N-FINDR all other ID-EEAs worked effectively. This made sense because all the panel pixels in TI3 have been corrupted by noise and they are not of 100% purity.

Comparing the results in Figures 9.9–9.11 to those in Figures 9.1–9.3 the conclusions drawn for TI scenarios can also be applied to TE scenarios. Interestingly, despite that TE1–TE3 contain no pixels of 100% purity due to target embeddedness all the ID EEAs could successfully extract five panel pixels as endmembers in Figures 9.9–9.11 except ID-PPIs and ID-kurtosis-EEA. Once again,



(a) IED-VCA (b) IED-1-SGA (c) IED-2-SGA (d) IED-UFCLS-EEA (e) IED-ATGP-EEA

(f) ATGP-PPI (g) Maxmin-PPI (h) ISODATA-PPI

(i) ATGP-N-FINDR (j) Maxmin-N-FINDR (k) ISODATA-N-FINDR

Figure 9.9 Results for TE1 scenario with six endmembers extracted by ID-EEAs.

the reason why PPI failed was that six specific directions generated by EIAs were not enough for PPI to find all endmembers.

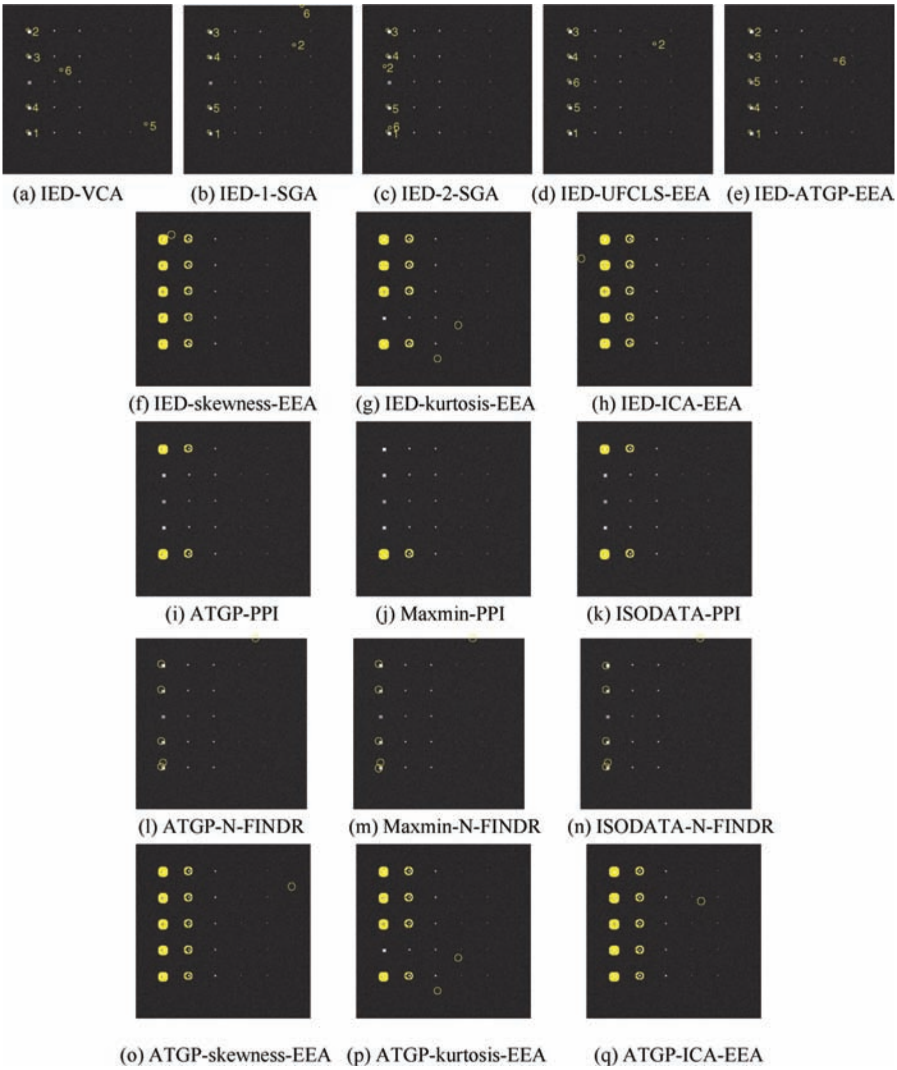
9.4.2 Real Image Experiments

The same real image data sets, the 15-panel HYDICE image scene, and Cuprite image used in Chapters 7 and 8 are also used for experiments for comparisons. Figure 9.12 shows 22 pixels extracted by various ID-EEAs where the *x*/*y* in parenthesis underneath each figure indicates that *x* is the number of endmembers corresponding to the five mineral signatures, A, B, C, K, and M, that are extracted and *y* is the mineral signature that a particular algorithm fails to extract, specifically, (5/0) indicating that all five minerals extracted as endmembers are extracted.

As shown in Figure 9.12, all tested ID-EEAs extracted at least four mineral signatures except PPI. The reason why PPI did poorly in Figure 9.12(i–k) was that the PPI required a large number of skewers to cover random directions and the number of endmembers determined by VD was too small to carry out this task unless there was a custom-designed EIA to find possible endmember directions instead of random directions.

As another example, the same HYDICE 15-panel image scene was also used for experiments. Figure 9.13 shows nine pixels extracted by various ID-EEAs where the pixels in the parenthesis underneath each figure were panel pixels in the five rows extracted by a particular algorithm.

Since most panels of interest as endmember pixels in this particular image scene have a size of one pixel or two pixels, these panel pixels cannot be either captured by spatial domain-based methods such as ISODATA in Figure 9.13(k, n) and Maxmin in Figure 9.13(j, m) or those algorithms characterized by second-order statistics such as UFCLS in Figure 9.13(d). Most interestingly, two most popular EEAs, PPI and N-FINDR, along with their sequential versions, VCA and SGA, failed to extract panel pixels that correspond to all the five endmembers. This indicated that the



(a) IED-VCA (b) IED-1-SGA (c) IED-2-SGA (d) IED-UFCLS-EEA (e) IED-ATGP-EEA

(f) IED-skewness-EEA (g) IED-kurtosis-EEA (h) IED-ICA-EEA

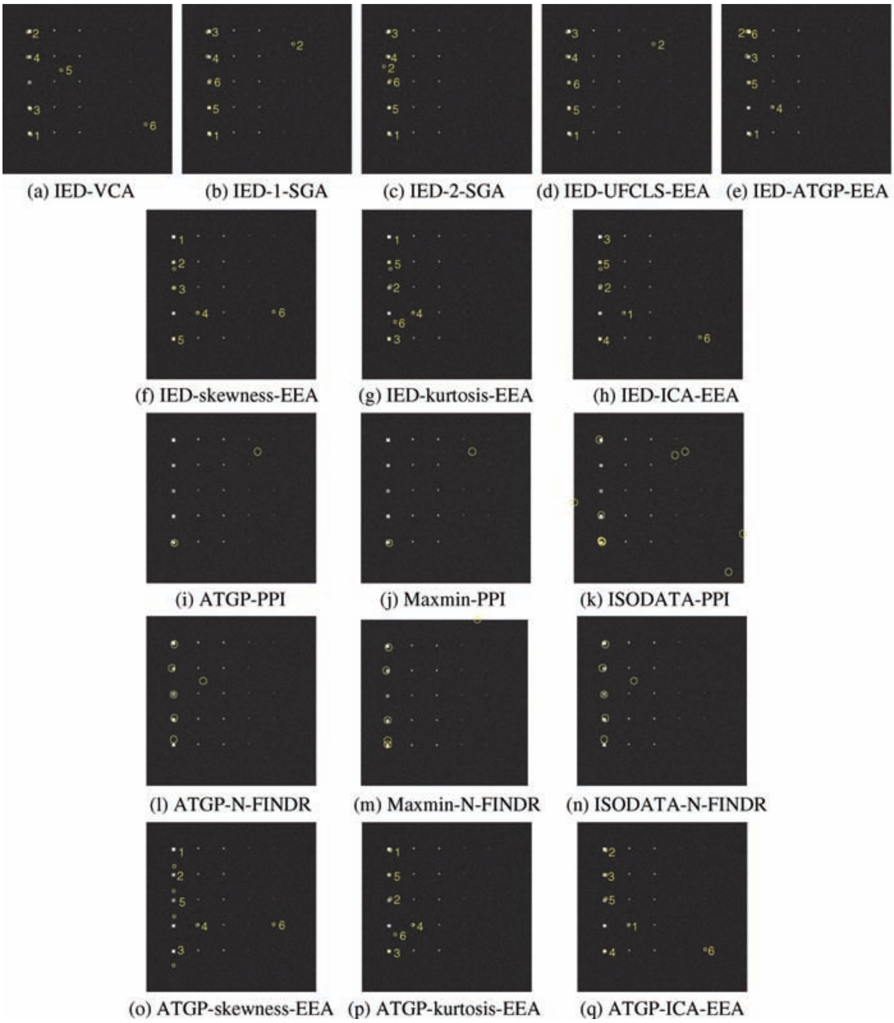
(i) ATGP-PPI (j) Maxmin-PPI (k) ISODATA-PPI

(l) ATGP-N-FINDR (m) Maxmin-N-FINDR (n) ISODATA-N-FINDR

(o) ATGP-skewness-EEA (p) ATGP-kurtosis-EEA (q) ATGP-ICA-EEA

Figure 9.10 Results for the TE2 scenario with six endmembers extracted by ID-EEAs.

effectiveness of endmember extraction was not only determined by the number of endmembers, *p*, but also by the size of an endmember sample pool. Unfortunately, these two issues are not really addressed in design and development of EEAs. Specifically, when an endmember sample size was too small such as panel pixels in the HYDICE data, PPI and N-FINDR became ineffective and broke down as demonstrated in Figure 9.13, where PPI may had tremendous difficulty in finding appropriate skewers to identify directions of such a small number of endmembers. Similarly, the same phenomenon is also applied to N-FINDR where the simplex volume of endmembers specified by panel pixels was not necessarily maximal as will be shown in Section 11.4. Under such circumstances, the desired endmembers can be only characterized by high-order statistics (HOS). The results produced by HOS-EEAs in Figure 9.13(f–h, o–q) reflected this interesting fact.



(a) IED-VCA (b) IED-1-SGA (c) IED-2-SGA (d) IED-UFCLS-EEA (e) IED-ATGP-EEA

(f) IED-skewness-EEA (g) IED-kurtosis-EEA (h) IED-ICA-EEA

(i) ATGP-PPI (j) Maxmin-PPI (k) ISODATA-PPI

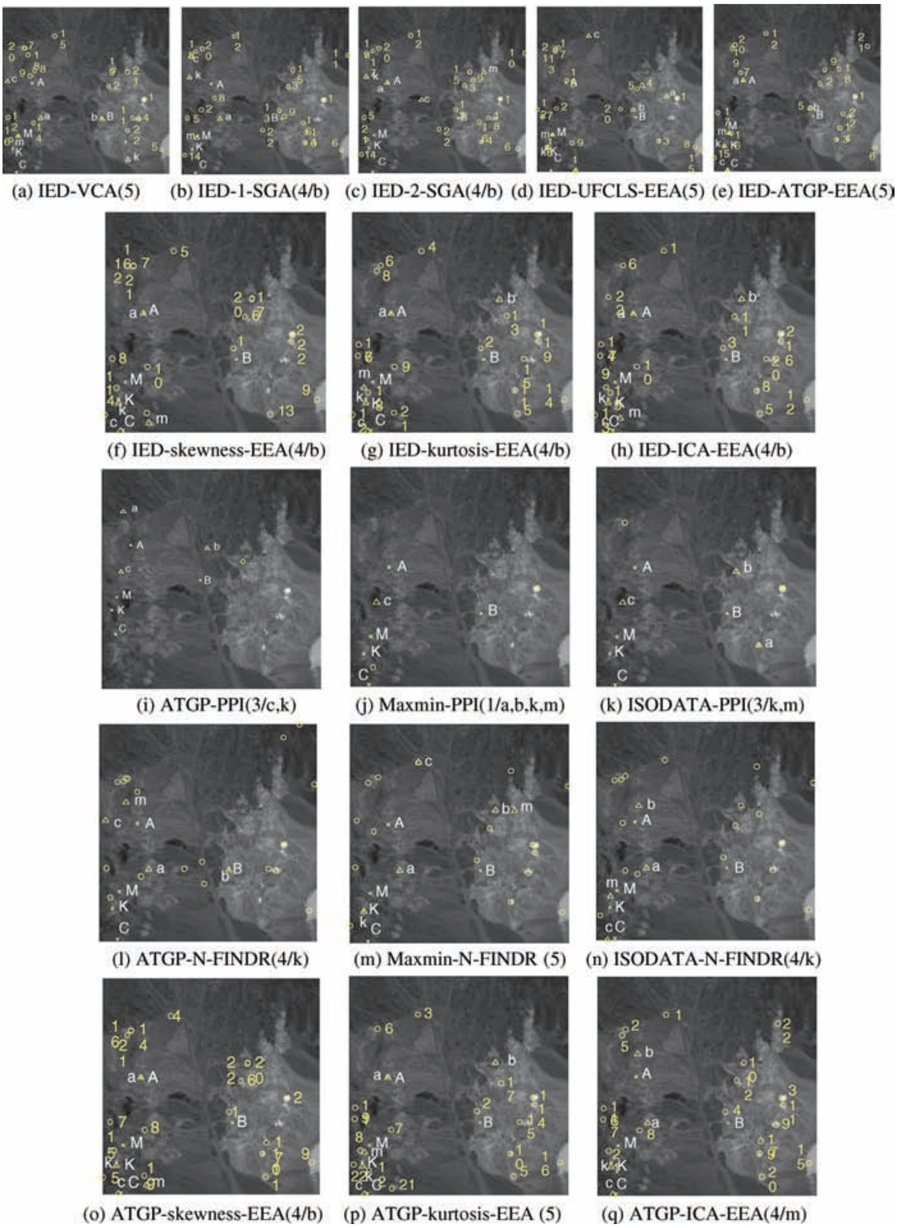
(l) ATGP-N-FINDR (m) Maxmin-N-FINDR (n) ISODATA-N-FINDR

(o) ATGP-skewness-EEA (p) ATGP-kurtosis-EEA (q) ATGP-ICA-EEA

Figure 9.11 Results for the TE3 scenario with six endmembers extracted by ID-EEAs.

9.5 Conclusions

When an algorithm, specifically an iterative algorithm, is implemented without prior knowledge, how to select an appropriate initial condition is a crucial step in convergence as well as reduction of computing time. Due to the lack of prior information a general and common practice is to use randomly generated initial conditions to initialize an algorithm. Such an approach seems natural and well accepted in algorithm design. However, associated with it three issues are needed to be addressed. First, there is a convergence issue. A bad initial condition may cause an algorithm to be divergent such as Newton’s method or to be trapped in local optimality. This issue has received considerable interest such as vector quantization. Another is an inconsistency issue. Different random initial conditions generally result in different results in the end. A good example is the ISODATA, also known as *K*-means or *C*-means clustering techniques. To resolve this dilemma, it


(a) IED-VCA(5) (b) IED-1-SGA(4/b) (c) IED-2-SGA(4/b) (d) IED-UFCLS-EEA(5) (e) IED-ATGP-EEA(5)

(f) IED-skewness-EEA(4/b) (g) IED-kurtosis-EEA(4/b) (h) IED-ICA-EEA(4/b)

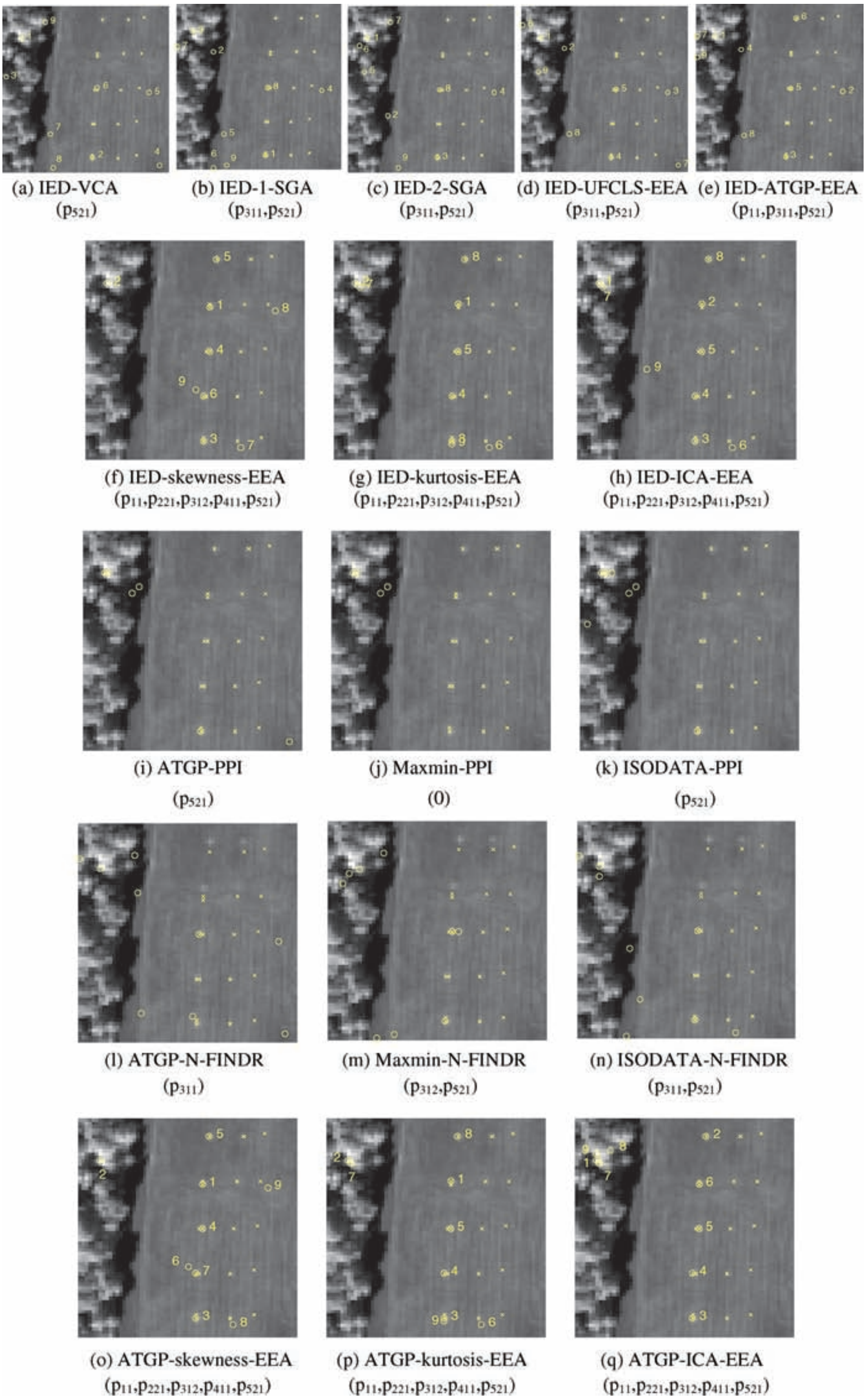
(i) ATGP-PPI(3/c,k) (j) Maxmin-PPI(1/a,b,k,m) (k) ISODATA-PPI(3/k,m)

(l) ATGP-N-FINDR(4/k) (m) Maxmin-N-FINDR (5) (n) ISODATA-N-FINDR(4/k)

(o) ATGP-skewness-EEA(4/b) (p) ATGP-kurtosis-EEA (5) (q) ATGP-ICA-EEA(4/m)

Figure 9.12 Results for cuprite scene with 22 pixels extracted by ID-EEAs.

generally runs an algorithm with a number of different random initial conditions and their results are then averaged to be used for final results. This does not guarantee that the obtained results are the desired results. The third and last issue is high computational complexity. A search using a random initial condition can take a long journey to find its destiny. An appropriate initial condition can facilitate a searching process and cut convergence rate significantly. Unfortunately, all these three issues have not been investigated for EEAs over the past years until recently (Plaza and



(a) IED-VCA (p_{521})

(b) IED-1-SGA (p_{311},p_{521})

(c) IED-2-SGA (p_{311},p_{521})

(d) IED-UFCLS-EEA (p_{311},p_{521})

(e) IED-ATGP-EEA (p_{11},p_{311},p_{521})

(f) IED-skewness-EEA (p_{11},p_{221},p_{312},p_{411},p_{521})

(g) IED-kurtosis-EEA (p_{11},p_{221},p_{312},p_{411},p_{521})

(h) IED-ICA-EEA (p_{11},p_{221},p_{312},p_{411},p_{521})

(i) ATGP-PPI (p_{521})

(j) Maxmin-PPI (0)

(k) ISODATA-PPI (p_{521})

(l) ATGP-N-FINDR (p_{311})

(m) Maxmin-N-FINDR (p_{312},p_{521})

(n) ISODATA-N-FINDR (p_{311},p_{521})

(o) ATGP-skewness-EEA (p_{11},p_{221},p_{312},p_{411},p_{521})

(p) ATGP-kurtosis-EEA (p_{11},p_{221},p_{312},p_{411},p_{521})

(q) ATGP-ICA-EEA (p_{11},p_{221},p_{312},p_{411},p_{521})

Figure 9.13 Results for the HYDICE scene with nine pixels extracted by ID-EEAs.

Chang, 2006). This chapter addresses the issue in the use of random initial endmembers and further derives ID-EEAs which implement specific initial endmembers for EEAs developed in Chapters 7 and 8. Since there are two types of EEAs, that is, SM-EEAs in Chapter 7 and SQ-EEAs in Chapter 8, ID-EEAs are also considered by two types. One is IED-EEAs developed for SQ-EEAs where a random initial endmember is specified by one at a time successively. The other is EIA-EEAs developed for an EEA such as SM-EEA which requires an EIA to produce *p* specific initial endmembers to initialize the algorithm where *p* is the number of endmembers for an EEA to generate. In both types of ID-EEAs, one needs to know the value of *p*. This issue can be resolved by taking advantage of VD introduced in Chapter 5. According to our experiments, VD provides a good estimate of the number of endmembers. Experimental results reveal that an EIA not only speed up algorithm performance, but also produces a set of initial endmembers, many of which turn out to be final endmembers.

10

Random Endmember Extraction Algorithms (REEAs)

The initialization-driven endmember extraction algorithms (ID-EEAs) developed in Chapter 9 intend to address the issue arising in the use of random initial endmembers, which causes inconsistency in final results. Interestingly, this disadvantage can become an advantage if we look at this issue from a different point of view. This chapter presents a rather different approach that allows an EEA to take advantage of the randomness to produce a set of desired endmembers. The idea is to implement an EEA using random initial endmembers as a random algorithm where a single run of a random EEA is defined as a process of running an EEA using a set of random initial endmembers and the results produced by a single run are considered as a realization. If there is an endmember, it should appear in realizations regardless of what random initial endmembers are used. In light of this interpretation, taking the intersection of realizations eventually converges to a common set made up of the desired endmembers in which case an EEA is terminated and the number of endmembers is then automatically determined by this set without appealing for any criterion. An EEA implemented in such a manner is called random EEA (REEA).

10.1 Introduction

In Chapters 7 and 8, we have witnessed the inconsistency in final results produced by an EEA using random initial conditions. This dilemma can be resolved by ID-EEAs developed in Chapter 9, which use a set of specific initial endmembers generated by a custom-designed initialization algorithm. This chapter investigates an idea completely opposite to that used in ID-EEAs. It makes the disadvantage of using random initial endmembers an advantage for an EEA. Its idea originates from the concept of a random variable where realizations resulting from physical experiments conducted for a random variable constitute an ensemble of outcomes produced by the random variable. The following example will shed light on how an EEA using random initial conditions can be considered as a random algorithm.

Now consider a problem of estimating the bias of a coin, which is defined as the probability of a head turned up, θ. Then a process of flipping the coin in a fixed N trials (i.e., N times) can be considered a random algorithm where the randomness is caused by the uncertainty with the bias

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

specified by the parameter θ. In order to estimate the value of θ, we flip the coin N times, which produces N outcomes, each of which is either a head or a tail. So, a single run resulting from such a N-coin flipping experiment produces a realization which consists of N outcomes. Then the parameter θ can be estimated from a realization by an estimator $\hat{\theta}(\mathbf{y}) = n_{\mathrm{H}}(\mathbf{y})/N$ where $\mathbf{y} = (y_1, y_2, \ldots, y_N)$ represents N outcomes resulting from N coin flips with y_i being either a head or a tail and $n_{\mathrm{H}}(\mathbf{y})$ is the number of heads turned up in the occurrence of N outcomes, $y_1, y_2, \ldots, y_N$, specified by $\mathbf{y}$. Let the number of runs implemented by the random algorithm be denoted by n and the random algorithm be represented by the estimator$\hat{\theta}(\mathbf{y})$; then$\hat{\theta}(\mathbf{y}^{(n)})$ indicates the nth run by the random algorithm$\hat{\theta}(\mathbf{y})$. As a result, the average of a total of n runs by $\hat{\theta}(\mathbf{y})$ converges to θ, that is, $(1/n)\sum_{k=1}^{n} n_{\mathrm{H}}\left(\hat{\theta}^{(k)}(\mathbf{y}^{(k)})\right) \to \theta$ as $n \to \infty$.

In light of the above interpretation, we can view an EEA that uses random initial endmembers as a random endmember extraction algorithm (REEA) where a single run is defined by implementing an REEA using one set of random initial endmembers and the result produced by a single run is considered as one realization. In this case, all realizations constitute an ensemble of final endmembers extracted by a REEA using all possible sets of random initial endmembers. According to the asymptotic equipartition property theorem in information theory (Cover and Thomas, 1991), a realization that contains final endmembers can be considered a typical realization so that the probabilities of such typical realizations will approximately be equally likely. This suggests that the average of a total number of n runs can be interpreted as an intersection of all n runs. The underlying assumption is that if a certain piece of information carried by a random variable is crucial, this information will be preserved in its realizations. Keeping this in mind, if the information of endmembers is considered as important information, the endmembers should appear in the final selected set of endmembers in realizations regardless of what initial endmembers are used for algorithm initialization. In other words, if an EEA is implemented repeatedly with different sets of random initial endmembers, all the resulting realizations shall eventually converge to a common set, which should be the desired set of the true endmembers. That is, the intersection of the final endmembers generated by all different runs should include true endmembers in the set. Using this criterion, an REEA is terminated when the final sets of endmembers produced by consecutive runs converge to the same set of endmembers in which case an REEA automatically finds endmembers without knowing the value of p *a priori*, a requirement for most EEAs reported in the literature. But this price is also traded for high computational complexity, which is a result of running an EEA multiple times. However, given that it is nearly impossible to know an exact number of endmembers in real-world problems, this paid-off may be worthwhile.

This chapter extends the two most popular SM-EEAs, pixel purity index (PPI) and N-FINDR, in Chapter 7, along with their two sequential counterparts, VCA and SGA, in Chapter 8, to their random algorithm counterparts referred to as random PPI (RPPI), random N-FINDR (RN-FINDR), random VCA (RVCA), and random SGA (RSGA), respectively. Using a similar approach, all the statistics-based and linear-spectral-unmixing-based EEAs can also be extended to their random versions.

10.2 Random PPI (RPPI)

In order to implement PPI in Section 7.2.1, first, one must know how many skewers, K, are needed to be generated. This value of K is generally determined an empirical basis. Second, it also needs to know how many dimensions, denoted by q to be retained after dimensionality reduction. Third, despite the fact that PPI does not need the knowledge of the number of endmembers, p, to be generated, it does require a parameter t to be used to threshold PPI counts it generates to extract endmembers. Finally, it requires human intervention to manually select final endmembers from those data sample vectors whose PPI counts are thresholded by t. Most importantly, for PPI to work effectively, the skewers must also be generated in such a random manner that the skewers can cover as many directions on which the data samples are projected as possible. However, this practice also comes with a drawback, that is, the results obtained from different sets of same-number skewers are different. Consequently, it must be interpreted by human analysts who manipulate results to produce best possible sets of endmembers.

The RPPI presented in this section is originally derived from its earlier version, called the automatic PPI (APPI), developed in Chaudhry et al. (2006). It also inherits the original structure of PPI but remedies the above-mentioned drawbacks suffered in PPI. It does not require dimensionality reduction (DR) as does the APPI. More specifically, RPPI converts the disadvantage of using random vectors as skewers to an advantage that can surprisingly resolve all the above-mentioned issues except that it still needs to know the value of K that must be selected prior to PPI implementation. The idea can be illustrated by the coin flipping example described in the introduction. If each of the N trials conducted for the coin flipping experiment is considered as a skewer, the number of N trials is the same as the number of skewers, K. If we further interpret that the outcome of a head turned up is the outcome of a data sample vector $\mathbf{r}$ falling at one of two ends of a skewer, then the number of heads turned up in the N trials, n_{H}, can be interpreted as the number of skewers on which a data sample vector occurred at one of their end points after orthogonal projection. This number is exactly the PPI count of the data sample vector $\mathbf{r}$, $n_{\mathrm{PPI}}(\mathbf{r})$, defined by (7.1). Within this context, we can interpret that PPI using K skewers is a random experiment carried out by a random algorithm that makes use of K randomly generated skewers. In this case, a single run produced by PPI using one random set of skewers can be considered as a realization of such a random algorithm. The underlying assumption is that if there is an endmember present in the data, it should appear in every realization produced by PPI. If this is not true for one set of skewers used by PPI, a user who happens to generate this particular set of skewers may not be able to find this endmember after all. If PPI is implemented repeatedly using different sets of random initial endmembers, all the resulting realizations shall eventually converge to a common set, which is supposed to be the set of the true endmembers. As a result, the intersection of the final endmembers generated by all different runs should include all the true endmembers. PPI is then terminated when the final sets of endmembers produced by consecutive runs converge to the same set of endmembers. Accordingly, PPI automatically finds endmembers without appealing for any criterion to determine the p. PPI implemented in such a random fashion is called RPPI, which can be described as follows.

RPPI Algorithm

1. Let K be the number of skewers to be fixed.
2. Initially, set $k = 0$ and generate a set of K skewers, denoted by $K^{(0)}$. Run PPI using $K^{(0)}$ to produce an initial endmember set, $E^{(0)} = S^{(0)} = \{\mathbf{r}|N_{\mathrm{PPI}}(\mathbf{r}) > 0\}$ defined by a set made up of all data sample vectors with their PPI counts greater than 0.
3. For $k \geq 1$ generate a kth skewer set of K skewers, denoted by $\mathrm{SKEWER\_SET}(K)^{(k)}$, and run PPI using the $K^{(k)}$ to produce PPI counts for all data sample vectors and form an kth endmember set $S^{(k)} = \{\mathbf{r}|N_{\mathrm{PPI}}(\mathbf{r}) > 0\}$.
4. Let $E^{(k)} = \cap_{m=1}^{k} S^{(m)} = \left(\cap_{m=1}^{k-1} S^{(m)}\right) \cap S^{(k)} = E^{(k-1)} \cap S^{(k)}$ and $k \leftarrow k + 1$. If $k < 3$, go to step 3. Otherwise, continue.
5. If $E^{(k)} \not\subset E^{(k-1)}$, go to step 3. Otherwise, the algorithm is terminated. In this case, $E^{(k)} = E^{(k-1)}$ and the sample pixel vectors in $E^{(k)}$ are the desired true endmembers.

It should be noted that for a given set of K skewers, a cycle of implementation of steps 3–5 is considered as a single run for PPI.

There are four advantages of using RPPI over PPI. One is that RPPI automatically determines the actual number of endmembers, p. Second, the number of pixels corresponding to endmembers extracted by RPPI is generally much smaller than that extracted by PPI. Third, there is no need of performing dimensionality reduction. Finally and most importantly, RPPI does not use a threshold value to threshold PPI count of each data sample vector to produce endmembers as required by PPI. This is a significant advantage since determining an appropriate threshold for the PPI count is a key success for PPI. Intuitively, the higher the PPI count of a sample vector, the more likely the sample vector to be an endmember. Unfortunately, according to our extensive experiments this is generally not true, but it is true that endmembers must have their PPI counts at least greater than 0. This is the main reason that S_k in step 3 of RPPI is found by all data sample vectors $\mathbf{r}$ with $N_{\text{PPI}}(\mathbf{r}) > 0$. In the ENVI's PPI this requires an appropriate selection of a threshold value and human involvement to manually determine the final selection of endmembers. The only disadvantage of implementing RPPI is its computational complexity resulting from multiple runs by repeatedly implementing PPI.

10.3 Random VCA (RVCA)

VCA in Chapter 7 can be considered as a sequential version of PPI since it also uses OP as a criterion to find one endmember at a time by growing a series of convex hulls whose vertices can be used to identify endmembers. Because the initial endmembers used by VCA are generated by Gaussian random variables, VCA can be viewed as Gaussian VCA. However, according to experiments conducted in Chapter 7, the use of a Gaussian random variable is not crucial and can be replaced with any other random variable such as uniform random variable, which also serves as well as a Gaussian random variable does. In this case, we use a random generator to generate random initial conditions as do all the random EEAs to be designed in this chapter. The details of random VCA can be described as follows.

Random VCA Algorithm

1. Let VD to determine p. Set $E^{(1)} = 0$ and $k = 0$.
2. Apply VCA to generate p random initial endmembers, denoted by $S^{(k)} = \left\{\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}\right\}$.
3. If $k = 1$, let $k \leftarrow k + 1$ and go step 2. Otherwise, continue.
4. Find the intersection of $E^{(k)} = E^{(k-1)} \cap S^{(k)}$. It should be noted that due to spectral variation in real data, a perfect match is nearly impossible. In this case, a spectral measure such as spectral angle mapper (SAM) or SID is used to measure spectral similarity within a given tolerance.
5. If $E^{(k)} \neq E^{(k-1)}$then $k \leftarrow k + 1$ and go step 2. Otherwise, the algorithm is terminated and the endmembers in $E^{(k)}$ is the desired set of final endmembers.

10.4 Random N-FINDR (RN-FINDR)

The idea of RN-FINDR can also be illustrated by the coin flipping example described in the introduction in the same way as RPPI is interpreted in Section 10.2. In this case, the number of N trials is the total number of data sample vectors. Since the randomness of N-FINDR is caused by its use of p random initial endmembers and not by the number of endmembers, which is fixed at p, the number of the heads turned up in N trials used by the coin flipping experiment, n_{H}, is then interpreted as the number of times a data sample vector $\mathbf{r}$ extracted by N-FINDR as an endmember, denoted as $n_{\text{N-FINDR}}(\mathbf{r})$ where each trial represents one single run and N trials indicates that

N-FINDR has run N times. So, a single run is defined by implementing N-FINDR using one random set of p initial endmembers and the set of its final p extracted endmembers is called a realization of N-FINDR resulting from a random set of p initial endmembers. An algorithm implementing N-FINDR in such a manner is RN-FINDR.

Random N-FINDR (RN-FINDR)

1. Initialization
   Assume that n_{VD} is the value estimated by VD and p. Let $k = 0$ denote a counter to dictate the number of runs required to implement N-FNDR and $E^{(1)} = 0$ and $n^{(0)}_{\text{N-FINDR}}(\mathbf{r}) = 0$ for all data sample vectors, $\mathbf{r}$, and let $k = 1$.
2. At the kth run, apply N-FINDR on the sphered data cube to generate p random endmembers, denoted by $S^{(k)} = \left\{\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}\right\}$.
3. Calculate N-FINDR count, $N_{\text{N-FINDR}}(\mathbf{r})$, for each data sample vector $\mathbf{r}$ by
$$n^{(k)}_{\text{N-FINDR}}(\mathbf{r}) = \begin{cases} n^{(k-1)}_{\text{N-FINDR}}(\mathbf{r}) + 1 & \text{if } \mathbf{r} \in S^{(k)} \\ n^{(k-1)}_{\text{N-FINDR}}(\mathbf{r}) & \text{if } \mathbf{r} \notin S^{(k)} \end{cases}. \tag{10.1}$$
4. If $k = 1$, let $k \leftarrow k + 1$ and go to step 2. Otherwise, continue.
5. Find the intersection of $E^{(k)} = E^{(k-1)} \cap S^{(k)}$. It should be noted that due to spectral variation in real data, a perfect match is nearly impossible. In this case, a spectral measure such as SAM is used to measure spectral similarity within a given tolerance.
6. Stopping rule
   If $E^{(k)} \neq E^{(k-1)}$ then $k \leftarrow k + 1$ and go to step 2. Otherwise, the algorithm is terminated and the endmembers in $E^{(n)}$ are the desired set of endmembers. In this case, let n denote $n_{\text{N-FINDR}}$ and the desired endmember set is $E^{(k)}_{\text{N-FINDR}} = \left\{\mathbf{r} | n^{(k)}_{\text{N-FINDR}}(\mathbf{r}) = k\right\}$.

Several remarks are noteworthy.

i. It should be noted that RN-FINDR does not require VD to estimate the value of the p, that is, the number of endmembers. This p is ultimately determined by step 6 automatically. However, we can always use the knowledge provided by VD to reduce computing time by setting $p = 2n_{\text{VD}}$. This is based on the fact that n_{VD} has been shown to be a reasonable estimate of p and using $2p$ guarantees that no endmembers will be left out in taking intersection of all runs. However, if there is no limitation on computer power, the full data dimensionality, L, can be used instead. This indicates that the upper bound on the number of endmembers, p, cannot be greater than the total number of spectral bands, L.
ii. The only disadvantage of RN-FINDR is the burden on computing time resulting from repeatedly implementing N-FINDR over and over again for different runs. This is the major reason that the $p = 2n_{\text{VD}}$ is used in the algorithm for reducing computational complexity.
iii. RN-FINDR does not require dimensionality reduction. In this case, the determinant used in (7.3) to calculate the simplex volume becomes ill-rank in which case the pseudo-inverse is used for this purpose.
iv. Although the data on which RN-FINDR operates are the sphered data, it can also be applied to the original data. According to our experiments, the results obtained for both sets of data are similar and close. However, using the sphered data significantly reduces computing time as well as the number of runs required for IN-FINDR to implement.

v. In order to further reduce computational burden, N-FINDR implemented in RN-FINDR can be replaced by IN-FINDR, SQ N-FINDR, or SC N-FINDR in step 2. The resulting N-FINDR is called random IN-FINDR (RIN-FINDR), random SQ N-FINDR (RSQ N-FINDR), and random SC N-FINDR (RSC N-FINDR). There is a computational advantage of using SC N-FINDR over IN-FINDR because RSC N-FINDR works in the same way as the outer loop of IN-FINDR does except that IN-FINDR re-runs SC N-FINDR using the p endmember generated in a previous run as its initial endmembers, whereas RSC N-FINDR re-runs SC N-FINDR using a complete different set of random p initial endmembers. As a result, IN-FINDR still depends upon the random p initial endmembers used by SC N-FINDR in its inner loop, while RSC N-FINDR does not. This is because RSC N-FINDR already takes care of the outer loop run by IN-FINDR by running another set of random initial endmembers over again in such a way that IN-FINDR implements SC N-FINDR in its inner loop and runs SC N-FINDR over again in its outer loop using the final endmembers generated in its inner loop as inputs.

10.5 Random SGA (RSGA)

What VCA is viewed as a sequential version of PPI is exactly what SGA is considered as a sequential version of N-FINDR in the sense that SGA finds one endmember at a time when it grows a series of simplexes with maximal volumes. Similarly, SGA can also be extended to its random version, called RSGA, in the same way as both RSQ N-FINDR and RSC N-FINDR are extended with SGA used in each run.

Random SGA (RSGA)

1. Initialization
   a. Assume that the number of endmembers required to be generated is p.
   b. Let ε be the given tolerance value of spectral similarity.
   c. Set $E^{(1)} = 0$ and $k = 1$.
2. Randomly generate two data sample vectors as two initial endmembers, $\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}$.
3. Apply SGA to generate p endmembers, denoted by $S^{(k)} = \left\{\mathbf{e}_1^{(k)}, \mathbf{e}_2^{(k)}, \ldots, \mathbf{e}_p^{(k)}\right\}$.
4. If $k = 2$, let $k \leftarrow k + 1$ and go to step 2. Otherwise, continue.
5. Find the intersection of $E^{(k)} = E^{(k-1)} \cap S^{(k)}$. In this case, a spectral measure such as SAM is used to measure spectral similarity. If $SAM\left(\mathbf{e}_i, \mathbf{e}_j\right) < \varepsilon$, then $\mathbf{e}_i$ and $\mathbf{e}_j$ are considered to represent the same endmember class.
6. Stopping rule
   If $E^{(k)} \neq E^{(k-1)}$, then let $k \leftarrow k + 1$ and go to step 2. Otherwise, the algorithm is terminated and the endmembers in $E^{(k)}$ is the desired set of endmembers.

It should be noted that two initial endmembers instead of one initial endmember are randomly generated in step 2 of RSGA due to the fact that the smallest dimensionality of a simplex is 2.

10.6 Random ICA-Based EEA (RICA-EEA)

The idea of RICA-EEA is derived from a recent work on an independent component analysis (ICA)-based approach to DR (Wang and Chang, 2006a) where FastICA developed by Hyvarinen and Oja (1997) is used to generate independent components (ICs). Since FastICA also uses random projection vectors as its initial condition to initialize its algorithm, it also encounters the same problem as both PPI and N-FINDR do. Because the initial projection vector for each IC is randomly generated by FastICA for each run, the ICs generated by each run are also different. Nevertheless, if the information contained in an IC is significant, such an IC will always appear in each run. With this assumption, if FastICA is repeatedly implemented, the ICs that are common in all runs should be the desired ICs for endmember extraction. The detailed implementation of RICA-EEA is summarized as follows.

RICA-EEA Algorithm

1. Initialization
   Assume that the number of initial endmembers is p. Set $n = 0$.
2. At each n, run FastICA to find p independent components, $\left\{\text{IC}_i^{(n)}\right\}_{i=1}^{p}$, where each independent component, $\text{IC}_i^{(n)}$, can be formed as a vector denoted by $\mathbf{v}_i^{(n)}$. It should be noted that FastICA randomly generates a unit vector as an initial projection vector.
3. If $n < 1$, $n \leftarrow n + 1$ and go to step 2. Otherwise, continue.
4. Find common ICs for all runs up to the nth run. Two independent components for different runs, $\text{IC}_i^{(\bar{n})}$ and $\text{IC}_j^{(\tilde{n})}$, are considered to be distinct if the SAM between their corresponding vectors,$\mathbf{v}_i^{\bar{n}}$ and $\mathbf{v}_j^{\tilde{n}}$, is greater than a prescribed threshold ε. Let $\cap_{m=0}^{n}\left\{\text{IC}_k^{(m)}\right\}_{k=1}^{p}$ denote the common ICs obtained for all runs, $0 \leq m \leq n$.
5. If $\cap_{m=0}^{n-1}\left\{\text{IC}_k^{(m)}\right\}_{k=1}^{2p} \neq \cap_{m=0}^{n}\left\{\text{IC}_k^{(m)}\right\}_{k=1}^{2p}$, go to step 2. Otherwise, the algorithm is terminated and $\cap_{m=0}^{n}\left\{\text{IC}_k^{(m)}\right\}_{k=1}^{2p}$ is the desired set of ICs for endmembers.
6. For each of FastICA-generated IC images in $\cap_{m=0}^{n}\left\{\text{IC}_k^{(m)}\right\}_{k=1}^{2p}$, find a pixel with maximum absolute value, which is referred to as endmember pixel. The spectral signature of such found pixel is then selected as an endmember.
7. The spectral signatures of the endmember pixels produced in step 6 are our desirable endmembers.

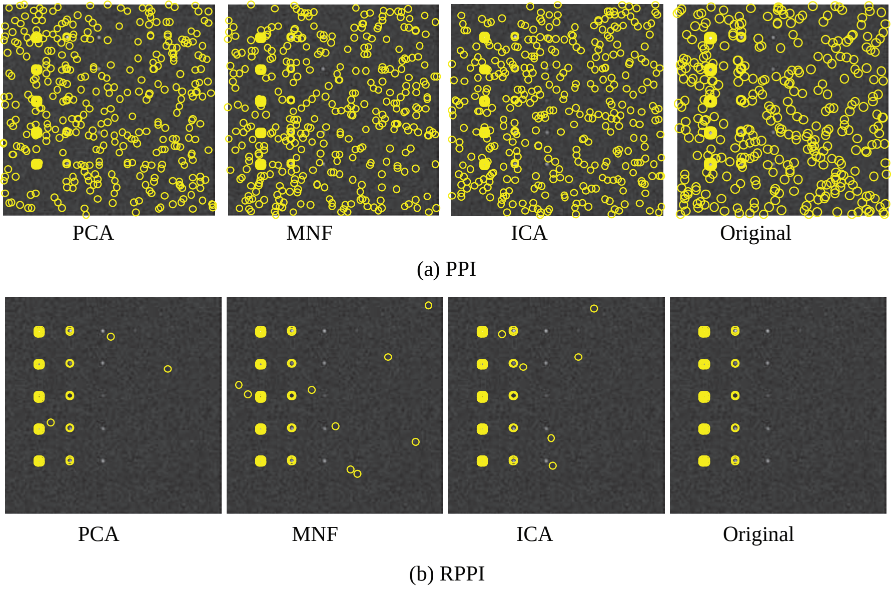
Like RPPI and RN-FINDR, VD is only used to provide a good reasonable upper bound on the number of random initial endmembers to be used for RICA-EEA.

10.7 Synthetic Image Experiments

Once again the synthetic images described in Chapter 4 were used for experiments to evaluate the performance of the proposed REEAs. Scenarios TI1 and TE1 were not included in the experiments because there is no randomness caused by noise, in which case REEAs do not work in this scenario due to the fact that they are designed as random algorithms. Three component transform techniques, PCA, MNF and ICA, were used to perform DR to reduce the original data space to a reduced data space with data dimensionality determined by VD.

10.7.1 RPPI

The experiments of RPPI were implemented by 500 skewers on two data sets, a reduced data cube by DR and the original data without DR where the data dimensionality to be retained after DR was $n_{\text{VD}} = 5$ estimated by the NWHFC method with $P_F = 10^{-4}$. Figures 10.1(a) and (b) and 10.2(a) and (b) show the results for TI2 and TI3 produced by PPI and RPPI, respectively, where pixel vectors in Figures 10.1(a) and 10.2(a) marked by open circles were extracted by PPI with PPI counts greater

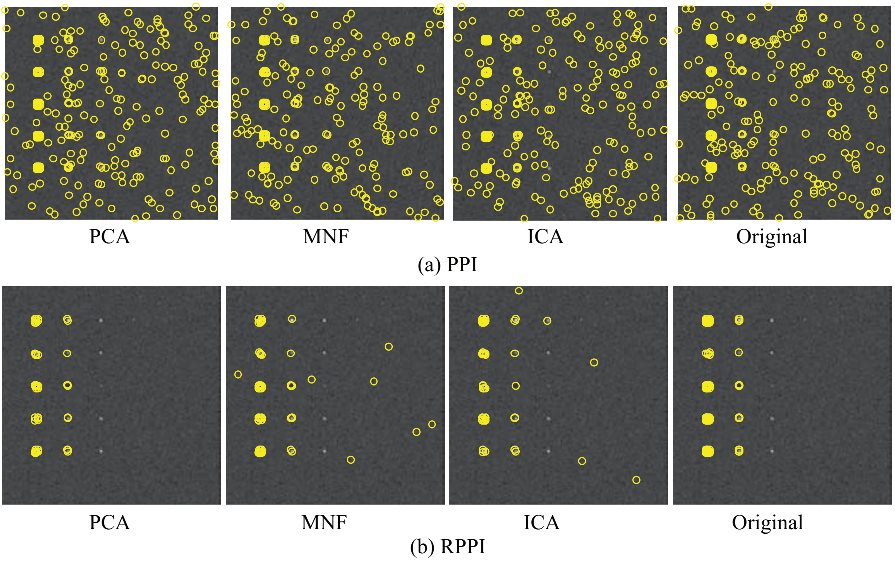


(a) PPI

(b) RPPI

Figure 10.1 Endmembers extracted by PPI and RPPI in scenario TI2.

than zero and all pixel vectors in Figures 10.1(b) and 10.2(b) were those in the intersection of all runs by RPPI. As shown in Figures 10.1 and 10.2, the number of pixels extracted by RPPI as endmembers was significantly reduced with only less than 10 falsely alarmed endmembers in Figures 10.1(a) and 10.2(a) compared to a very large number of pixels extracted by PPI in Figures 10.1(a) and 10.2(a) as endmembers with hundreds of falsely alarmed pixels.



(a) PPI

(b) RPPI

Figure 10.2 Endmembers extracted by PPI and RPPI in scenario TI3.

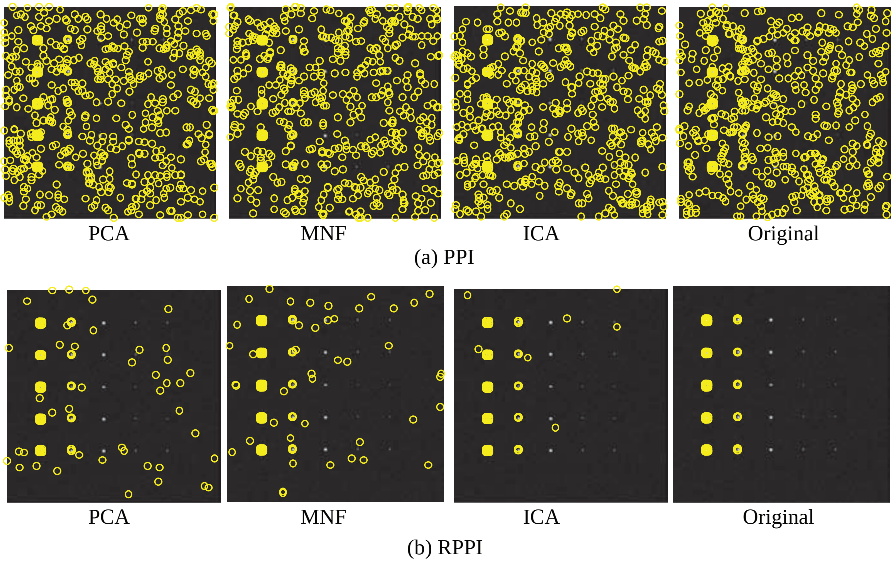


Figure 10.3 Endmembers extracted by PPI and RPPI in scenario TE2.

Similarly, Figures 10.3(a) and (b) and 10.4(a) and (b) show the results for TE2 and TE3 produced by the PPI and RPPI, respectively, where pixel vectors in Figures 10.3(a) and 10.4(a) marked by open circles were extracted by the PPI with PPI counts greater than zero and all pixel vectors in Figures 10.3(b) and 10.4(b) were those in the intersection of all runs by RPPI. Since there are no pure pixels present as endmembers in TE2 and TE3, PPI and RPPI made attempt to extract pixels that represent purest signatures. As expected, more falsely alarmed endmember pixels than those in Figures 10.1(b) and 10.2(b) were extracted as shown in Figures 10.3(b) and 10.4(b).

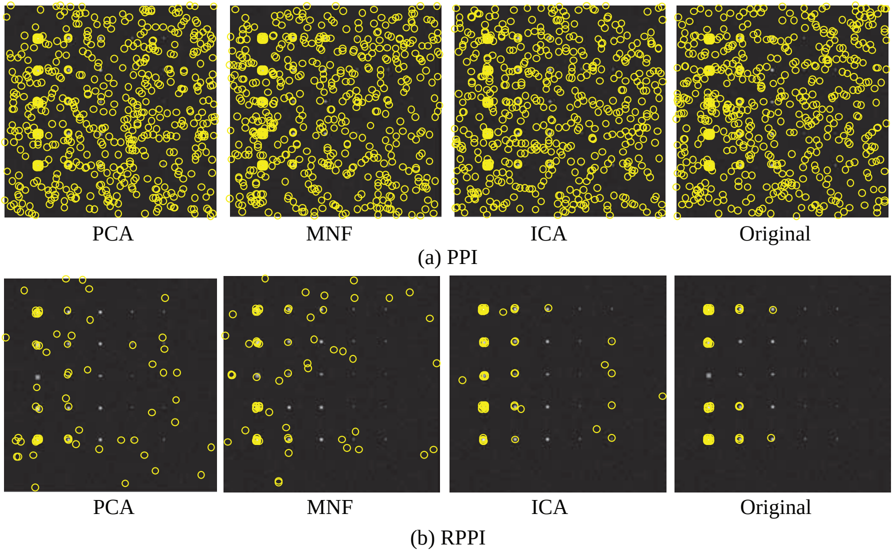


Figure 10.4 Endmembers extracted by PPI and RPPI in scenario TE3.

Table 10.1 Computing time and the number of iterations of RPPI running in different data sets

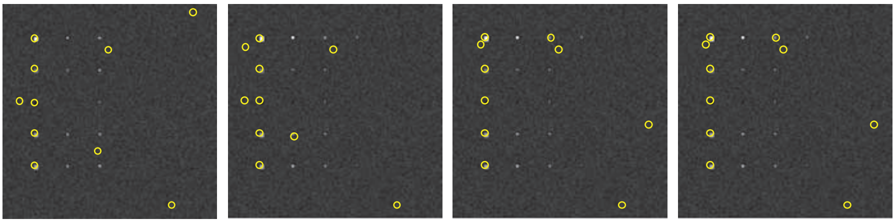
		Computing time (s)	Number of iterations
Scenario TI2	PCA	5.26	4
	MNF	5.44	4
	ICA	4.12	3
	Original	10.40	4
Scenario TI3	PCA	7.81	5
	MNF	7.77	5
	ICA	10.6	7
	Original	15.40	6
Scenario TE2	PCA	15.34	11
	MNF	19.08	13
	ICA	5.41	4
	Original	15.79	6
Scenario TE3	PCA	19.85	13
	MNF	21.64	15
	ICA	10.35	7
	Original	17.19	6

Interestingly, in the case of TE3 RPPI missed all the pure pixels in row 3 if it is implemented directly on the original data without DR. Besides, if PCA was used for DR, RPPI missed all the 16 purest panel pixels in the first column of row but it did extract panel pixels in the second column. This is mainly due to the fact that the calcite signature is very similar and close to the sample mean used to simulate the image background. As a result, the calcite was considered as a variant of the background signature and thus was not extracted as purest signature in Figure 10.4(b) by RPPI.

Table 10.1 tabulates the number of iterations and the computing time in seconds for RPPI to converge. As expected, except TE3 RPPI operating on the original data space without DR generally required the maximal computing time, while RPPI required the least amount of computing time if the ICA was used for DR.

10.7.2 Various Random Versions of IN-FINDR

A random version of N-FINDR, RN-FINDR, is quite different from RPPI presented in Section 10.6.1. First is the prior knowledge. RN-FINDR needs to know the number of endmembers, p, to be generated, which is not required for PPI but is traded off for two must-be-known parameters, the number of skewers, K, and the threshold t. Second is data to be processed. PPI generally requires DR as a preprocessing to reduce data dimensionality to compact information as well as reduce computational complexity. On the other hand, RN-FINDR does not perform DR. Instead, it spheres the data to remove the statistics of the first and second order to enhance and improve endmember extraction. In order to illustrate RN-FINDR for the performance evaluation, the experiments conducted in this section were particularly designed to address two main issues: (1) inconsistency caused by the use of two randomly generated sets of initial conditions, and (2) two different data sets to be used for processing, the original data and sphered data. Three random versions of N-FINDR were used for a comparative study, RIN-FINDR, RSC N-FINDR, and RSGA, all of which were implemented without DR. The number of endmembers required to be generated by



one run another run one run another run

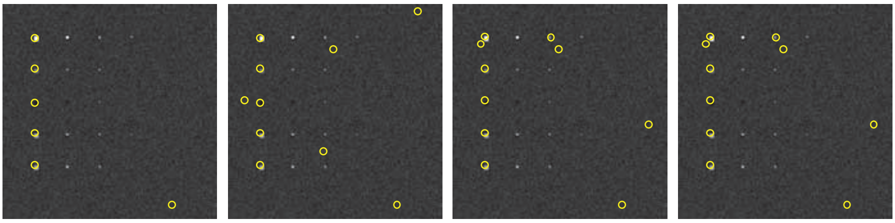
(a) Original data (b) Sphered data

Figure 10.5 Results of IN-FINDR with two different runs for TI2.

RN-FINDR for all the four synthetic image-based scenarios, TI2, TI3, TE2, and TE3, was determined by VD, which was $2n_{\mathrm{VD}} = 10$ for all the algorithms.

10.7.2.1 Scenario TI2

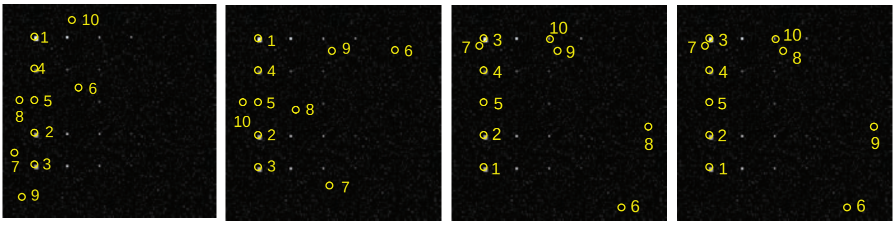
Figures 10.5–10.10 show the results of IN-FINDR, SC N-FINDR, and SGA along with their corresponding random counterparts, RIN-FINDR, RSC N-FINDR, and RSGA for TI2. There were two sets of endmembers generated in Figures 10.5, 10.7, and 10.9 by operating IN-FINDR, SC N-FNDR, and SGA on the original data and sphered data to illustrate inconsistent final results from the use of two randomly generated initial conditions. In contrast, Figures 10.6, 10.8, and 10.10, respectively, show the consistent final results of operating the RIN-FINDR, RSC N-FINDR, and



$\tau = 0.005$ $\tau = 0.05$ $\tau = 0.005$ $\tau = 0.05$

(a) Original data (b) Sphered data

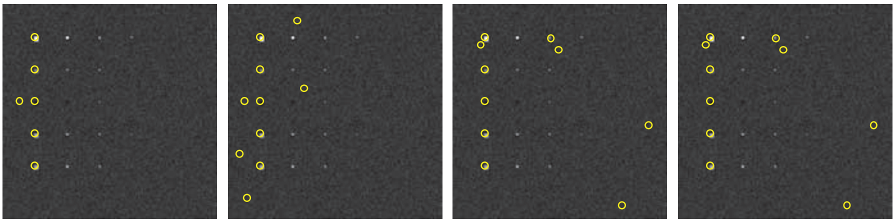
Figure 10.6 Results of RIN-FINDR with two different thresholds for TI2.



one run another run one run another run

(a) Original data (b) Sphered data

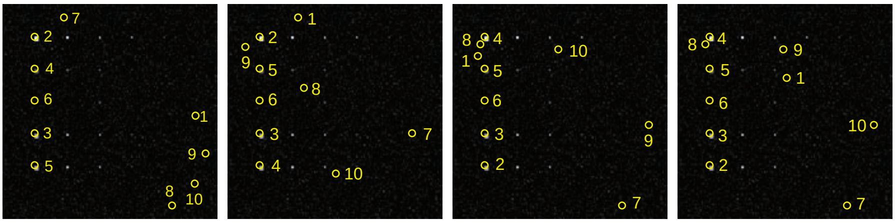
Figure 10.7 Results of SC N-FINDR with two different runs for TI2.



τ = 0.005 τ = 0.05 τ = 0.005 τ = 0.05

(a) Original data (b) Sphered data

Figure 10.8 Results of RSC N-FINDR with two different thresholds for TI2.



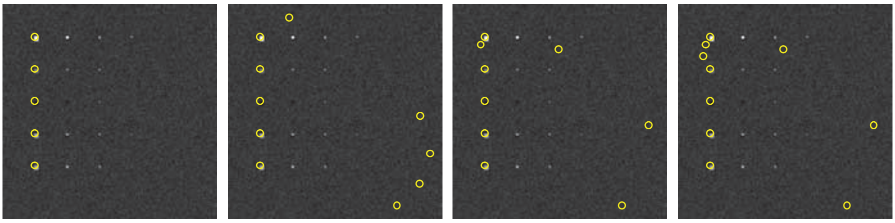
one run another run one run another run

(a) Original data (b) Sphered data

Figure 10.9 Results of SGA with two different runs for TI2.

RSGA on the original data and sphered data using two thresholds $\tau = 0.05$ and 0.005. It should be noted that since SC N-FINDR and SGA are SQ-EAAs, the endmembers were labeled by numbers to indicate the orders in which they were generated. However, there were no numbers associated with IN-FINDR-generated endmembers because these endmembers were generated iteratively at the same time.

Comparing the results in above figures, it is interesting to note that it only required the first five pixels for SC N-FINDR and the first six pixels for SGA to extract all the five mineral signatures as endmembers where the first pixels extracted by SGA were always background pixels. Nevertheless, with p set by $2n_{\mathrm{VD}} = 10$ all the algorithms successfully extracted pixels that specify all the five mineral signatures as endmembers. Table 10.2 tabulates the computing time and the number of



τ = 0.005 τ = 0.05 τ = 0.005 τ = 0.05

(a) Original data (b) Sphered data

Figure 10.10 Results of RSGA with two different thresholds for TI2.

Table 10.2 Computing time and number of iterations for RN-FINDR, RSC N-FINDR, and RSGA

		T	Computing time (s)	Number of iterations
RN-FINDR	Original	0.005	1592.8	4
		0.05	752.10	2
	Sphered data	0.005	763.84	2
		0.05	762.79	2
RSC N-FINDR	Original	0.005	259.12	3
		0.05	171.15	2
	Sphered data	0.005	174.51	2
		0.05	169.45	2
RSGA	Original	0.005	153.11	3
		0.05	101.03	2
	Sphered data	0.005	152.53	3
		0.05	101.97	2

iterations required for a random algorithm to complete its search where RSGA required the least amount of computing time which was only one-seventh of time required by RIN-FINDR. However, it should be noted that the computing time needed by REEAs was heavily determined by the number of iterations required to complete the endmember search.

10.7.2.2 Scenario TI3

Despite that the 100 pure panel pixels in TI3 have been Gaussian noise-corrupted, these pure panel pixels no longer represent pure signatures but they are still those with purest signatures. Figures 10.11–10.16 show that for $p = 2n_{\text{VD}} = 10$, IN-FINDR, SC N-FINDR, and SGA along with their random versions, RIN-FINDR, RSC N-FINDR, and RSGA were all able to extract panel pixels which represented most purest signatures corresponding to the five distinct minerals. However, the threshold value, τ, used by the random versions of N-FINDR should have a greater threshold tolerance to accommodate the added Gaussian noise such as $\tau = 0.05$. If the threshold value was set too small, e.g., $\tau = 0.005$ used for TI2, random versions of N-FINDR would miss one mineral signature, M in row 5 in Figures 10.12, 10.14, and 10.16.

Table 10.3 tabulates the computing time and the number of iterations required for a random algorithm to complete its search, where RSGA required the least amount of computing time. The computing time documented in the table was essentially determined by the number of iterations required to complete the endmember search.

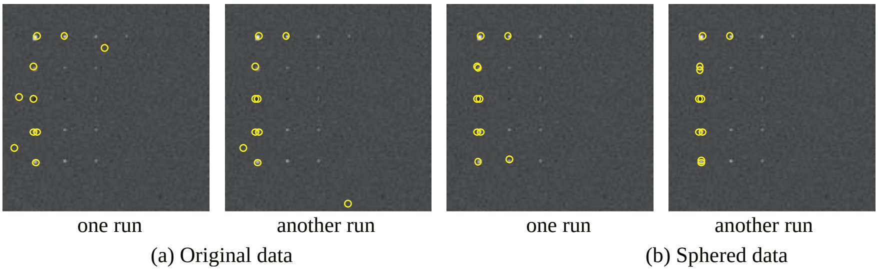
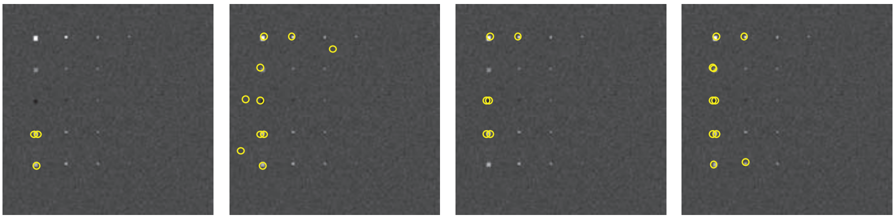


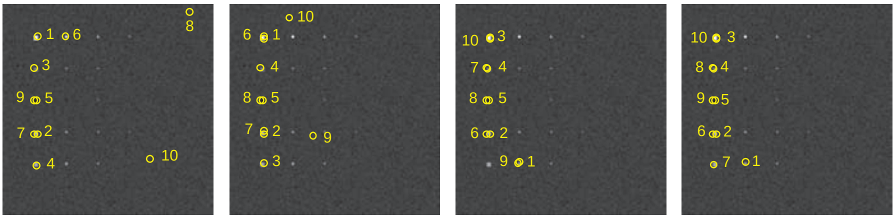
Figure 10.11 Results of IN-FINDR with two different runs.



$\tau = 0.005$ $\tau = 0.05$
(a) Original data

$\tau = 0.005$ $\tau = 0.05$
(b) Sphered data

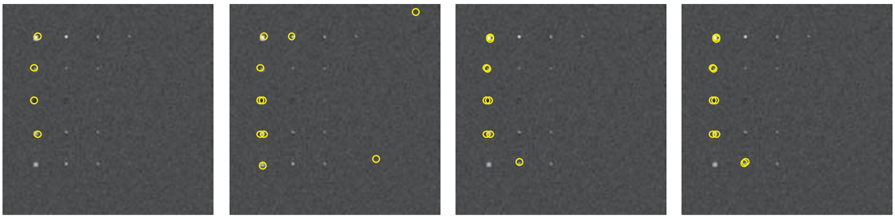
Figure 10.12 Results of RIN-FINDR with two different thresholds.



one run another run
(a) Original data

one run another run
(b) Sphered data

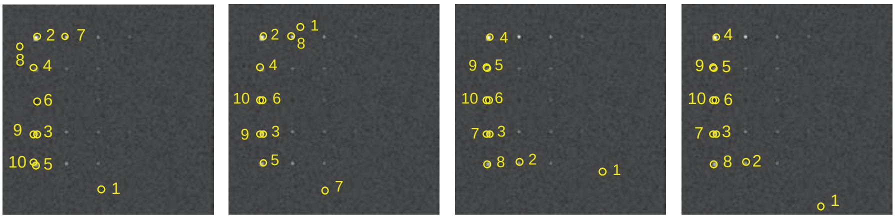
Figure 10.13 Results of SC N-FINDR with two different runs.



$\tau = 0.005$ $\tau = 0.05$
(a) Original data

$\tau = 0.005$ $\tau = 0.05$
(b) Sphered data

Figure 10.14 Results of RSC N-FINDR with two different thresholds.



one run another run
(a) Original data

one run another run
(b) Sphered data

Figure 10.15 Results of SGA with two different runs.

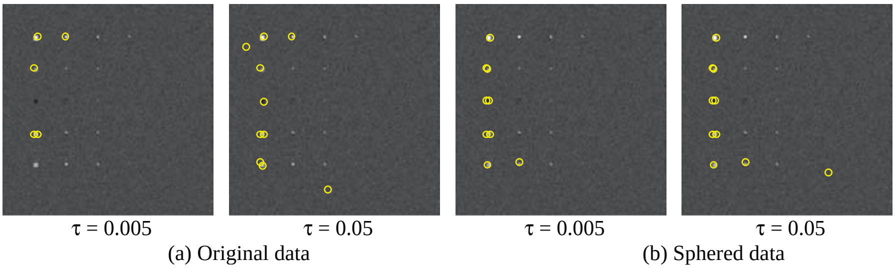


Figure 10.16 Results of RSGA with two different thresholds.

Table 10.3 Computing time and number of iterations for RIN-FINDR, RSC N-FINDR, and RSGA

		τ	Computing time (s)	Number of iteration
RIN-FINDR	Original	0.005	1669.6	5
		0.05	751.18	2
	Sphered data	0.005	1018.7	3
		0.05	679.51	2
RSC N-FINDR	Original	0.005	430.08	5
		0.05	173.43	2
	Sphered data	0.005	2510.76	3
		0.05	174.57	2
RSGA	Original	0.005	154.20	3
		0.05	102.79	2
	Sphered data	0.005	153.49	3
		0.05	103.60	2

10.7.2.3 TE2

Following the same treatment carried out for TI2, experiments were also conducted for TE2. The only difference is that the pure panel pixels are superimposed over the background pixels in which case no pure signatures are present in TE scenarios. Nevertheless, the purest pixels in TE2 were still the same as the pure panel pixels in TI2. Figures 10.17–10.22 show that all algorithms were able to extract panel pixels corresponding to the five distinct mineral signatures. Specifically, the

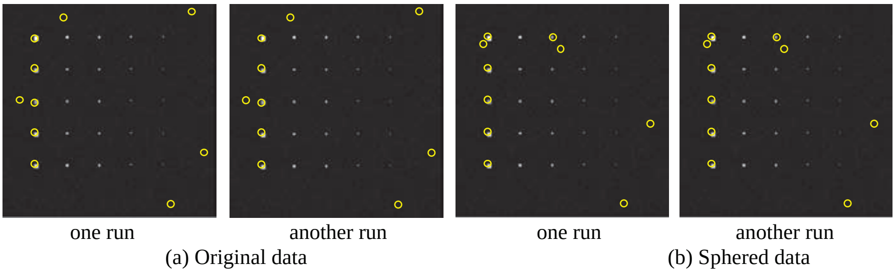
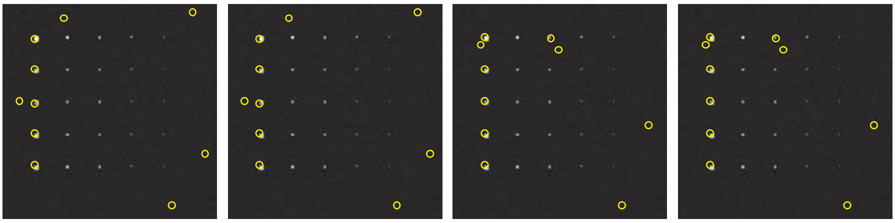


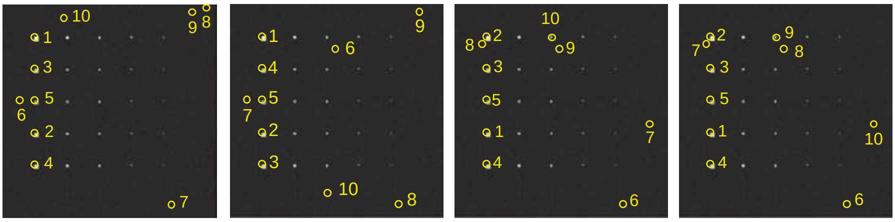
Figure 10.17 Results of IN-FINDR with two different runs.



τ = 0.005 τ = 0.05 τ = 0.005 τ = 0.05

(a) Original data (b) Sphered data

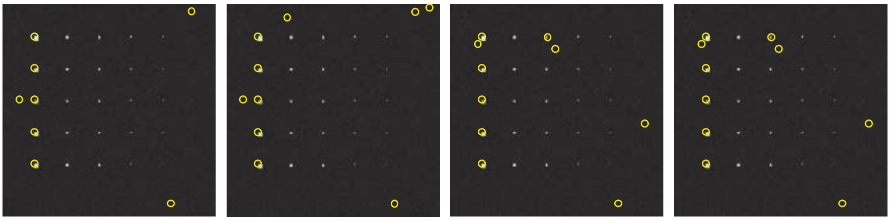
Figure 10.18 Results of RIN-FINDR with two different thresholds.



one run another run one run another run

(a) Original data (b) Sphered data

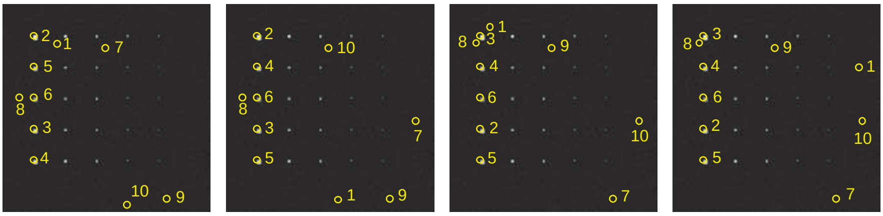
Figure 10.19 Results of SC N-FINDR with two different runs.



τ = 0.005 τ = 0.05 τ = 0.005 τ = 0.05

(a) Original data (b) Sphered data

Figure 10.20 Results of RSC N-FINDR with two different thresholds.



one run another run one run another run

(a) Original data (b) Sphered data

Figure 10.21 Results of SGA with two different runs.

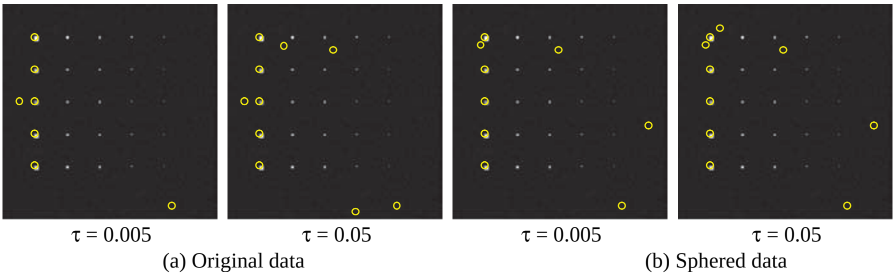


Figure 10.22 Results of RSGA with two different thresholds.

Table 10.4 Computing time and number of iterations for RN-FINDR, RSC N-FINDR, and RSGA

		τ	Computing time (s)	Number of iterations
RN-FINDR	Original	0.005	503.07	2
		0.05	505.39	2
	Sphered data	0.005	764.11	2
		0.05	680.15	2
RSC N-FINDR	Original	0.005	255.48	3
		0.05	170.37	2
	Sphered data	0.005	171.44	2
		0.05	169.37	2
RSGA	Original	0.005	201.73	4
		0.05	101.38	2
	Sphered data	0.005	151.15	3
		0.05	100.78	2

RIN-FINDR and RSC N-FINDR extracted the same identical five panel pixels that corresponded to the five distinct pure mineral signatures.

Table 10.4 tabulates the computing time and the number of iterations required for a random algorithm to complete its search where once again RSGA required the least amount of computing time.

10.7.2.4 TE3 Scenario

TE3 is an additive Gaussian noise-corrupted scenario of TE2. Figures 10.23–10.28 show the results of IN-FINDR, SC N-FINDR, and SGA along with their random counterparts. Analogous to TI3, it

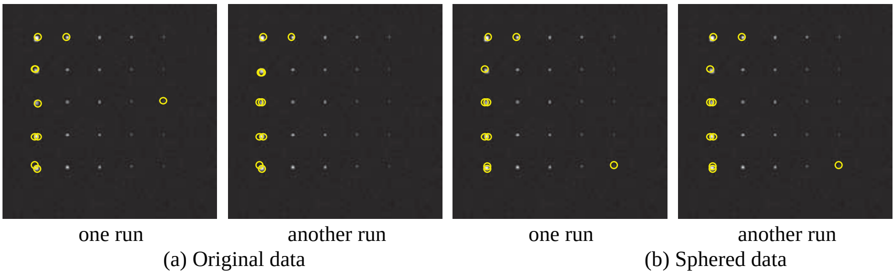
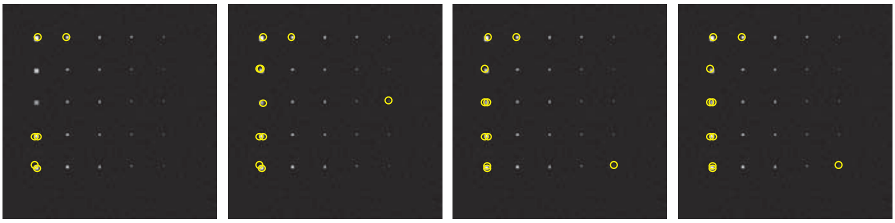


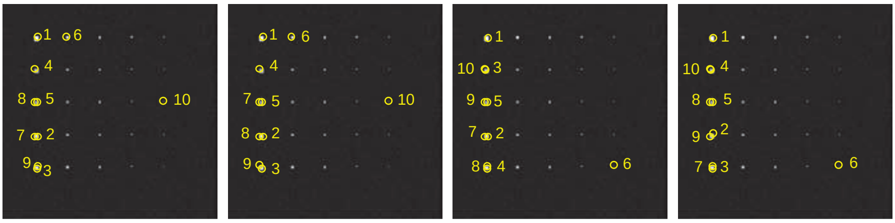
Figure 10.23 Results of IN-FINDR with two different runs.



t =0.005 t =0.05 t =0.005 t =0.05

(a) Original data (b) Sphered data

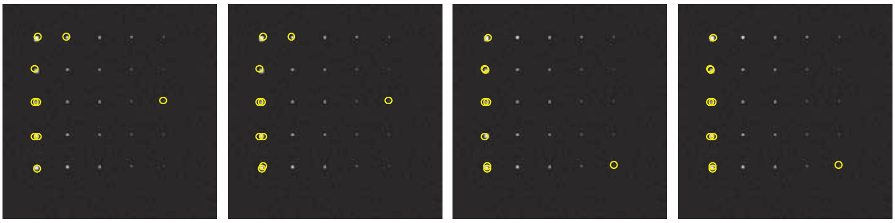
Figure 10.24 Results of RIN-FINDR with two different thresholds.



one run another run one run another run

(a) Original data (b) Sphered data

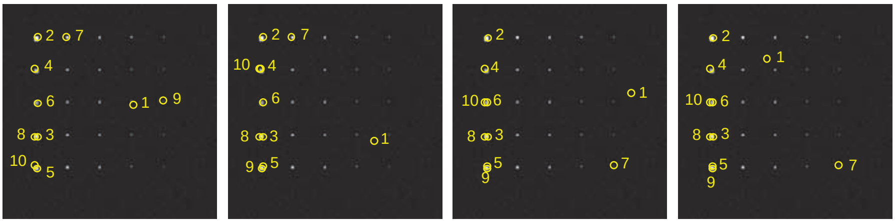
Figure 10.25 Results of SC N-FINDR with two different runs.



τ = 0.005 τ = 0.05 τ = 0.005 τ = 0.05

(a) Original data (b) Sphered data

Figure 10.26 Results of RSC N-FINDR with two different thresholds.



one run another run one run another run

(a) Original data (b) Sphered data

Figure 10.27 Results of SGA with two different runs.

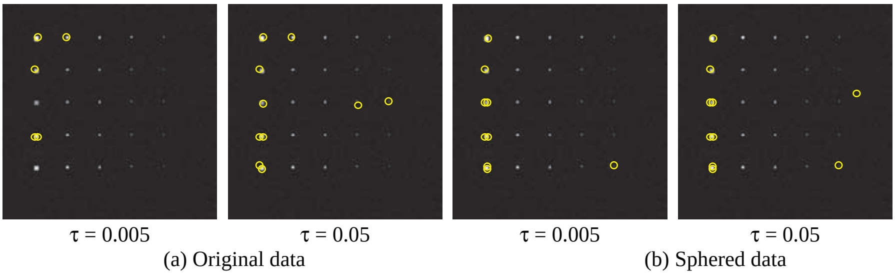


Figure 10.28 Results of RSGA with two different thresholds.

Table 10.5 Computing time and number of iterations for RIN-FINDR, RSC N-FINDR, and RSGA

		τ	Computing time (s)	Number of iteration
RIN-FINDR	Original	0.005	918.26	3
		0.05	669.42	2
	Sphered data	0.005	675.30	2
		0.05	675.95	2
RSC N-FINDR	Original	0.005	254.20	3
		0.05	1610.34	2
	Sphered data	0.005	258.72	3
		0.05	172.14	2
RSGA	Original	0.005	199.74	4
		0.05	110.35	2
	Sphered data	0.005	151.20	3
		0.05	101.54	2

was expected that a larger threshold value, $\tau = 0.05$, was also required for random versions of N-FINDR to combat the added Gaussian noise so that panel pixels with the purest signatures that represented the five distinct mineral signatures could be extracted as shown in Figures 10.24, 10.26, and 10.28.

Table 10.5 tabulates the computing time and the number of iterations required for a random algorithm to complete its search where once again RSGA required the least amount of computing time.

10.8 Real Image Experiments

The synthetic image experiments conducted in Section 10.6 demonstrate the effectiveness of REEAs in extracting endmembers. This section further provides evidence of the ability of REEAs in endmember extraction. Due to the availability of the ground truth only the HYDICE 15-panel scene in Figure 1.15(a) and cuprite data in Figure 1.12 were used for experiments.

10.8.1 HYDICE Image Experiments

With the ground truth given in Figure 1.15(b) there are 19 panel pixels marked by red that are assumed to be pure pixels specified by five panel signatures in Figure 1.16. Using this information, a quantitative study of extracted endmembers by various algorithms was performed for analysis. As usual, VD used for the HYDICE data was 9. So, the data dimensionality reduction performed

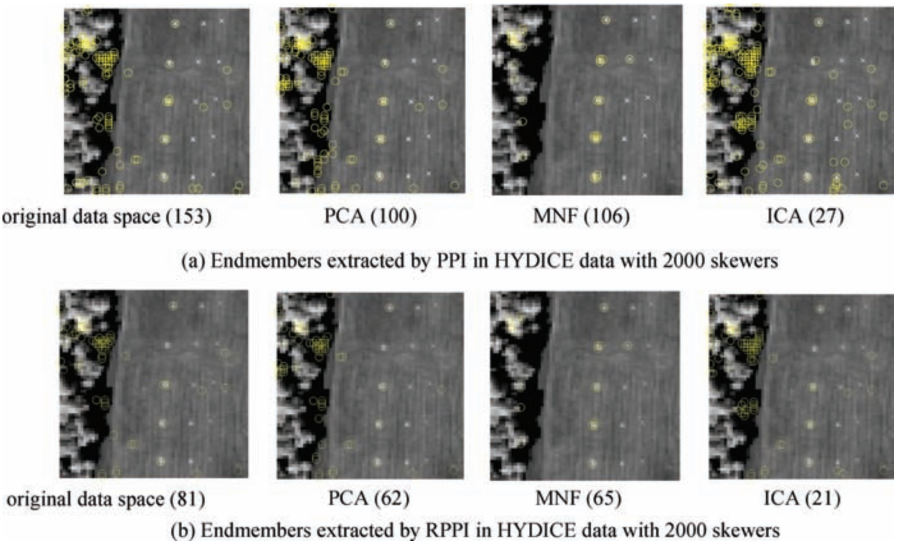

original data space (153)
PCA (100)
MNF (106)
ICA (27)
(a) Endmembers extracted by PPI in HYDICE data with 2000 skewers
original data space (81)
PCA (62)
MNF (65)
ICA (21)
(b) Endmembers extracted by RPPI in HYDICE data with 2000 skewers

Figure 10.29 Endmembers extracted by PPI and RPPI in HYDICE data with 2000 skewers.

for PPI was reduced to 9, and the number of endmembers required for RPPI and random versions of N-FINDR was set to twice VD, $2n_{\mathrm{VD}} = 18$.

10.8.1.1 RPPI

Figure 10.29(a) and (b) shows the results of operating PPI and RPPI using 2000 skewers on the original data and reduced nine-dimensional data sets where data samples marked by circles are those with their PPI counts greater than 0 in Figure 10.30(a) with the total number tallied in the parentheses and the final intersection produced by the RPPI in Figure 10.30(b) with the total number tallied in the parentheses. As noted, the total number of *K* skewers, 2000, was selected empirically. It is interesting to note that compared to PPI, which could find all the five panel signatures, only the four panel signatures $\mathbf{p}_1$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ were found by RPPI except the one using the ICA to perform DR. This may be due to the fact that the panel signature $\mathbf{p}_2$ is not really pure at all as shown in Chang et al. (2004, Tables III and IV). In this case, $\mathbf{p}_2$ did not appear in every realization produced by RPPI as other four panel signatures did. According to the results obtained in Chang et al. (2004), only the four panel signatures $\mathbf{p}_1$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ could be considered as endmembers in terms of the purest signatures in which case RPPI was the best and worked exactly as it is designed for. It has also been shown in Wang and Chang (2006a) that when ICA was used for DR, all the five panel signatures could be extracted because ICA can preserve small targets characterized by high-order statistics such as panel pixels.

10.8.1.2 RN-FINDR

Once again, using $2n_{\mathrm{VD}} = 18$ for RIN-FINDR, RSC N-FINDR, and RSGA to generate 18 endmembers, Figures 10.31–10.36 show the results of IN-FINDR, SC N-FINDR, and SGA along with their random counterparts where 18 endmembers were extracted by IN-FINDR in two runs using two different sets of random initial conditions. Unlike the experiments conducted for the TI and TE scenarios where the results are the same by operating IN-FINDR on the original data and sphered

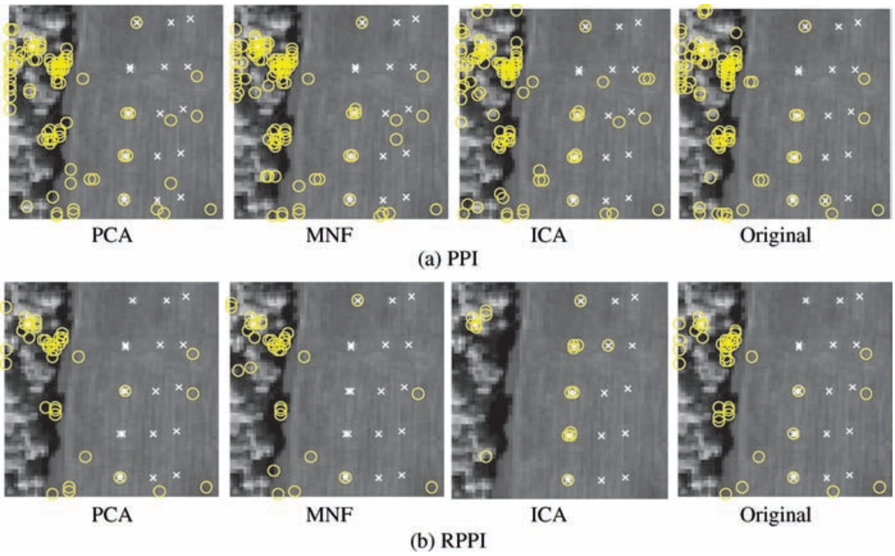
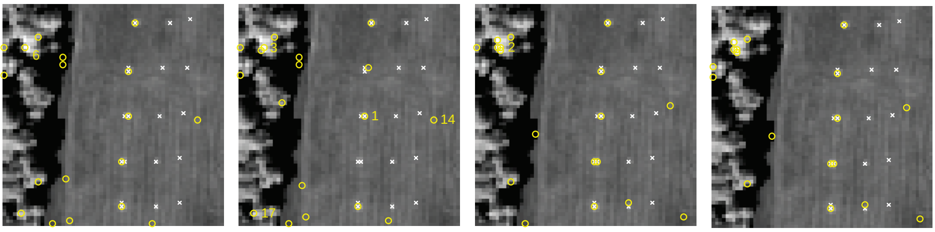


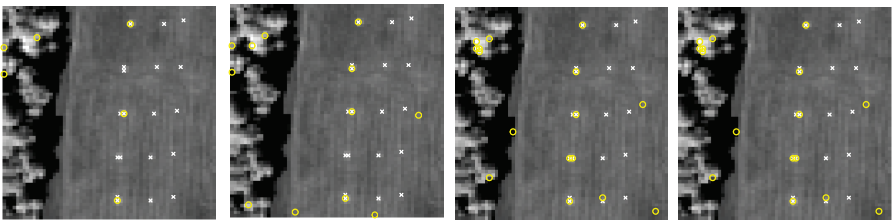
Figure 10.30 Endmembers extracted by PPI and RPPI in HYDICE data.



one run another run one run another run

(a) Original data (b) Sphered data

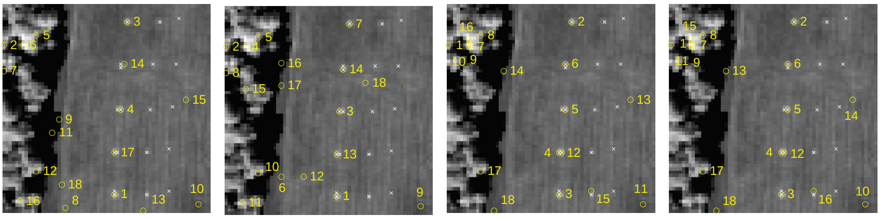
Figure 10.31 Results of IN-FINDR with two different runs.



$\tau = 0.005$ $\tau = 0.05$ $\tau = 0.005$ $\tau = 0.05$

(a) Original data (b) Sphered data

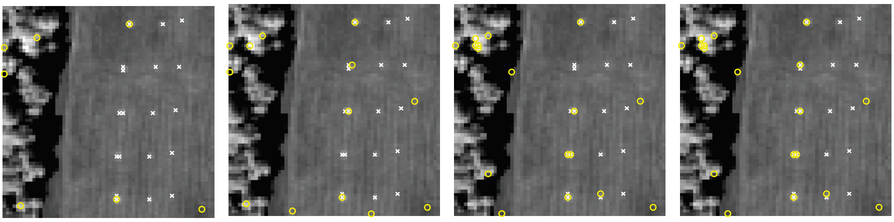
Figure 10.32 Results of RIN-FINDR with two different thresholds.



one run another run one run another run

(a) Original data (b) Sphered data

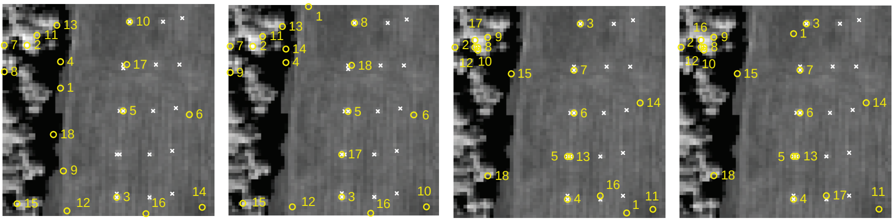
Figure 10.33 Results of SC N-FINDR with two different runs.



τ = 0.005 τ = 0.05 τ = 0.005 τ = 0.05

(a) Original data (b) Sphered data

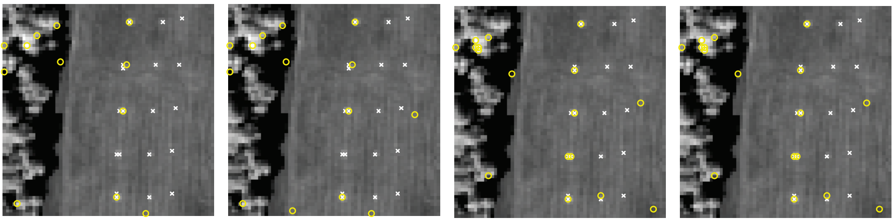
Figure 10.34 Results of RSC N-FINDR with two different thresholds.



one run another run one run another run

(a) Original data (b) Sphered data

Figure 10.35 Results of SGA with two different runs.



τ = 0.005 τ = 0.05 τ = 0.005 τ = 0.05

(a) Original data (b) Sphered data

Figure 10.36 Results of RSGA with two different thresholds.

data, there were 6 panel pixels in the first column extracted by IN-FINDR from the sphered data in Figure 10.31(b) compared to five and four panel pixels in the first column extracted from the original data in two separate runs in Figure 10.31(a). Figure 10.32(a) and (b) also shows similar results produced by RIN-FINDR using two different thresholds, $\tau = 0.005$ and 0.05, where the notation *x*/*y* indicates that a total of *y* endmembers were extracted by RIN-FINDR, among which *x* pixels were panel pixels corresponding to pure mineral signatures. Interestingly, RIN-FINDR performed exactly the same as IN-FINDR in terms of extracting six panel pixels in the first column when both algorithms operated on the sphered data in Figure 10.31(b), but was worse than that obtained by operating IN-FINDR on the original data. Similar results were also observed in Figures 10.33 and 10.34 produced by operating SC N-FINDR and RSC N-FINDR on the original data and sphered data, respectively. In the latter case, RSC N-FINDR was able to produce the same results as RIN-FINDR did when the threshold τ was set to 0.05. As shown in Figure 10.34 (b), RSC N-FINDR missed one panel pixel in row 2 if τ was set too small, 0.005. This made sense since the panel signature $\mathbf{p}_2$ is not really pure at all as shown in Chang et al. (2006). In this case, $\mathbf{p}_2$ was considered as a mixed signature and a variation of $\mathbf{p}_3$. So, $\mathbf{p}_2$ did not appear in every realization produced by RSC N-FINDR as other four panel signatures did. According to the results obtained in Chang et al. (2006), only the four panel signatures $\mathbf{p}_1$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ could be considered as endmembers in terms of the purest signatures in which case RSC N-FINDR was the best and worked exactly as it is designed for. However, there is an interesting finding in Figure 10.33(a) where SC N-FINDR was able to extract all the five panel pixels in the first column in both runs, while IN-FINDR missed one panel pixel in row 4 in one run in Figure 10.31(a). This evidence shows an important impact of the use of initial conditions on the final results. Similarly, all the conclusions drawn for SC N-FINDR and RSC N-FINDR were also applied to SGA and RSGA as shown in Figures 10.35 and 10.36. Finally, according to Figures 10.31–10.36, all the considered algorithms were able to extract panel pixels that represented all the five panel signatures if the data to be processed was sphered data.

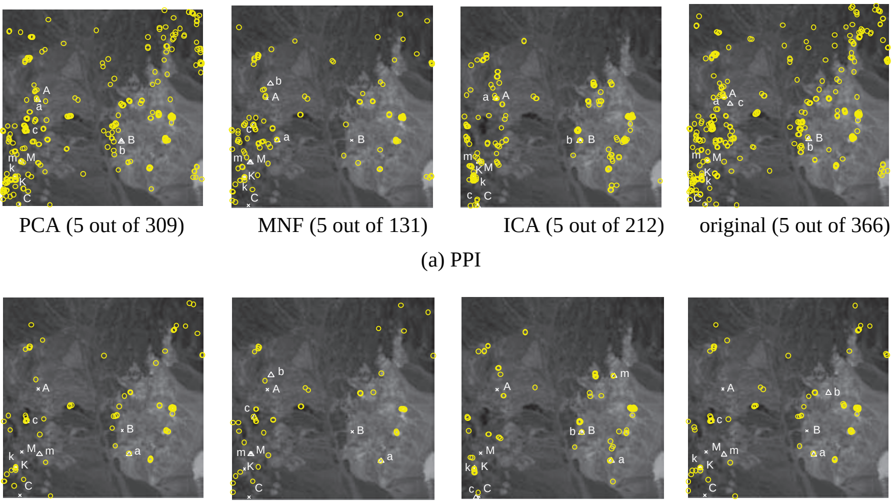
It should be noted that all the extracted pixels were measured by SAM between the extracted pixels and the 19 panel pixels in Figure 1.15(b) to see if the five endmembers corresponding to five panel signatures were successfully extracted. They were not performed by visual inspection. The experiments in Figures 10.31–10.36 show that all the extracted endmembers were identical to their corresponding (same) panel pixels specified by the ground truth in Figure 1.15(b) in which case their SAM values were zero. Since all other pixels extracted by RIN-FINDR and RSC N-FINDR were background pixels, they were of no interest to be included from a viewpoint of endmember extraction.

10.8.2 AVIRIS Image Experiments

The Cuprite data used for experiments in this section was a real image scene with reflectance values. VD estimated for this scene was $n_{\mathrm{VD}} = 22$. So, the data dimensionality reduction performed for PPI was reduced to 22, and the number of endmembers required for random versions of N-FINDR was set to twice VD, which is $2n_{\mathrm{VD}} = 44$.

10.8.2.1 RPPI

Figure 10.37(a-b) shows the data sample vectors extracted by operating PPI and RPPI on three DR-reduced image cubes by PCA, MNF, and ICA, and the original data space where the data samples marked by circles are those with PPI counts greater than 0 and PPI-extracted pixels considered to be endmembers are labeled by lowercase letters compared to the ground truth samples labeled by uppercase letters. The numbers in the parentheses underneath each of figures indicate the number



PCA (5 out of 309) MNF (5 out of 131) ICA (5 out of 212) original (5 out of 366)

(a) PPI

PCA (4 out of 91) MNF (4 out of 56) ICA (5 out of 99) original (5 out of 95)

(b) RPPI

Figure 10.37 Endmembers extracted RPPI and PPI in cuprite data with 2000 skewers.

Table 10.6 Computing time and the number of iterations of RPPI running in different data sets

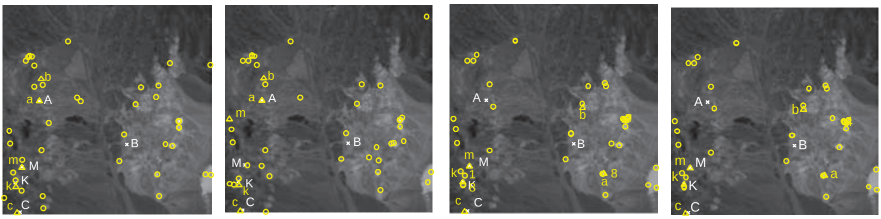
		Computing time (s)	Number of iterations
HYDICE data	PCA	1.61	9
	MNF	1.92	12
	ICA	0.48	3
	Original	1.66	6
Cuprite reflectance data	PCA	41.68	8
	MNF	25.66	5
	ICA	41.73	8
	Original	104.44	12

of endmembers extracted out of data sample vectors with their PPI counts greater than 0 in Figure 10.37(a) and the number of endmembers extracted out of data sample vectors found in the final intersection by RPPI in Figure 10.37(b). Comparing Figure 10.37(b) and 10.37(a), we found that RPPI produced much fewer data sample vectors than PPI did, while also performing as well as PPI in the original data space and its ICA-reduced image data cube by finding pixels that specified all the five mineral signatures for endmember extraction.

Table 10.6 tabulates the computing time and the number of iterations required by RPPI to run HYDICE and Cuprite image data scenes where the computing time required by PPI can be found by dividing the time required by RPPI by the number of iterations.

10.8.2.2 RN-FINDR

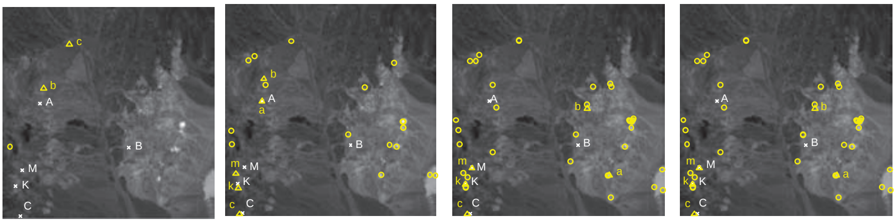
Figures 10.38–10.43 show data sample vectors extracted by IN-FINDR and SC N-FINDR and SGA along with their random counterparts where the number of endmembers required to be generated was $2p = 44$ by setting $p = 22$. In Figures 10.39, 10.41, and 10.43, the numbers x/y in the parentheses underneath the figures indicate that a total of y endmembers were extracted



one run (5) another run (5) one run (5) another run (5)

(a) Original data (b) Sphered data

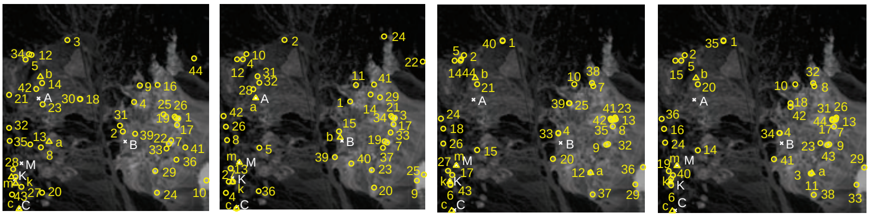
Figure 10.38 Results of IN-FINDR with two different runs.



τ = 0.005 (2) τ = 0.05 (5) τ = 0.005 (5) τ = 0.05 (5)

(a) Original data (b) Sphered data

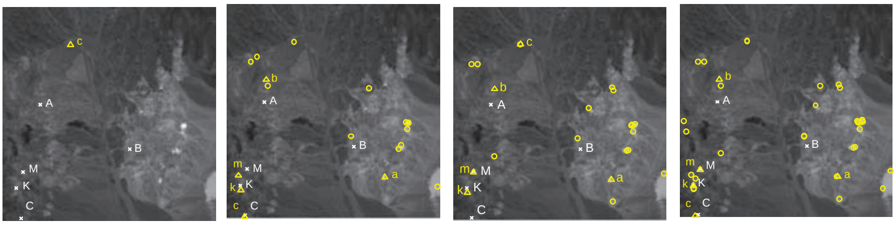
Figure 10.39 Results of RIN-FINDR with two different thresholds.



one run (5) another run (5) one run (5) another run (5)

(a) Original data (b) Sphered data

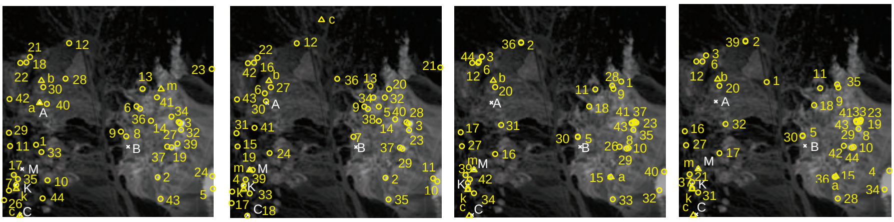
Figure 10.40 Results of SC N-FINDR with two different runs.



τ = 0.005 (1) τ = 0.05 (5) τ = 0.005 (5) τ = 0.05 (5)

(a) Original data (b) Sphered data

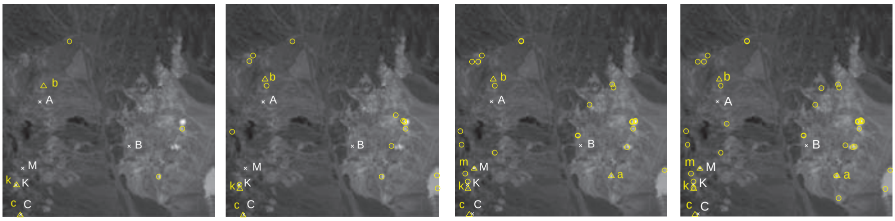
Figure 10.41 Results of RSC N-FINDR with two different thresholds.



one run (5) another run (3/a) one run (5) another run (5)

(a) Original data (b) Sphered data

Figure 10.42 Results of SGA with two different runs.



τ = 0.005 (3) τ = 0.05 (3) τ = 0.005 (5) τ = 0.05 (5)

(a) Original data (b) Sphered data

Figure 10.43 Results of RSGA with two different thresholds.

by an algorithm among which x pixels were endmembers corresponding to pure mineral signatures. As we can see from the figures, both IN-FINDR and SC N-FINDR were able to extract the pixels corresponding to the five distinct pure mineral signatures in different runs in Figures 10.38 and 10.40, while SGA missed the alunite signature in Figure 10.42 in one of two runs where the uppercase letters A, B, C, K, and M and the lowercase letters a, b, c, k, and m are used to indicate the ground truth pixels and pixels extracted by the algorithms, respectively. However, if the data to be used was sphered data, all the three algorithms could do equally well by finding all the five signature endmembers.

It is worth noting from Figures 10.38–10.43 that the extracted pixels corresponding to the ground truth pixels were not necessarily the same and their similarity was measured by SAM and MSE. Since only five mineral signatures were of major interest and represented five endmembers in the image scene, only those pixels extracted in Figures 10.38–10.43 were identified to be the closest to these five endmembers in terms of SAM and MSE. As a matter of fact, the spatial locations of the pixels corresponding to the same mineral signature provided by a ground truth pixel may be far away from each other. This is particularly true for those pixels extracted from the original data and sphered data.

Tables 10.7 and 10.8 tabulate the computing time and the number of iterations required by RIN-FINDR, RSC N-FINDR, and RSGA to run HYDICE and Cuprite image data scenes, respectively, where the computing time required by PPI was calculated by dividing the time required by RPPI by the number of iterations.

Table 10.7 Computing time and number of iterations required for HYDICE data by RIN-FINDR, RSC N-FINDR, and RSGA

		τ	Computing time (s)	Number of iterations
RIN-FINDR	Original data	0.005	486.44	5
		0.05	476.43	5
	Sphered data	0.005	446.83	3
		0.05	446.92	3
RSC N-FINDR	Original data	0.005	88.95	4
		0.05	92.29	4
	Sphered data	0.005	71.66	3
		0.05	72.69	3
RSGA	Original data	0.005	52.59	4
		0.05	54.12	4
	Sphered data	0.005	40.28	3
		0.05	28.60	2

Table 10.8 Computing time and number of iterations required for cuprite data by RN-FINDR, RSC N-FINDR, and RSGA

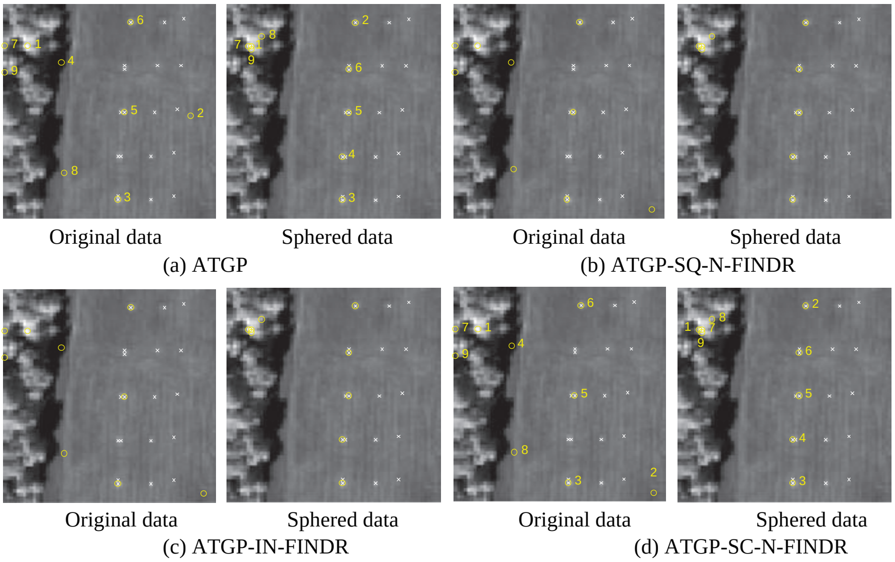
		τ	Computing time (s)	Number of iterations
RN-FINDR	Original data	0.005	4.6063×10^5	8
		0.05	1.7548×10^6	8
	Sphered data	0.005	1.7380×10^5	3
		0.05	1.7379×10^5	2
RSC N-FINDR	Original data	0.005	9.4422×10^4	9
		0.05	5.6226×10^4	6
	Sphered data	0.005	8.7037×10^4	8
		0.05	4.8172×10^4	5
RSGA	Original data	0.005	2.6691×10^4	7
		0.05	3.0511×10^4	8
	Sphered data	0.005	2.3454×10^4	6
		0.05	1.5644×10^4	4

10.9 Conclusions

An EEA generally requires a set of initial endmembers for algorithm initialization. In doing so a random generator is usually used to produce a set of random initial endmembers for this purpose. Unfortunately, this practice affects the reproducibility of final results. Two approaches are explored to address this issue. One is considered in Chapter 9 where various custom-designed initialization algorithms are developed to produce specific sets of initial conditions that can lead to good results. The other adopts a completely opposite approach discussed in this chapter, by taking advantage of random initial conditions to convert an EEA into a random EEA. In order to materialize this idea, two most popular and widely used EEAs, PPI and N-FINDR, are extended as random PPI (RPPI) and random N-FINDR (RN-FINDR) and the the same treatment can be also applied to other EEAs to derive their random counterparts.

PPI has become a standard technique for endmember extraction due to its popularity and availability in the ENVI software. This chapter investigates several practical issues in implementing PPI. Among them two major issues are particularly severe. One is in-reproducibility caused by the use of randomly generated vectors, called skewers where different sets of the same number of skewers produce different results. Another is the requirement of a visualization tool for users to manually manipulate the final selection of endmembers. As a result, a novice and an experienced user will produce different results. To resolve these issues, this chapter develops an RPPI that implements PPI as a random algorithm in the sense that PPI using one set of skewers is considered as a single run of RPPI and its result is considered as a realization. With this interpretation, the disadvantage of using random initial endmembers becomes an advantage that actually resolves two major long standing issues: determination of the number of endmembers, p, required for PPI to generate and inconsistent selection of final endmembers. The study shows that RPPI can be as competitive as PPI.

In addition to PPI, the N-finder algorithm (N-FINDR) is also another algorithm that has been widely used for endmember extraction. In analogy with PPI, it also suffers from a number of issues that have prevented it from being used in practical implementation; the use of random initial endmembers for algorithm initialization is the only issue both PPI and N-FINDR share. As a result, an approach that is used to derive RPPI can also be applied to N-FINDR to develop a random version of N-FINDR. However, N-FINDR is a very computationally expensive algorithm, discussed in Chapter 7, and implementing such an N-FINDR repeatedly as a RN-FINDR is formidable. It must be redesigned to mitigate this dilemma. To accomplish this goal, two sequential versions of the N-FINDR, iterative N-FINDR (IN-FINDR) and SuCcessive N-FINDR (SC N-FINDR) developed in Chapter 8, can be used to replace N-FINDR for this purpose. Then the use of random initial



Original data Sphered data
(a) ATGP

Original data Sphered data
(b) ATGP-SQ-N-FINDR

Original data Sphered data
(c) ATGP-IN-FINDR

Original data Sphered data
(d) ATGP-SC-N-FINDR

Figure 10.44 Endmember extraction results of (a) ATGP, (b) ATGP-SQ-N-FINDR, (c) ATGP-IN-FINDR, and (d) ATGP-SC-N-FINDR.

conditions by N-FINDR can be resolved by further extending IN-FINDR and SC N-FINDR to random IN-FINDR (RIN-FINDR) and random SC N-FINDR (RSC N-FINDR) in such a way that an algorithm using one random set of initial endmembers is considered as a single run of the algorithm and the result obtained by a single run constitutes one realization. Accordingly, the RIN-FINDR and RSC N-FINDR convert the random issue to an advantage that eventually resolves the problem of inconsistency in final results. Besides, to further improve the computational complexity and performance of RN-FINDR, the original data are sphered to enhance the extractability of endmembers.

Since both ID-EEAs and REEAs are designed by two completely opposite rationales to cope with the inconsistency issue of random initial endmembers, it is interesting to compare their relative performance in terms of endmember extraction. In doing so, the HYDICE data are chosen for experiments and the automatic target generation process (ATGP) is used as the endmember initialization algorithm (EIA) to generate an appropriate set of initial endmembers for N-FINDR because ATGP has been shown to be the best among EIA-EEAs in Chapter 9. Figure 10.44 shows the endmember extraction results of ATGP, ATGP-SQ-N-FINDR, ATGP-IN-FINDR, and ATGP-SC-N-FINDR where all the EIA-EEAs could extract all the five panel signatures if the sphered data are used, but only three panel signatures if the original data are used.

Comparing the results in Figure 10.44 to those in Figures 10.32, 10.34, and 10.36, we found that RIN-FINDR performed better than ATGP-N-FINDR on original data and worked as effectively as the ATGP-N-FINDR if the sphered data was used. Nevertheless, this benefit was traded for high computational complexity. So, the preference of one approach over another is determined by the users.

11

Exploration on Relationships among Endmember Extraction Algorithms

Chapters 7–8 provide a series of studies on development and design of endmember extraction algorithms (EEAs) from an algorithmic implementation's point of view where Chapter 7 presents various simultaneous EEAs (SM-EEAs) that find all the desired endmembers simultaneously as opposed to sequential EEAs (SQ-EEAs) in Chapter 8 that find all the desired endmembers one after another in sequence. Due to the use of random initial endmembers SM-EEAs and SQ-EEAs produce final extracted endmembers that are inconsistent and not repeatable. To address this issue, two completely opposite approaches are investigated: initialization-driven EEAs (ID-EEAs) in Chapter 9 with initial endmembers generated by a custom-designed algorithm and random EEAs (REEAs) in Chapter 10 with random initial endmembers considered as realizations of a random algorithm. Despite that each of these four types of EEAs, that is, SM-EEAs, SQ-EEAs, ID-EEAs, and REEAs, is treated in individual chapters it is very interesting to investigate and explore their correlations and relationships. In particular, the two most widely used EEAs, PPI and N-FINDR, can be actually implemented in various versions derived from these four types of EEAs. This chapter is included for this purpose to further explore insights into the EEAs derived in Chapters 7–10.

11.1 Introduction

A wide variety of algorithms have been developed for endmember extraction in recent years. One of early developed EEAs is the so-called pixel purity index (PPI) proposed by Boardman (1994), which has become one of most popular EEAs due to its availability in the software, environment for visualizing images (ENVI). The design rationale of PPI is based on the concept of convex geometry, especially on the fact that a line segment connected by two points includes all the points mixed by the two end points as a result of convexity in which case two end points are considered as pure points, whereas the points right in between can be obtained by mixing these end points with appropriate portions. In order to make it work, PPI randomly generates a set of unit vectors to be considered as line segments, called skewers, and then use these skewers as basis vectors onto which all data sample vectors are projected. The number of times a data sample vector orthogonally projected at end points of these skewers is defined as its PPI count that will be used to determine if this data sample vector is an endmember (see Figure 7.1). Theoretically speaking, higher the PPI count of a data sample vector, more likely the data sample vector to be an endmember. As

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

already demonstrated in the experiments of six scenarios conducted in Chapter 4, this is generally not the case. Nevertheless, it is usually true that the PPI count of an endmember is always nonzero, that is, greater than 0, but could be also a very small value such as one. There are several issues arising in implementing PPI. One is the number of skewers to be used must be sufficiently large to cover all possible random directions that an endmember can be orthogonally projected. (There is no guideline provided regarding how large this number should be. It must be done on a trial-and-error basis.) As a trade-off, many data sample vectors are also falsely extracted as endmembers. Another drawback resulting from the use of PPI counts is that all data sample vectors with PPI counts greater than a threshold will be extracted as endmembers in which case a large number of data sample vectors corresponding to the same type of endmembers are also extracted. Such a large number of extracted endmembers usually cause confusion to image analysts. Besides, PPI also requires human intervention to adjust a value to appropriately threshold PPI counts. As a result, different users may produce different results. Despite that in Nascimento and Dias (2005) the authors did not specify their developed EEA, called vertex component analysis (VCA), as a variant of PPI, the same concept of orthogonal projection (OP) on skewers used by PPI is also used by VCA except that VCA implemented Gaussian random variables to produce skewers as initial conditions and then generated a successive set of projection vectors in sequence onto which data sample vectors can be orthogonally projected. So, from an algorithmic implementation view point VCA can be considered as a sequential version of PPI because it does not simultaneously use all projection vectors to find all endmembers as the way PPI does by thresholding PPI counts of all data sample vectors in one shot operation. Instead, VCA generates an orthogonal projection vector to find one endmember at a time in sequence. Since VCA inherits the concept of skewers to perform OP, it also suffers from the same drawback that requires a large number of random projection vectors to be able to cover all possible directions resulting from OP. In a rather different application, the concept of OP was also used by an algorithm developed by Ren and Chang (20003) called automatic target generation process (ATGP), to find targets of interest via a series of successive orthogonal projections. As already shown in Chapter 8, when ATGP is used for endmember extraction as ATGP-EEA, many of such ATGP-generated targets turned out to be endmembers. So, it should not be surprising to see that PPI, VCA, and ATGP actually belong to the same family.

As an alternative to the use of OP as a criterion to extract endmembers, another popular EEA, called N-FINDER algorithm (N-FINDR) developed by Winter (1999), uses the concept of simplex volume. Unlike PPI, which uses a large set of skewers to generate a PPI count for each of data sample vectors via OP to find endmembers, N-FINDR extends line segments to simplexes with a line segment being regarded as a two-dimensional (2D) degenerated simplex. In order to realize its applicability to endmember extraction, N-FINDR borrows the concept from Craig's minimal volume transform (MVT) (Craig, 1994) that used the minimal simplex volume as a criterion to find endmembers. Rather searching for a simplex that embraces all data sample vectors with the minimal simplex volume as MVT does, N-FINDR finds a simplex that is embedded within data space with the maximal simplex volume. More specifically, for a given positive integer p designated as the number of endmembers a p-vertex simplex that yields the maximal volume among all possible p-vertex simplexes embedded in a data space should have all of its vertices specified by endmembers. Using PPI as an example, the two points connecting a line segment with maximal length are more likely to be endmembers since this pair of end points produce the maximal volume among all 2D degenerated simplexes (line segments) embedded in a data space.

Since OP and simplex volume are two basic principles of convex geometry that can be used to find endmembers, they have been widely used to design and develop most EEAs reported in the literature. However, there are several drawbacks when PPI and N-FINDR are implemented as SM-EEAs. One is very high computational cost resulting from an exhaustive search for all endmembers simultaneously as noted in Chapter 7. Another is inconsistency in final results due to the use of random initial conditions. A third one is that the number of endmembers, *p*, which is practically unknown, must assume to be known *a priori* before data processing takes place. Chapter 8 address the first issue by making an SM-EEA a sequential EEA (SQ-EEA) that can be implemented to extract one endmember at a time in a sequential manner to reduce computational complexity at the expense of optimality. In particular, VCA and ATGP-EEA developed in Chapter 8 can be considered as sequential versions of PPI, while the sequential N-FINDR (SQ N-FINDR) (Wu et al., 2008) and simplex growing algorithm (SGA) (Chang et al., 2006) developed in Chapter 8 can be viewed as sequential versions of N-FINDR. Chapter 9 address the second issue by developing initialization-driven EEAs (ID-EEAs) to produce consistent results in endmember extraction where an initial condition is produced by a custom-designed algorithm either selecting specific data sample vectors or using endmember initialization algorithm (EIA) to generate appropriate sets of *p* initial endmembers. Finally in order to address the third issue, the concept of virtual dimensionality (VD) discussed in Chapter 5 is used to provide an estimate on the number of endmembers for SM-EEAs in Chapter 7, SQ-EEAs in Chapter 8, and ID-EEAs in Chapter 9, all of which require this knowledge *a priori*. By taking an opposite approach, Chapter 10 considers an EEA using random initial endmembers as a random EEA (REEA) where a single run is defined as results of running an REEA using one set of random initial endmembers. With this interpretation, a single run of an REEA is simply viewed as a realization of an REEA. If there is a true endmember present in the data, theoretically speaking, it should appear in realizations. With this assumption an REEA can find desired endmembers by taking intersection of realizations resulting from a number of runs. In this case, the total number of endmembers will be automatically determined by the intersection set and there is no need of VD to determine the number of endmembers *a priori* as required by SM-EEAs, SQ-EEAs, and ID-EEAs. While four types of EEAs, SM-EEAs, SQ-EEAs, ID-EEAs, and REEAs, are treated in separate and individual chapters, some interesting relationships connecting one to another were not be able to be discussed in these chapters. This chapter explores insights into ideas used to design EEAs, especially OP used by PPI and simplex volume used by N-FINDR and their algorithm development.

11.2 Orthogonal Projection-Based EEAs

In this section, we further show that PPI, VCA, and ATGP-EEA are essentially the same type of algorithms that rather appear in different forms. All of these three EEAs use the OP derived from the principle of orthogonality to find endmembers. OP is one of most widely used concepts in statistical signal processing and plays a key role in mean squared error or least squares error-based approaches (Poor, 1994). It basically says that any new or innovations information must be orthogonal in the sense of mean squared error or least squares error to the information that is already known or the information that can be inferred from the data samples that were already processed. A best example of using OP is the orthogonal subspace projection (OSP) approach developed for linear spectral mixture analysis (Harsanyi and Chang, 1994; Chang, 2003a; Chang, 2005) as well as ATGP used for target detection and classification (Ren and Chang, 2003; Chang, 2003a). Interestingly, the first use of OP in endmember extraction is PPI, which assumes that the more likely the data sample to be an endmember, the better chance the data sample to be orthogonally projected at end points of a skewer where a skewer is a randomly generated unit vector. A further use of OP is exploited by VCA that selects a data sample vector with the maximal OP in the OSP-projected space as a potential endmember. Interestingly, PPI, ATGP, and VCA share the same idea of using OP to find endmembers, but their relationships are never realized and explored until Wu et al. (2007). As a matter of fact, these three seemingly different EEAs are indeed closely related. The following section reinterprets them from an OP perspective in a unified framework.

11.2.1 Relationship among PPI, VCA, and ATGP

The idea of PPI is very simple and intuitive. It assumes that for a given "*right*" vector an endmember should have its OP falling at one of two end points of the vector with either maximal or minimal OP. A key issue is how to find such a "*right*" vector. Since there is no prior knowledge about endmembers, the best way is to randomly generate as many vectors as needed to find "*right*" vectors so that endmembers can be orthogonally projected at as many their end points as possible. A vector randomly generated with unit length for this purpose is called a skewer. In order to accomplish its goal, PPI requires a large number of skewers, K, and then assigns a count to each of data sample vectors, to be called the PPI count, by counting the number of skewers on which the data sample vector is projected orthogonally as their end points. Hopefully, the higher PPI count of a data sample vector, the more likely it is an endmember. Despite that such an approach is elegant, two major drawbacks need to be addressed. First is how large the number of skewers is sufficient. Second is how to determine an appropriate cut-off value for PPI counts to produce a desired set of endmembers. Unfortunately, none of these two issues is easy to deal with.

The development of ATGP is originally designed to find targets of interest in data when no prior target knowledge is available. It repeatedly implements orthogonal projections to produce a nested sequence of decreasing orthogonal complement subspaces, each of which can be used to extract a specific target with maximal orthogonal projection. As shown in Chapter 8, most of ATGP-generated target pixels indeed ended up desired endmembers. This is certainly not a surprise because the concept behind ATGP is OP that is exactly the same criterion used by PPI. Nevertheless, there are two exceptions. First is that PPI requires a very large number of skewers to find maximal or minimal OP compared to ATGP that finds targets of interest from a sequence of orthogonal projection subspaces with maximal projections. Accordingly, PPI simultaneously extracts all endmembers compared to ATGP that extracts targets sequentially one after another. Second is that PPI takes advantage of random nature in skewers to uncover all possible endmembers as opposed to ATGP that takes a deterministic approach to search for definite directions of maximal projections to find possible target candidates in a sequence of orthogonal projection subspaces. Nonetheless, both PPI and ATGP share the same principle and implement OP to find what they are designed for. The goal of VCA is to improve PPI by replacing the skewers with Gaussian random vectors, which can be viewed as Gaussian skewers, and then repeatedly using OP as the way ATGP does to generate a sequence of subspaces via OP in which data with the maximal OP are extracted as an endmember. In addition to the use of Gaussian skewers to produce initial endmembers, VCA also differs from PPI in three aspects. VCA does not need a large number of Gaussian skewers, but it does require prior knowledge about the number of endmembers needed to be generated. Another is that VCA is a SQ-EEA and can be considered as a sequential version of PPI which is in fact an SM-EEA. ATGP actually fills the gap between PPI and VCA. Finally, in order to ensure that endmembers to be extracted are non-negative, i.e., satisfying abundance non-negativity constraint (ANC), VCA restricts its search region in the first quadrant, while PPI selects skewers from all possible random directions.

11.2.1.1 Relationship Between PPI and ATGP

As noted in the introduction, PPI and ATGP-EEA share the principle of orthogonality to find their targets of interest, that is, endmembers in our case. In what follows, we conduct an item-by-item in-depth comparative study between these two. For the sake of simplicity the ATGP-EEA is replaced with ATGP throughout the chapter.

1. *Dimensionality reduction (DR)*: PPI requires DR to reduce computational complexity, while ATGP does not. However, in fact, PPI does not necessarily require DR as shown by RPPI. Its inclusion of DR is simply for the purpose of computation relief.
2. *Initial process*: PPI requires a random generator as an initial process to produce a set of skewers, while ATGP selects a data sample with maximal length as an initial target to initialize the algorithm.
3. For each $\mathbf{skewer}_k$ PPI calculates $(\mathbf{skewner}_k)^T\mathbf{x}$ for all $\mathbf{x} \in \mathbf{X}$ and finds its maximal OP and minimal OP, denoted by max($\mathbf{skewner}_k$) and min($\mathbf{skewner}_k$), respectively. Then for each $\mathbf{x} \in \mathbf{X}$, PPI count of the $\mathbf{x}$ is calculated by counting the number of skewers such that either $(\mathbf{skewner}_k)^T\mathbf{x} = \max(\mathbf{skewner}_k)$ or $(\mathbf{skewner}_k)^T\mathbf{x} = \min(\mathbf{skewner}_k)$ is true. It should be noted that OP performed by PPI on a skewer can be along the same direction or opposite direction of the skewer depending on whether the OP is the maximal or minimal projection as illustrated in Figure 7.1. However, according to (8.6), which is always non-negative, ATGP only finds maximal projections every time when it searches a new endmember. Compared to the PPI, the ATGP finds $\mathbf{t}^{(k)}$ via (8.6) from a sequence of decreasing projection subspaces, $\langle \mathbf{U}^{(0)} \rangle^{\perp} \supset \langle \mathbf{U}^{(1)} \rangle^{\perp} \supset \cdots \supset \langle \mathbf{U}^{(k)} \rangle^{\perp}$.
4. PPI requires an appropriate value to simultaneously threshold PPI counts of all data sample vectors to produce a set of endmembers. Unfortunately, there is no provided criterion for such threshold selection. As a result, PPI generally extracts a large number of endmembers and many of them turn out to be not real endmembers. By contrast, ATGP produces targets of interest one at a time sequentially until the total number of extracted targets reaches the required number of endmembers. So, PPI can be considered as an SM-EEA as opposed to ATGP that is an SQ-EEA.

The above-mentioned relationship between PPI and ATGP was not explored in Chang and Plaza (2006) and Plaza and Chang (2006), but their conducted experiments did show that most of final PPI-selected endmembers are the same data sample vectors produced by ATGP.

11.2.1.2 Relationship Between PPI and VCA

The relationship between PPI and VCA is more obvious than that between PPI and ATGP described above.

1. *Dimensionality reduction*: Both require DR with the principal components analysis (PCA) or maximum noise fraction (MNF) for PPI and singular value decomposition (SVD) for VCA. Same comment made on item 1 in Section 11.2.1.1 is also applicable to this item. Both PPI and VCA do not really need DR. However, it should be noted that an appropriate DR can actually improve their performance such as using ICA to perform DR in Chapter 8.
2. *Initial process*: PPI uses a random generator to produce unit vectors as skewers, while VCA uses Gaussian random vectors as Gaussian skewers.
3. PPI generates all skewers at the same time, while VCA find the desired targets one at a time from a sequence of orthogonal projection subspaces with dimensionality reduced by one at a time, that is, $\langle \mathbf{A}^{(0)} \rangle^{\perp} \supset \langle \mathbf{A}^{(1)} \rangle^{\perp} \supset \cdots \supset \langle \mathbf{A}^{(k)} \rangle^{\perp}$.
4. A skewer $\mathbf{skewer}_k$ in PPI plays the same role as $\mathbf{f}^{(k)}$ does in VCA where PPI uses a random generator to produce skewers all together at the same time as opposed to VCA that makes use of Gaussian variables to initialize the algorithm every time it searches for a new endmember.
5. The OP calculated by $(\mathbf{skewer}_k)^T\mathbf{x}$ is identical to the OP performed by $\left(\mathbf{f}^{(k)}\right)^T\mathbf{x}$.

6. The only difference between PPI and VCA is that PPI performs $(\mathbf{skewer}_n)^T\mathbf{x}$ for all skewers and finds all desired endmembers simultaneously, while VCA performs $\left(\mathbf{f}^{(k)}\right)^T\mathbf{x}$ sequentially one after another, that is, $\mathbf{f}^{(0)}, \mathbf{f}^{(1)}, \ldots, \mathbf{f}^{(k)}$.
7. For each data sample vector $\mathbf{x}$, PPI first finds both

$$\begin{aligned} n_{\max}(\hat{\mathbf{x}}) &= \#\,\text{of skewers that yield the maximal projection for}\,\mathbf{x} \\ &= \#\left\{\mathbf{skewer}_k|(\mathbf{skewer}_k)^T\mathbf{x} \geq (\mathbf{skewer}_k)^T\mathbf{y}\;\text{for all}\;\mathbf{y} \in \mathbf{X}\right\} \end{aligned}$$

and

$$\begin{aligned} n_{\min}(\mathbf{x}) &= \#\,\text{of skewers that yield the minimal projection for}\,\mathbf{x} \\ &= \#\left\{\mathbf{skewer}_k|(\mathbf{skewer}_k)^T\mathbf{x} \leq (\mathbf{skewer}_k)^T\mathbf{y}\,\text{for all}\,\mathbf{y} \in \mathbf{X}\right\}, \end{aligned}$$

and calculate its PPI count of the data sample vector $\mathbf{x}$ by $N_{\text{PPI}}(\mathbf{x}) = n_{\max}(\mathbf{x}) + n_{\min}(\mathbf{x})$, then uses a prescribed value to threshold PPI counts of all data sample vectors to produce desired endmembers. By contrast, VCA finds $\mathbf{e}^{(k)}$ via (8.5) to find the one with maximal projection at a time.
8. PPI finds all endmembers simultaneously by thresholding PPI counts of each data sample vector, whereas VCA produces endmembers, one after another sequentially, $\mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \ldots, \mathbf{e}^{(k)}$.

11.2.1.3 Relationship Between ATGP and VCA

The relationship between ATGP and VCA was first noted in Wu et al. (2007) and later in Greg (2010). Technically speaking, both ATGP and VCA are essentially the same algorithm in terms of design rationale as they can be interpreted one from another as follows.

1. *Dimensionality reduction*: VCA requires SVD to perform DR that is not required by ATGP. In this case, the notations of $\mathbf{r}$ and $\mathbf{x}$ are used to indicate sample vectors in the original data space and the DR-reduced space, respectively, for distinction. As noted above, the DR is not crucial to VCA and can be skipped.
2. ATGP uses $\left(P^{\perp}_{\mathbf{U}^{(k-1)}}\mathbf{r}\right)^T\left(P^{\perp}_{\mathbf{U}^{(k-1)}}\mathbf{r}\right)$ to produce the maximal projection, while VCA uses

$$\left(\mathbf{f}^{(k)}\right)^T\hat{\mathbf{x}} = \left(P^{\perp}_{\mathbf{A}^{(k-1)}}\mathbf{w}^{(k)}\right)/\left(\left\|P^{\perp}_{\mathbf{A}^{(k-1)}}\mathbf{w}^{(k)}\right\|\right)\mathbf{x} \tag{11.1}$$

to produce the maximal projection with $P^{\perp}_{\mathbf{U}^{(k-1)}}$ in ATGP corresponding to $P^{\perp}_{\mathbf{A}^{(k-1)}}$ in VCA. The only difference between these two is that a randomly generated Gaussian vector $\mathbf{w}^{(k)}$ is used in $\mathbf{f}^{(k)}$ in conjunction with data sample vector $\mathbf{x}$ in VCA compared to the same data vector $\mathbf{r}$ used in $\left(P^{\perp}_{\mathbf{U}^{(k-1)}}\mathbf{r}\right)^T\left(P^{\perp}_{\mathbf{U}^{(k-1)}}\mathbf{r}\right)$ for ATGP. So, if we replace $\mathbf{w}^{(k)}$ in (11.1) with $\hat{\mathbf{x}}$, that is,

$$\left(\mathbf{f}^{(k)}\right)^T\hat{\mathbf{x}} = \left(P^{\perp}_{\mathbf{A}^{(k-1)}}\hat{\mathbf{x}}\right)/\left(\left\|P^{\perp}_{\mathbf{A}^{(k-1)}}\hat{\mathbf{x}}\right\|\right)\hat{\mathbf{x}} \tag{11.2}$$

then VCA becomes a variant of ATGP.
3. ATGP produces a sequence of target pixels of interest

$$\left\{\mathbf{t}^{(0)}\right\} \subset \left\{\mathbf{t}^{(0)}, \mathbf{t}^{(1)}\right\} \subset \cdots \subset \left\{\mathbf{t}^{(0)}, \mathbf{t}^{(1)}, \ \ldots, \mathbf{t}^{(k)}\right\} \tag{11.3}$$

from which a sequence of projection subspaces that are orthogonal to the target sequence in (11.3) can be also produced as follows:

$$\langle \mathbf{t}^{(0)} \rangle^{\perp} \supset \langle \mathbf{t}^{(0)}, \mathbf{t}^{(1)} \rangle^{\perp} \supset \cdots \supset \langle \mathbf{t}^{(0)}, \mathbf{t}^{(1)}, \ \ldots, \mathbf{t}^{(k)} \rangle^{\perp} \tag{11.4}$$

Analogous to ATGP, VCA also generates a sequence of endmembers

$$\left\{ \mathbf{e}^{(0)} \right\} \subset \left\{ \mathbf{e}^{(0)}, \mathbf{e}^{(1)} \right\} \subset \cdots \subset \left\{ \mathbf{e}^{(0)}, \mathbf{e}^{(1)}, \ \ldots, \mathbf{e}^{(k)} \right\} \tag{11.5}$$

from which a sequence of projection subspaces that are orthogonal to the endmember sequence in (11.5) can be also produced in the same fashion as that carried out by ATGP as follows:

$$\langle \mathbf{A}^{(0)} \rangle^{\perp} \supset \langle \mathbf{A}^{(1)} \rangle^{\perp} \supset \cdots \supset \langle \mathbf{A}^{(k)} \rangle^{\perp} \tag{11.6}$$

11.2.1.4 Discussions

VCA suffers from the uncertainty caused by the use of random initial endmembers, as PPI does. In order resolve this issue, VCA requires a large number of initial endmembers as PPI does to constitute reliable statistics. Therefore, the number of vertices required for VCA must be sufficiently large. As a consequence, it may result in more vertices than VCA needs. Such a dilemma is avoided by PPI that provides a visualization tool to allow users to manually perform what VCA cannot do. ATGP introduces a deterministic approach to correct this random issue and uses a VD-estimated value to generate a desired set of target sample vectors. Such an approach is adopted by PPI and referred to as fast iterative PPI (FIPPI) in Chang and Plaza (2006) and will be referred to as ATGP-PPI for consistency in this chapter. A similar approach can be also applied to VCA, called ATGP-VCA, to mitigate the same issue caused by the use of Gaussian random variables by VCA.

Figure 11.1 summarizes the relationships among PPI, ATGP, and VCA along with ATGP-PPI and ATGP-VCA, both of which implement ATGP as their initialization algorithm to produce an initial set of endmembers. As noted, VCA, ATGP-VCA, and ATGP-PPI play a role bridging PPI and ATGP.

The insights into relationships in Figure 11.1 have never been investigated in the literature. As a matter of fact, VCA is nothing more than an orthogonal projection-based endmember extraction algorithm that can be considered as a variant of PPI or ATGP. However, the connection between PPI and VCA is not clear until ATGP is used to bridge the gap between these two. Since both PPI and VCA use randomly generated vectors as their initial endmembers, two issues become

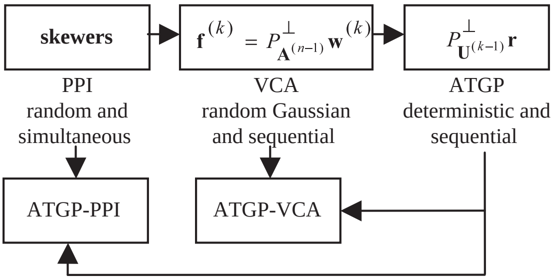

skewers
$\mathbf{f}^{(k)} = P^{\perp}_{\mathbf{A}^{(n-1)}} \mathbf{w}^{(k)}$
$P^{\perp}_{\mathbf{U}^{(k-1)}} \mathbf{r}$
PPI
random and
simultaneous
VCA
random Gaussian
and sequential
ATGP
deterministic and
sequential
ATGP-PPI
ATGP-VCA

Figure 11.1 Relationships among the PPI, VCA, and ATGP.

evidential: inconsistency in final results and insufficient statistics to cope with randomness. In order to cope with these two issues, ATGP-PPI and ATGP-VCA are introduced to produce a set of ATGP-generated target sample vectors that can be used as initial endmembers.

Over the past years, VCA has been a well-accepted endmember extraction algorithm in the literature. However, the insights into the design rationale into VCA have not been explored. In this section, we trace back to its origin and show that VCA is actually a variant of ATGP and is derived from the concept of PPI. Furthermore, as shown in Chang et al. (2006), Chapter 8 and the following experiments in Section 11.2.2 the performance of VCA is generally not as good as many people expect. Specifically, as an endmember extraction algorithm, VCA generally cannot compete against simplex-based methods such as N-FINDR, SGA. This is because its used criterion is maximal orthogonal subspace projection not maximal simplex volume. As a result, VCA does not satisfy the sum to one abundance constraint (ASC) compared to simplex volume criterion which satisfies both convexity constraints, ANC and ASC imposed on extraction of endmembers. On the other hand, due to the fact that both VCA and ATGP use maximal orthogonal subspace projection, ATGP indeed performs better than VCA does in endmember extraction on many cases because ATGP finds the maximal orthogonal projection over the entire data space orthogonal complement to the space linearly spanned by already found endmembers compared to VCA which restricts its search region to convex hulls in the first quadrant. This may result in the fact that some ATGP-extracted samples may not be in the first quadrant. But according to our experiments, such case did not occur often. Additionally, as an unsupervised target detection algorithm, VCA does not perform as well as ATGP. Consequently, in either case of endmember extraction or unsupervised target detection, ATGP is generally doing better than VCA. Finally, as for computational complexity, both VCA and ATGP perform orthogonal projections and their computing time is very fast. Because of that, VCA may have an edge over simplex volume-based techniques in the sense that only orthogonal projections are performed compared to calculating matrix determent for a simplex volume. But this advantage is poorly traded for its performance in endmember extraction.

11.2.2 Experiments-Based Comparative Study and Analysis

In order to conduct a comprehensive comparative analysis, two data sets from the real AVIRIS Cuprite image data shown in Figure 1.12(c) and (d) are used to simulate TI2. Since TI1 has no noise simulated in the image and TI3 contains endmembers corrupted by Gaussian noise, only TI2 is selected for experiments because it simulates exactly pure signatures as endmembers implanted in the image scene.

11.2.2.1 Synthetic Image Experiment: TI2

The reflectance data of Figure 1.12(c) is used for experiments where the VD estimated for this data set by the Hrasanyi–Farrend–Chang method is $n_{\mathrm{VD}} = 6$ with $\mathrm{P_F} \leq 10^{-1}$. In this case, the value of 6 is used as the number of endmembers, $p = 6$, as well as the number of dimensions or components required for DR, $q = 6$. But in fact, $q = p-1$. So in experiments, these two values are used for experiments.

Reflectance Data

Table 11.1 tabulates endmember pixels extracted by three OP-based EEAS along with two versions of PPI and VCA using ATGP-generated pixels as initial endmembers, ATGP-PPI and ATGP-VCA, respectively, where four transforms, PCA, MNF, SVD, and ICA, are implemented for DR with $q = 6$ (Note: the DR techniques used by the original VCA is SVD). Here, an endmember pixel is defined as a pixel whose spectral signature is an endmember and the number of skewers used by PPI is $K = 200$.

Table 11.1 Endmembers extraction by PPI, ATGP, VCA, ATGP-PPI, and ATGP-VCA (TI2 with reflectance data)

EEAs	Endmembers corresponding to five minerals							
	$K = 200$				$p = 6$			
PPI	PCA $(q = 6)$ a, b, c, k, m	MNF $(q = 6)$ a, b, c, k, m	ICA $(q = 6)$ a, b, c, k, m	SVD $(q = 5)$ a, b, c, k, m	N/A			
ATGP	N/A				a, b, c, k, m			
VCA	N/A				PCA $(q = 6)$ a, b, c, k, m	MNF $(q = 6)$ a, b, c, k, m	ICA $(q = 6)$ a, b, c, k, m	SVD $(q = 6)$ a, b, c, k, m
ATGP-PPI	N/A				PCA $(q = 6)$ a, b, c, k, m	MNF $(q = 6)$ a, b, c, k, m	ICA $(q = 6)$ a, b, c, k, m	SVD $(q = 6)$ a, b, c, k, m
ATGP-VCA	N/A				PCA $(q = 6)$ a, b, c, k, m	MNF $(q = 6)$ a, b, c, k, m	ICA $(q = 6)$ a, b, c, k, m	SVD $(q = 6)$ a, b, c, k, m

Interestingly, the level of knowledge required by the three algorithms, PPI, ATGP, and VCA, can be closely related to one another. Despite that PPI does not require the knowledge of the number of endmembers, p, it does need to know the value of q, the number of dimensions or components when it performs DR. Quite oppositely, VCA and ATGP only require to know the number of target pixels they must generate, in which case this number was set to p and q was set to be equal to p, that is, $p = q$.

According to the results of Table 11.1, all the five algorithms performed comparably. In particular, the results demonstrated that $p = 6$ was sufficient for the five algorithms to successfully extract all the five mineral endmembers. It is also worth noting that since both the PPI and the VCA used p randomly generated vectors as their initial endmembers, the final results were not consistent. As a result, on some occasions, they may not extract all the five endmembers. The results of the PPI and VCA in Table 11.1 were obtained by their best runs.

Radiance Data

As another example, a second synthetic image was simulated by radiance data in exactly the same manner as the first synthetic image, TI2, except that the five mineral signatures were the radiance data in Figure 1.12(d) instead of the reflectance data used in Figure 1.12(c).

Interestingly, VD estimated for this radiance data-based synthetic image, TI2, was $n_{\text{VD}} = 5$ with $P_F \leq 10^{-1}$ instead of six estimated for the reflectance data-based synthetic image. In other words, the background simulated by the sample mean of the radiance data-based synthetic image was not considered as an endmember as it was where the background was simulated by the sample mean of the reflectance data-based synthetic image due to our belief that the reflectance data are calibrated. So, the value of p was set to 5 and the number of dimensions or components required for DR was also set to $q = 5$.

Table 11.2 Endmembers extraction by PPI, ATGP, VCA, ATGP-PPI, and ATGP-VCA (TI2 with radiance data)

EEAs	Endmembers corresponding to five minerals							
	$K=200$				$p=5$			
PPI	PCA ($q=5$) b, c, k, m	MNF ($q=5$) a, b, c, k, m	ICA ($q=5$) a, b, c, k, m	SVD ($q=5$) a, b, c, k, m	N/A			
ATGP	N/A				a, b, c, k, m			
VCA	N/A				PCA ($q=5$) b, k, m	MNF ($q=5$) a, b, c, k, m	ICA ($q=5$) a, b, c, k, m	SVD ($q=5$) b, k, m
ATGP-PPI	N/A				PCA ($q=5$) a, b, c, k, m	MNF ($q=5$) a, b, c, k, m	ICA ($q=5$) a, b, c, k, m	SVD ($q=5$) a, b, c, k, m
ATGP-VCA	N/A				PCA ($q=5$) b, k, m	MNF ($q=5$) a, b, c, k, m	ICA ($q=5$) a, b, c, k, m	SVD ($q=5$) b, m

Table 11.2 tabulates endmember pixels extracted by three OP-based EEAS along with two versions of the PPI and VCA using ATGP-generated pixels as initial endmembers, ATGP-PPI and ATGP-VCA, respectively, where three transforms, PCA, MNF, and ICA, were implemented for DR with $q=5$ and the results highlighted by shade indicate failures of extracting all five endmembers.

As shown in the table, the best performance was ATGP, while the worst one was VCA. Two observations are noteworthy. One is that ATGP-VCA using ATGP as its initialization algorithm did not improve its performance over VCA with the use of random initial endmembers even though ATGP was the one that yields the best performance. This implies that the target sample vectors generated by ATGP that were supposed to be endmember pixels were actually compromised by the process of finding the maximal OP implemented in VCA where ATGP-generated target pixels were replaced by the data sample vectors that yielded the maximal OPs at each iteration but turned out to be not endmembers. This finding further demonstrated that the maximal OP used by VCA as a criterion was ineffective and may not be as good as the criterion of using maximal simplex volume. On the contrary, ATGP did improve PPI with PCA where ATGP-generated target sample vectors helped PPI find correct projection directions due to the reason that PCA compromised endmembers by retaining second-order statistics in a few principal components, while endmembers should be characterized by high-order statistics. A comparison of results between Tables 11.1 and 11.2 indicates that the radiance data set presented more challenges for an EEA than the reflectance data set.

11.2.2.2 Real Image Experiments

In this section, we repeat the same synthetic image experiments conducted in Section 11.2.2.1.3 for two real hyperspectral image scenes, HYDICE data in Figure 1.15 (a) and Cuprite data in Figure 1.11(b) for experiments. It has been shown in Chang (2003a), Chang and Du (2004), and Chang et al. (2006) that a good VD estimate for the HYDICE data was $n_{\mathrm{VD}}=9$ with P_F equal to or smaller

Table 11.3 Endmembers extraction by PPI, ATGP, VCA, ATGP-PPI, and ATGP-VCA (HYDICE data)

EEAs	Endmembers corresponding to five minerals							
	$K = 200$				$p = 9$			
PPI	PCA ($q = 9$) $\mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$	MNF ($q = 9$) $\mathbf{p}_1, \mathbf{p}_4, \mathbf{p}_5$	ICA ($q = 9$) $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$	SVD ($q = 9$) $\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_5$	N/A			
ATGP	N/A				$\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_5$			
VCA	N/A				PCA ($q = 9$) $\mathbf{p}_3, \mathbf{p}_5$	MNF ($q = 9$) $\mathbf{p}_3, \mathbf{p}_5$	ICA ($q = 9$) $\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$	SVD ($q = 9$) $\mathbf{p}_3, \mathbf{p}_5$
ATGP-PPI	N/A				PCA ($q = 9$) $\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_5$	MNF ($q = 9$) $\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_5$	ICA ($q = 9$) $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$	SVD ($q = 9$) $\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_5$
ATGP-VCA	N/A				PCA ($q = 9$) $\mathbf{p}_1, \mathbf{p}_3, \mathbf{p}_5$	MNF ($q = 9$) $\mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$	ICA ($q = 9$) $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \mathbf{p}_4, \mathbf{p}_5$	SVD ($q = 9$) $\mathbf{p}_3, \mathbf{p}_5$

than 10^{-3} by the HFC method. By letting $p = q = 9$ Table 11.3 tabulates the extracted endmembers by PPI with $K = 200$, ATGP, VCA, ATGP-PPI, and ATGP-VCA that correspond to the five panel signatures in Figure 1.16.

According to Table 11.3, it is obvious that for PPI, ATGP-PPI, and ATGP-VCA to be able to extract all the five panel signatures ICA must be used to perform DR prior to endmember extraction except the case that VCA used alone with ICA as DR extracted only four not five panel signatures. More interestingly, without using ICA as DR none of PPI and VCA along with their ATGP versions could extract more than three panel signatures if any of PCA, MNF, and SVD was used to perform DR. This simple example demonstrates two important facts. One is that DR is a crucial preprocessing step for endmember extraction. The other is that endmembers can be better characterized by high-order statistics-based DR transforms than second-order statistics-based DR transforms.

As for Cuprite image scene in Figures 1.11(b) and 1.12 with reflectance data and radiance data it has been shown in Chang et al. (2006) and Chapter 4 that for the image with reflectance data $n_{\mathrm{VD}} = 22$ with $P_F = 10^{-4}$ is a good estimate for the number of endmembers, $p = 22$. In order to make a fair comparison, the same false alarm probability $P_F = 10^{-4}$ chosen for the image with reflectance data was also selected for the real image with radiance data, in which case the $n_{\mathrm{VD}} = 15$ according to Table 11.4.

Tables 11.5 and 11.6 tabulate endmember pixels extracted by PPI, ATGP, and VCA along with two versions of the PPI and VCA using ATGP-generated pixels as initial endmembers, ATGP-PPI

Table 11.4 VD-estimated value for the Cuprite scene with various false alarm probabilities

n_{VD}	$P_F = 10^{-1}$	$P_F = 10^{-2}$	$P_F = 10^{-3}$	$P_F = 10^{-4}$	$P_F = 10^{-5}$
Reflectance data	34	30	24	22	20
Radiacne data	29	18	17	15	15

Table 11.5 Endmembers extraction by PPI, ATGP, VCA, ATGP-PPI, and ATGP-VCA (reflectance data)

EEAs	Endmembers corresponding to five minerals							
	$K = 1000$				$p = 22$			
PPI	PCA $(q = 22)$ a, b, c, k, m	MNF $(q = 22)$ a, b, c, k, m	ICA $(q = 22)$ a, b, c, k, m	SVD $(q = 22)$ a, b, c, k, m	N/A			
ATGP	N/A				a, b, c, k, m			
VCA	N/A				PCA $(q = 22)$ b, c, k, m	MNF $(q = 22)$ a, b, c, k, m	ICA $(q = 22)$ a, b, c, k, m	SVD $(q = 22)$ a, b, c, m
ATGP-PPI	N/A				PCA $(q = 22)$ a, b, c, k, m	MNF $(q = 22)$ a, b, c, k, m	ICA $(q = 22)$ a, b, c, k, m	SVD $(q = 22)$ a, b, c, k, m
ATGP-VCA	N/A				PCA $(q = 22)$ a, c, k, m	MNF $(q = 22)$ a, c, k	ICA $(q = 22)$ a, b, c, k, m	SVD $(q = 22)$ b, c, k, m

and ATGP-VCA, respectively, where four transforms, PCA, MNF, SVD, and ICA, were implemented for DR with $q = 22$ in Table 11.5 and $q = 15$ in Table 11.6. The results highlighted by shade in Tables 11.5 and 11.6 indicate that the algorithm failed to extract all the five mineral signatures.

In analogy with synthetic image experiments the results obtained for real image data in Tables 11.5 and 11.6 also confirmed that the real image with radiance data were generally more difficult

Table 11.6 Endmembers extraction by PPI, ATGP, VCA, ATGP-PPI, and ATGP-VCA (radiance data)

EEAs	Endmembers corresponding to five minerals							
	$K = 1000$				$p = 15$			
PPI	PCA $(q = 15)$ a, b, c, k, m	MNF $(q = 15)$ a, b, k, m	ICA $(q = 15)$ a, b, c, k, m	SVD $(q = 15)$ a, b, k	N/A			
ATGP	N/A				a, b, k, m			
VCA	N/A				PCA $(q = 15)$ a, c, k, m	MNF $(q = 15)$ a, c, m	ICA $(q = 15)$ a, b, c, k, m	SVD $(q = 15)$ a, k, m
ATGP-PPI	N/A				PCA $(q = 15)$ a, b, k, m	MNF $(q = 15)$ a, b, k, m	ICA $(q = 15)$ a, b, k, m	SVD $(q = 15)$ a, b, k, m
ATGP-VCA	N/A				PCA $(q = 15)$ k, m	MNF $(q = 15)$ a, b, c, k, m	ICA $(q = 15)$ a, b, k, m	SVD $(q = 15)$ a, c, k, m

and challenging to deal with than the real image with reflectance data. Four interesting and intriguing findings are also observed in Tables 11.5 and 11.6.

1. VCA has received attention recently and shown promise in endmember extraction. The experiments conducted in this chapter demonstrated otherwise. The results in both synthetic and real image experiments suggested that VCA was always the worst among all the three OP-based algorithms (i.e., PPI, ATGP, and VCA). Moreover, PPI with a sufficient number of skewers and ATGP seemed to produce best results in general. Two causes may be attributed to the poor performance of VCA. One is its use of random initial endmembers, and the other is an insufficient number of vertices required by VCA. In fact, these two causes are closely related. In order to overcome the uncertainty resulting from the use of random initial endmembers, the number of vertices to be used by VCA must be sufficiently large. That explains why PPI requires a large number of skewers to perform well and better than VCA. On the other hand, in order to eliminate randomness caused by the use of random initial endmembers, an appropriate set of initial endmembers must be preselected for VCA to perform well if the number of vertices to be used is small and the selected initial endmembers must be more reliable and less random. This is the main reason why ATGP performed well and better than VCA. Unfortunately, VCA inherits the nature of PPI that is the use of random initial endmembers as well as another nature from ATGP that is a small number of endmembers determined by VD, p; both actually contradicts each other. This implies that if random endmembers are used, its number should be sufficiently large to overcome randomness. Such a dilemma can be resolved in two ways. One is that more vertices must be needed to deal with the use of random initial endmembers as required by PPI at the expense of more falsely alarmed endmembers. The other is to use an initialization algorithm to produce an appropriate set of initial endmembers to remove randomness as ATGP does. In this case, a reliable and accurate estimate of the number of endmembers, p, must be provided *a priori*. Generally, VD provides a good estimate of p as shown by experiments conducted for the synthetic images and real image data with reflectance values. However, for real image experiments with radiance values, this number seemed insufficient. So, if the value of p is chosen to be $2p = 30$, that is, twice the value estimated by VD, VCA as well as ATGP-VCA indeed performed well and successfully extracted all endmembers as shown in Table 11.7. A transform such as PCA or ICA is appropriately selected for DR.
2. Secondly, as also demonstrated in above-mentioned experiments, ATGP generally performs better than VCA. However, an unexpected and interesting finding is that ATGP-VCA could not improve the performance of VCA. The reason for this is the following. VCA finds endmembers

Table 11.7 Endmembers extraction by ATGP, VCA, ATGP-PPI, and ATGP-VCA (radiance data)

EEAs	Endmembers corresponding to five minerals			
ATGP	$p = 30$ a, b, c, k, m			
VCA	PCA ($q = 30$) a, b, c, k, m	MNF ($q = 30$) a, b, c, k	ICA ($q = 30$) a, b, c, k, m	SVD ($q = 30$) a, b, k, m
ATGP-PPI	PCA ($q = 30$) a, b, c, k, m	MNF ($q = 30$) a, b, c, k, m	ICA ($q = 30$) a, b, c, k, m	SVD ($q = 30$) a, b, c, k, m
ATGP-VCA	PCA ($q = 30$) a, b, c, k, m	MNF ($q = 30$) a, c, k, m	ICA ($q = 30$) a, b, c, k, m	SVD ($q = 30$) a, c, k, m

via growing convex hulls by a sequence of maximal orthogonal projections. Its found convex hulls do not necessarily produce maximal volumes as N-FINDR which produces simplexes with maximal volumes. Therefore, VCA-found vertices may not be the same as those found by N-FINDR. As a result, technically speaking, VCA-found maximal volume is based on a sequence of successively generated convex hulls and is actually not the true maximal volume for a given number of endmembers, *p*. That is the reason why ATGP-generated pixels used initial endmember pixels that are supposed to be endmember pixels but are unfortunately replaced by subsequent VCA-generated pixels that are not endmember pixels during the process because VCA looks for endmembers to produce maximal orthogonal projections not maximal convex hull volumes.

3. Thirdly, ATGP, which is originally developed for target detection and classification and never considered as an EEA, can be made a very effective EEA. On many occasions, it even outperforms VCA.
4. Fourthly, according to the results provided by Tables 11.5 and 11.6, ICA is the best DR transform among all four DR transforms used for endmember extraction regardless of which endmember extraction algorithm is used. However, it should be mentioned that ATGP alone does not require DR.

In summary, this section investigates three seemingly different algorithms PPI, VCA, and ATGP, in endmember extraction and explores their relationships. The insights into such relationships are very enlightening. In particular, it has shown that VCA is nothing more than an orthogonal projection-based endmember extraction algorithm that can be considered as a variant of PPI and ATGP in either way. However, the connection of PPI to VCA is not clear until ATGP fits in between and bridges their gap. In doing so, this section re-interprets PPI, ATGP, and VCA from the principle of orthogonality. Most importantly, an in-depth study is also conducted to reveal close relationships among these three algorithms via orthogonal projection. Since both PPI and VCA use randomly generated vectors as their initial endmembers, two issues become evidential: (1) inconsistency in final results resulting from random initial conditions and (2) accurate number of endmembers, *p*. In order to address the first issue, two variants of PPI and VCA are also introduced in this chapter, called ATGP-PPI and ATGP-VCA that implements the ATGP as their initialization algorithm to produce a set of target pixels to be used as their initial endmembers. It is also interesting to note that the ATGP alone can also be considered as an initialization algorithm for any arbitrary EEA to produce a better set of initial endmembers. Finally, these five algorithms are further evaluated and compared via synthetic and real image experiments for performance analysis. The results showed that using such ATGP-generated target sample vectors as initial endmembers for any EEA resulted in only a small change in final selected set of endmembers. A surprising finding is that ATGP that is primarily developed for automatic target detection and classification indeed performed very well in endmember extraction and even outperformed VCA in most cases. A second surprising finding is that according to conducted experiments VCA did not perform as well as it was originally designed for. Similar observations are also demonstrated in Chang et al. (2006) and Wu et al. (2009).

Finally, if we further construct two *p*-vertex simplexes with their *p* vertices specified by the *p* endmembers obtained by ATGP and VCA, respectively, we can calculate their corresponding simplex volumes via (7.3) in comparison with the volume calculated from a simplex formed by the *p* endmembers generated by N-FINDR. If the volume of ATGP-generated simplex is close to that of N-FINDR-generated simplex, it implies that the *p* ATGP-generated endmembers can be considered as desired endmembers, even though these endmembers may not be the same as the endmembers generated by N-FINDR. Similarly, it can be also applied to VCA-generated simplex.

11.3 Comparative Study and Analysis Between SGA and VCA

Two major design criteria, OP and simplex volume, have been widely used to design EEAs. In Section 11.2, relationships among three popular algorithms, PPI, VCA, and ATGP, were explored from a perspective of OP. As also demonstrated in Section 11.2, simplexes formed by VCA-found endmembers did not necessarily yield maximal simplex volumes. This is because maximal OP does not imply maximal simplex volume. This fact is further confirmed and supported in the following experiments. In this section, we follow a similar treatment from a perspective of simplex volume. Despite that there are several simplex volume-based EEAs, N-FINDR remains the most popular EEA that has been used as a base to derive new EEAs. However, as discussed in Chapter 7, N-FINDR has several difficulties with practical implementation. As a matter of fact, many simplex volume-based algorithms claimed to be implemented as N-FINDR are not its original version but actually sequential versions of N-FINDR. Recently, a rather different sequential algorithm, called simplex growing algorithm (SGA) proposed by Chang et al. (2006), finds endmembers one after another by growing simplexes with the maximal volumes one vertex at a time. It can be considered as a sequential N-FINDR in the same way as VCA is considered as PPI by growing convex hulls with the maximal orthogonal projections one vertex at a time. This similarity allows us to make a fair and very interesting comparison between these two criteria via their respective growing convex hull algorithm, VCA and growing simplex algorithm SGA. Those who are interested in comparisons between SGA and N-FINDR should refer to Chapters 7 and 8 as well as Chang et al. (2006) and Wu et al. (2009).

Once again, three airborne visible/infrared imaging spectrometer (AVIRIS) Cuprite data sets are used to conduct comparative study and analysis between VCA and SGA. One set is the reflectance laboratory data consisting of five mineral spectra with 224 bands, alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M), shown in Figure 1.9 and replotted in Figure 11.2,

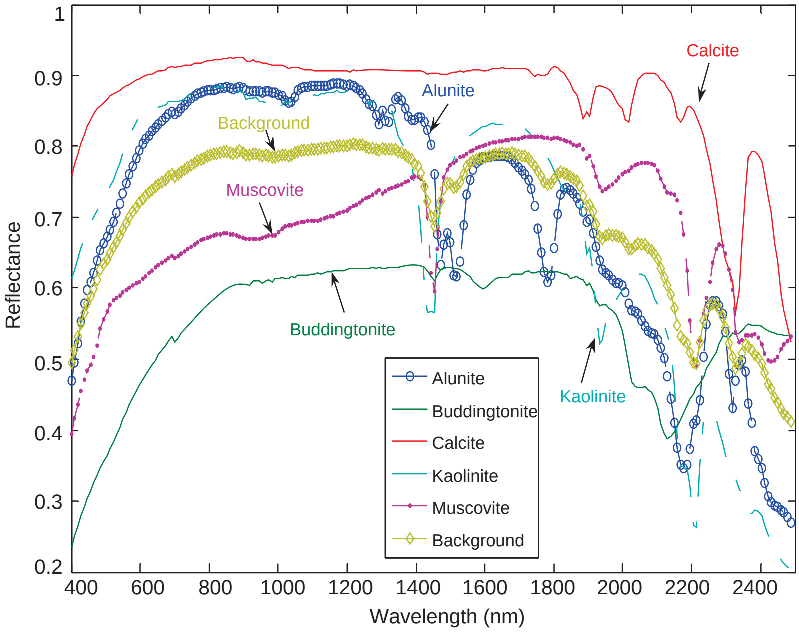
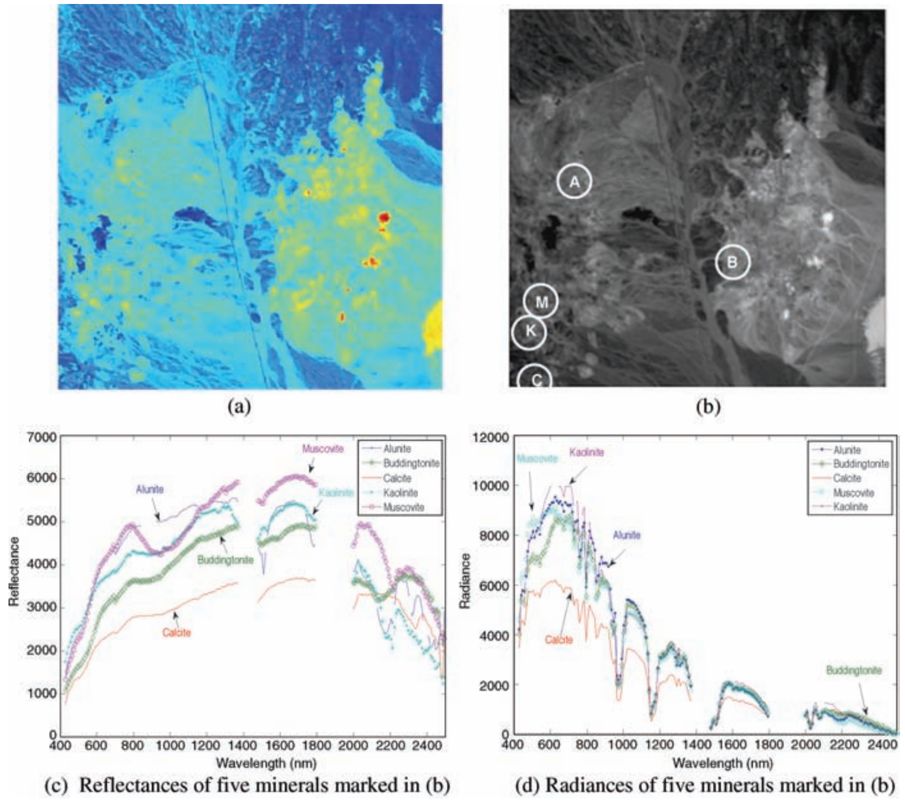


Figure 11.2 Spectra of five pure pixels corresponding to minerals: alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M) provided by the USGS plus a background signature obtained by equally mixing all the five mineral spectra.



(a)

(b)

(c) Reflectances of five minerals marked in (b)

(d) Radiances of five minerals marked in (b)

Figure 11.3 (a) Spectral band number 170 of the cuprite AVIRIS image scene; (b) spatial positions of five pure pixels corresponding to minerals: alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M); (c) reflectances of five minerals marked in (b) in wavelengths; (d) radiances of five minerals marked in (b).

along with a background signature obtained by equally mixing all the five mineral spectra. Two other data sets obtained directly from the real AVIRIS Cuprite image data shown in Figure 1.12(a) and (b) with their corresponding reflectance values and radiance values shown in Figure 1.12(c) and (d) will be also used for experiments. Figure 11.3 is a reproduction of Figure 1.12(a)–(d) for reference.

By means of these three data sets five different scenarios are designed to show interesting and quite different results produced by SGA and VCA.

Scenario 1 (using the data in Figure 11.2)

First of all, we generate a synthetic image that has the size of 200×200 pixel vectors with 25 panels of various sizes that are arranged in a 5×5 matrix and located at the center of the scene shown in Figure 11.4(a).

The five mineral spectral signatures, $\{\mathbf{m}_i\}_{i=1}^5$ in Figure 11.2 were used to simulate these 25 panels where each row of five panels was simulated by the same mineral signature and each column of five panels has the same size. Among 25 panels are five 4×4-pure pixel panels for endmember extraction, $p^i_{4\times4}$ for $i = 1, \ldots, 5$ in the first column, 5 2×2-pure pixel panels for training samples, $p^i_{2\times2}$ for $i = 1, \ldots, 5$ in the second column, five 2×2-mixed pixel panels, $\{p^i_{3,jk}\}_{j=1,k=1}^{2,2}$ for $i = 1, \ldots, 5$ in the third column for mixed pixel classification, five subpixel panels, $p^i_{4,1}$ for

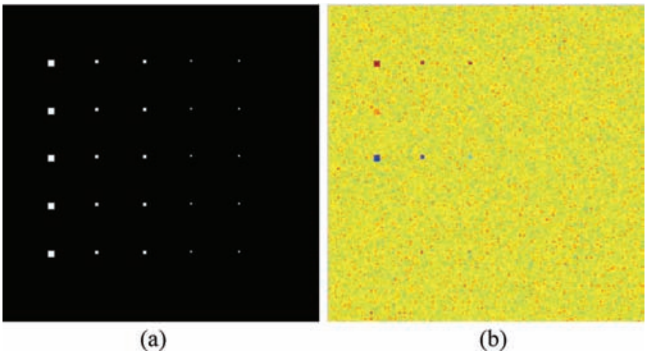


Figure 11.4 Scenario 1: (a) 25 simulated panels; (b) a synthetic image having the 25 panels simulated in (a) implanted in the background with an additive Gaussian noise to achieve SNR 20 : 1.

$i = 1, \ldots, 5$ in the fourth column for subpixel classification, and five sub-pixel panels, $p^i_{5,1}$ for $i = 1, \ldots, 5$ in the fifth column for subpixel classification. The reason that the five panels in the second column are included in the image scene is to use them as training samples for supervised classification. The purpose of introducing five panels in the third column is to conduct a study and analysis on five mineral signatures with different mixing in a pixel. Table 11.8 tabulates the mixing details of mineral composition in 20 panels.

The inclusion of the panels in the fourth and fifth columns is to investigate subpixel effects on endmember extraction, and their simulated abundance fractions are tabulated in Table 11.9 where

Table 11.8 Simulated 20 mixed panel pixels in the third column

Row 1	$p^1_{3,11} = 0.5\mathrm{A} + 0.5\mathrm{B}$	$p^1_{3,12} = 0.5\mathrm{A} + 0.5\mathrm{C}$
	$p^1_{3,21} = 0.5\mathrm{A} + 0.5\mathrm{K}$	$p^1_{3,22} = 0.5\mathrm{A} + 0.5\mathrm{M}$
Row 2	$p^2_{3,11} = 0.5\mathrm{A} + 0.5\mathrm{B}$	$p^2_{3,12} = 0.5\mathrm{B} + 0.5\mathrm{C}$
	$p^2_{3,21} = 0.5\mathrm{B} + 0.5\mathrm{K}$	$p^2_{3,22} = 0.5\mathrm{B} + 0.5\mathrm{M}$
Row 3	$p^3_{3,11} = 0.5\mathrm{A} + 0.5\mathrm{C}$	$p^3_{3,12} = 0.5\mathrm{B} + 0.5\mathrm{C}$
	$p^3_{3,21} = 0.5\mathrm{C} + 0.5\mathrm{K}$	$p^3_{3,22} = 0.5\mathrm{C} + 0.5\mathrm{M}$
Row 4	$p^4_{3,11} = 0.5\mathrm{A} + 0.5\mathrm{K}$	$p^4_{3,12} = 0.5\mathrm{B} + 0.5\mathrm{K}$
	$p^4_{3,21} = 0.5\mathrm{C} + 0.5\mathrm{K}$	$p^4_{3,22} = 0.5\mathrm{K} + 0.5\mathrm{M}$
Row 5	$p^5_{3,11} = 0.5\mathrm{A} + 0.5\mathrm{M}$	$p^5_{3,12} = 0.5\mathrm{B} + 0.5\mathrm{M}$
	$p^5_{3,21} = 0.5\mathrm{C} + 0.5\mathrm{M}$	$p^5_{3,22} = 0.5\mathrm{K} + 0.5\mathrm{M}$

Table 11.9 Simulated 20 mixed panel pixels in the third column

Row	Fourth column	Fifth column
Row 1	$p^1_{4,11} = 0.5\mathrm{A} + 0.5\mathrm{BKG}$	$p^1_{5,11} = 0.25\mathrm{A} + 0.75\mathrm{BKG}$
Row 2	$p^2_{4,11} = 0.5\mathrm{B} + 0.5\mathrm{BKG}$	$p^2_{5,11} = 0.25\mathrm{B} + 0.75\mathrm{BKG}$
Row 3	$p^3_{4,11} = 0.5\mathrm{C} + 0.5\mathrm{BKG}$	$p^3_{5,11} = 0.25\mathrm{C} + 0.75\mathrm{BKG}$
Row 4	$p^4_{4,11} = 0.5\mathrm{K} + 0.5\mathrm{BKG}$	$p^4_{5,11} = 0.25\mathrm{K} + 0.75\mathrm{BKG}$
Row 5	$p^5_{4,11} = 0.5\mathrm{M} + 0.5\mathrm{BKG}$	$p^5_{5,11} = 0.25\mathrm{M} + 0.75\mathrm{BKG}$

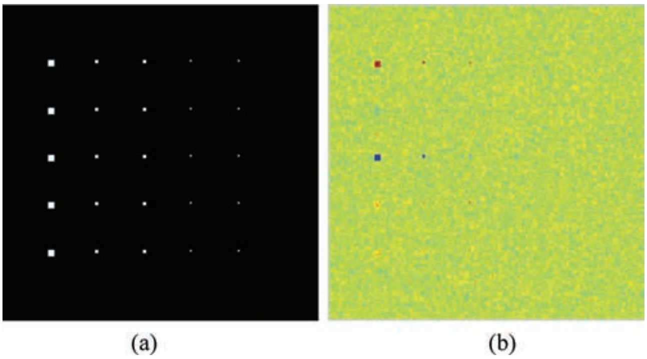


Figure 11.5 Scenario 2: (a) 25 simulated panels; (b) a synthetic image having the 25 panels simulated in (a) implanted in the background with an additive Gaussian noise to achieve SNR 20 : 1.

the background (BKG) is simulated by mixing 20% of each of five mineral signatures, A, B, C, K, and M, that is, 20%A+20%B+20%C+20%K+20%M shown in Figure 11.2. So, there are 100 pure pixels, 20 mixed pixels, and 10 subpixels, all of which are simulated by five distinct pure mineral signatures. These 25 panels in Figure 11.4(a) are then implanted in the image background in a way that the background pixels are replaced with the implanted panel pixels shown in Figure 11.4(b) where the image background is specified by BKG.

Scenario 2 (using the data in Figure 11.3(c))

This scenario is exactly the same as Scenario 1 except that the reflectance laboratory data in Figure 11.2 used to simulate the synthetic image in Figure 11.4 was replaced with the reflectance spectra of the real image in Figure 11.3(c). The resulting images are shown in Figure 11.5.

Scenario 3 (using the data in Figure 11.2 and sample mean in Figure 11.3(b)).

This scenario is more interesting than Scenarios 1 and 2 in the sense that the background signature was not simulated by equally mixing the five mineral signatures. Instead, the background signature was simulated by the sample mean of the entire real image in Figure 12.3(b), while all the implanted 25 panels are exactly the same as those simulated in Scenario 2. The images resulting from Scenario 3 are shown in Figure 11.6.

Figure 11.6 also shows a comparison between the spectral signatures of the two background signatures used in Scenarios 2 and 3.

Scenario 4 (using the data in Figure 11.3(d))

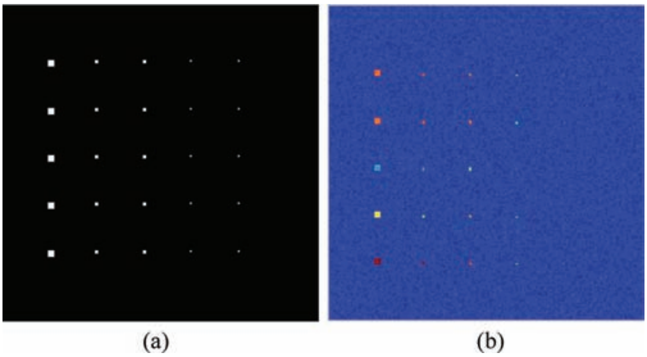


Figure 11.6 Reflectance spectra of two background signatures used in Scenarios 2 and 3.

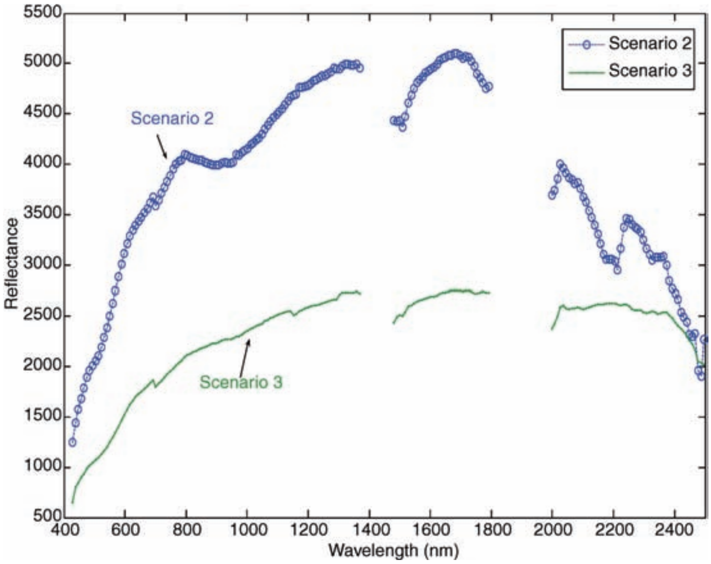


Figure 11.7 Scenario 4: (a) 25 simulated panels; (b) a synthetic image having the 25 panels simulated in (a) implanted in the background with an additive Gaussian noise to achieve SNR 20 : 1.

This scenario is exactly the same as Scenario 2 except that the reflectance data in Figure 11.3(c) used to simulate the synthetic image in Figure 11.5 was replaced with the radiance spectra of the real image in Figure 11.3(d). The resulting images are shown in Figure 11.7.

Scenario 5 (using the sample mean in Figure 11.3(b) and data in Figure 11.3(d))

This scenario is also exactly the same as Scenario 3 with the reflectance data in Figure 11.3(c) replaced with the radiance spectra in Figure 11.3(d). The images resulting from Scenario 5 are shown in Figure 11.8.

Figure 11.9 shows a comparison between the spectral signatures of the two background signatures used in Scenarios 4 and 5.

Table 11.10 tabulates different values of p estimated by VD with various false alarm probabilities, P_F as well as by signal subspace estimation (SSE) developed by Bioucas-Dias and Nascimento

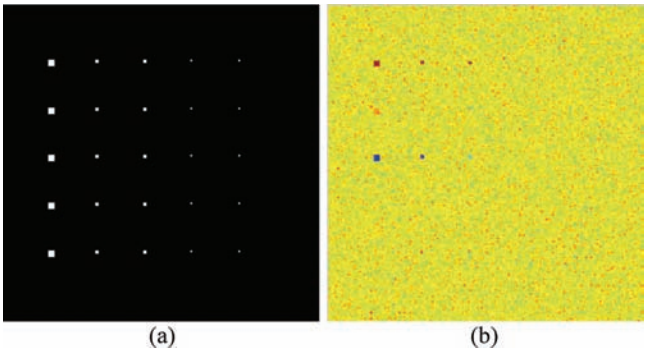


Figure 11.8 Scenario 5: (a) 25 simulated panels; (b) a synthetic image having the 25 panels simulated in (a) implanted in the background with an additive Gaussian noise to achieve SNR 20 : 1.

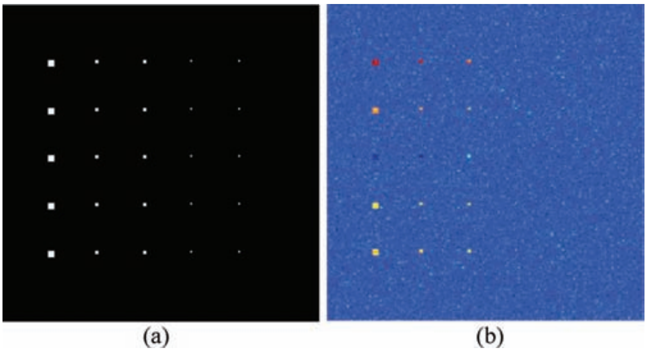


Figure 11.9 Radiance spectra of two background signatures used in Scenarios 4 and 5.

(2005) for the five scenarios. Despite that an improved version of SSE, called hyperspectral signal subspace identification by minimum error (HySime), was also developed by the same authors (Bioucas-Dias and Nascimento, 2008) in Section 5.4.2, we include SSE instead of HySime in the following experiments for the simple reason that both SSE and VCA were developed by the same authors nearly at the same time. However, for those who are interested in these two criteria more detailed discussions can be found in Sections 5.4.2 and 5.4.3.

Figures 11.10 and 11.11 show endmember extraction results by VCA and SGA, respectively, using four different dimensionality reduction techniques, PCA, MNF, SVD, and ICA for visual assessment where the numbers labeling the pixels in the figures indicate the order that the pixels are extracted by algorithms and the numbers in parentheses are values of the p estimated by either SSE or VD.

As we can see from Figures 11.10 and 11.11, the best among five scenarios was Scenario 3 for which both SGA and VCA produced their best results, while both criteria also worked at their best in estimating the values of p. Coincidently, Scenario 3 is exactly the case discussed in Chang et al. (2006) when it is designed for experiments. In addition, several interesting and intriguing findings are noteworthy.

1. VD always produced better estimates than SSE did regardless of which scenario is used. Interestingly, the five scenarios produced different values of p. This implies that the determination of

Table 11.10 Values of p estimated by VD and SSE

	VD ($P_F = 10^{-1}$)	VD ($P_F = 10^{-2}$)	VD ($P_F = 10^{-3}$)	VD ($P_F = 10^{-4}$)	SSE
Scenario 1					
Lab data (signature mean)	4	4	4	4	3
Scenario 2					
Reflectance (signature mean)	3	3	3	3	2
Scenario 3					
Reflectance (sample mean)	6	6	6	6	5
Scenario 4					
Radiance (signature mean)	5	4	4	4	1
Scenario 5					
Radiance (sample mean)	5	5	5	5	2

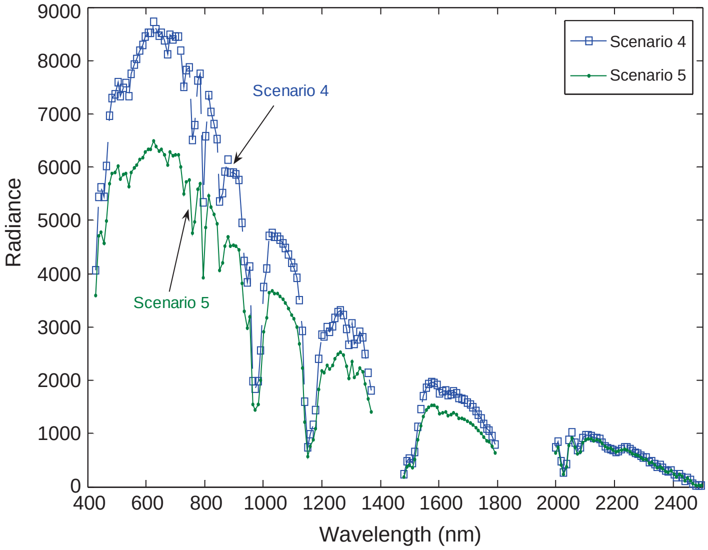
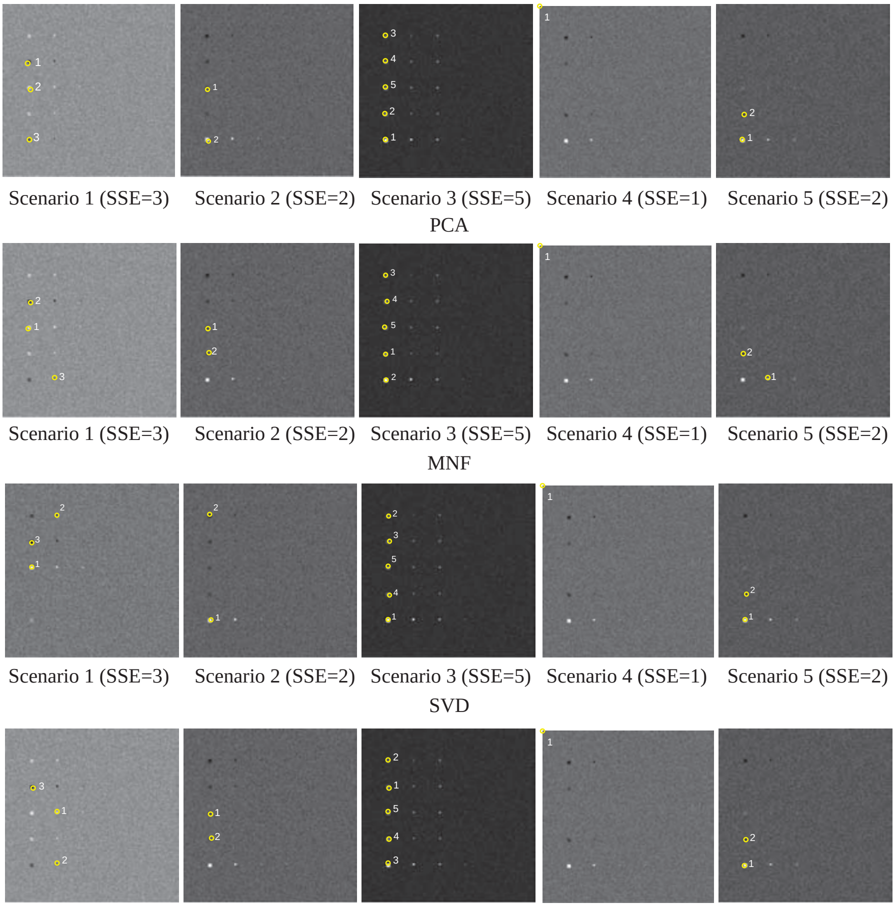


Figure 11.10 Panel pixels extracted as endmembers by VCA.

p was heavily dependent on the data to be processed. Additionally, it also demonstrated that estimation of the *p* by VD and SSE for scenarios using sample means as background was better than those using signature means. This made sense since the sample mean is a mixture of all signatures in the entire image that is generally more spectrally distinct from the five mineral signatures than the signature mean that contains 20% of each of five mineral signatures. Therefore, in this case, the sample mean represents a spectral class corresponding to the background, and thus may be considered as an endmember even if it may not be a pure signature. This explained why the background signature was extracted as an endmember in many cases in Scenarios 3 and 5. Therefore, the simulated synthetic scene should have six endmembers, five of which represent five distinct mineral signatures and one endmember specifies the background. Also, Scenario 3 is only the case that the value of the *p* estimated by SSE was correct. However, it has been shown in Chang et al. (2006) and Wu et al. (2009) that in this particular scenario there was no way to extract all the five minerals by either SGA or VCA using $p = 5$ because the last extracted mineral signature is always the sixth signature to be extracted. In other words, a background pixel was always among the first five pixels extracted by SGA and VCA. This indicates that the background must be considered and included as one endmember. So, when VD estimated the value of *p* to be 6, all the five distinct mineral signatures could be successfully extracted by SGA and VCA.

2. When reflectance data are used (Scenarios 1–3), the values estimated by SSE were always one less than the values estimated by VD. However, for radiance data used in Scenarios 4–5 SSE seemed to not work at all, while VD still worked well. This implies that VD was a much more reliable technique in estimating the value of *p* than SSE when data to be processed are real data.
3. For any estimated value of the *p*, SGA performed at least as well as VCA did. As a matter of fact, on many occasions SGA actually outperformsed the VCA.
4. There are two reasons that VCA could not perform as well as SGA did. One was its use of a random initial endmember every time it generated a new endmember. In order to improve its performance, VCA must need more than *p* vertices to find a desired set of endmembers.



Scenario 1 (SSE=3) Scenario 2 (SSE=2) Scenario 3 (SSE=5) Scenario 4 (SSE=1) Scenario 5 (SSE=2)

PCA

Scenario 1 (SSE=3) Scenario 2 (SSE=2) Scenario 3 (SSE=5) Scenario 4 (SSE=1) Scenario 5 (SSE=2)

MNF

Scenario 1 (SSE=3) Scenario 2 (SSE=2) Scenario 3 (SSE=5) Scenario 4 (SSE=1) Scenario 5 (SSE=2)

SVD

Scenario 1 (SSE=3) Scenario 2 (SSE=2) Scenario 3 (SSE=5) Scenario 4 (SSE=1) Scenario 5 (SSE=2)

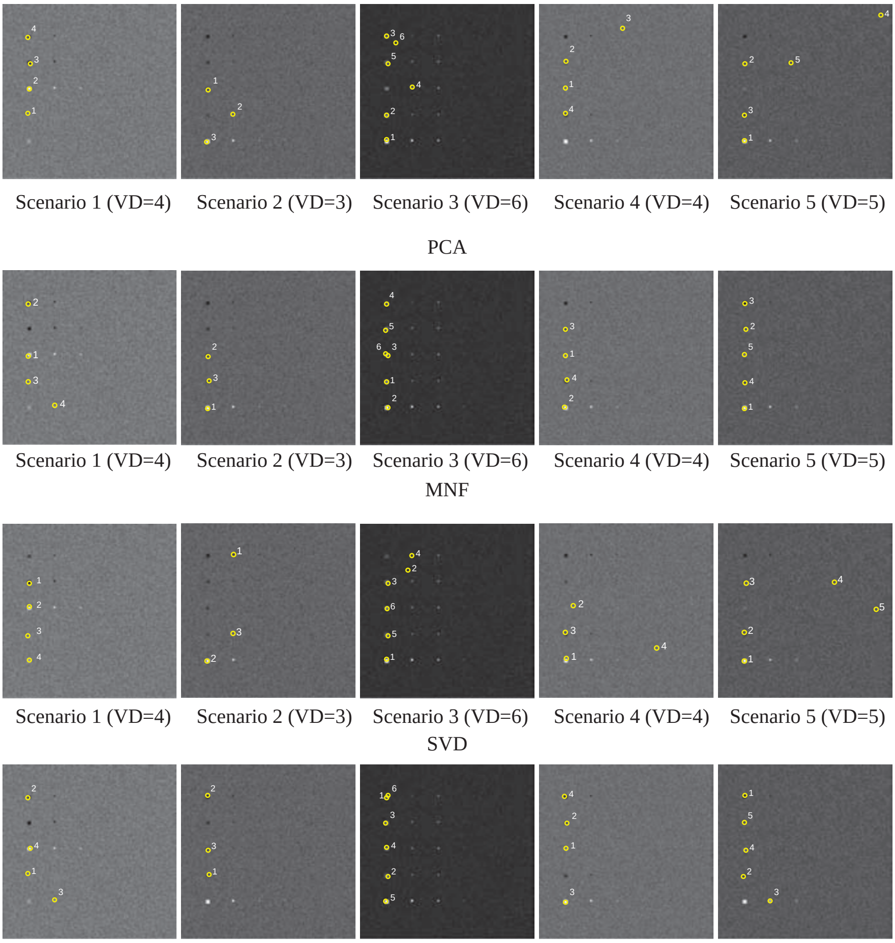
ICA

(a) Values of p estimated by SSE

Figure 11.11 Panel pixels extracted as endmembers by SGA.

According to our extensive experiments, if VCA used twice the value of p as the number of vertices it needed to generate, all the desired p endmembers would be among the $2p$ vertices. However, in this case, not all the vertices extracted by VCA were true endmembers. This was the same drawback also found in PPI that requires as many skewers as possible to cover all potential directions on which endmembers may be projected. Another was its used criterion for endmember extraction which is maximal orthogonal projection not maximal simplex volume used by SGA. Therefore, it could be expected that its performance is not as good as SGA.

5. Based on our experiments, the best among all the four dimensionality reduction techniques, PCA, MNF, ICA, and SVD, was ICA and SVD seemed to be the worst in most of cases. It was also noted that VCA used in this chapter was provided by one of the authors in Nascimento and Dias (2005) who actually used SVD in their VCA. The experiments suggested that VCA using ICA could certainly improve their original version of VCA. There are reasons for it. One is that, in many cases, endmembers can be considered as anomalies that can be only captured by high-order statistics not second-order statistics such as variance,



Scenario 1 (VD=4) Scenario 2 (VD=3) Scenario 3 (VD=6) Scenario 4 (VD=4) Scenario 5 (VD=5)

PCA

Scenario 1 (VD=4) Scenario 2 (VD=3) Scenario 3 (VD=6) Scenario 4 (VD=4) Scenario 5 (VD=5)

MNF

Scenario 1 (VD=4) Scenario 2 (VD=3) Scenario 3 (VD=6) Scenario 4 (VD=4) Scenario 5 (VD=5)

SVD

Scenario 1 (VD=4) Scenario 2 (VD=3) Scenario 3 (VD=6) Scenario 4 (VD=4) Scenario 5 (VD=5)

ICA

(b) Values of p estimated by VD

Figure 11.11 (*Continued*)

signal-to-noise ratio. Another is that the presence of endmembers has low probability that cannot be described by second-order statistics. A third reason is that when endmembers appear, their sample pools are generally small. As a result, the statistics constituted by endmembers can only contribute very little to second-order statistics.

6. The best performance among all the five scenarios was always the one that implemented SGA in conjunction with ICA and the value of p was estimated by VD. This conclusion is also supported by the real image experiments conducted in Chang et al. (2006).

Despite that all the experiments are conducted above based on synthetic images, their studies provide very useful guidelines and reference when it comes to real image experiments, such as which method can estimate the value of p more reliably and accurately, which algorithm is more appropriate for endmember extraction, and which dimensionality reduction technique is more effective in preserving information of endmembers.

11.4 Does an Endmember Set Really Yield Maximum Simplex Volume?

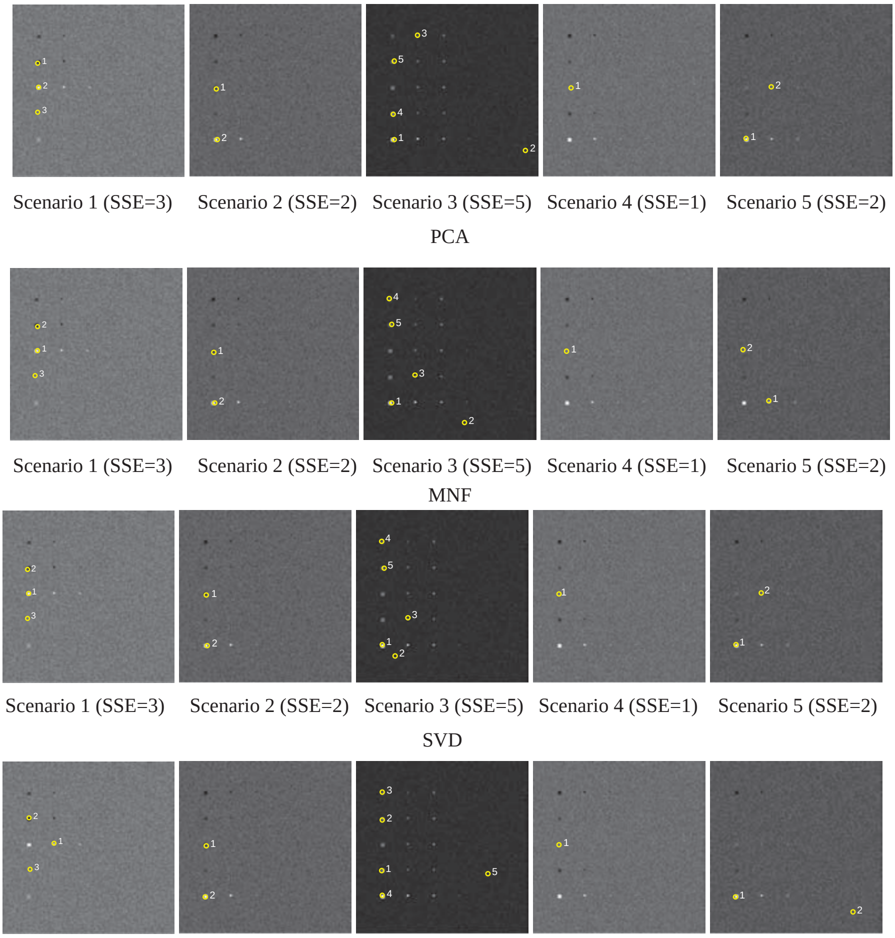
One of commonly used criteria for finding an endmember set is to assume that for a given number of endmembers, *p*, a *p*-vertex simplex with its vertices specified by *p* endmembers always yields the maximal volume. Since there are also other criteria that have been widely used for endmember extraction, an issue of interest is “does an endmember set really produce a simplex with maximal volume?” In other words, using the criterion of maximal simplex volume is a better and more effective measure than other criteria currently being used by EEAs such as OP fully constrained least squares-based spectral unmixing, etc. This section explores this issue by investigating a number of popular EEAs that are designed by different criteria. An extensive experiment-based study is also conducted for comparative analysis.

Endmember extraction has received considerable interest recently. Many EEAs have been also developed for this purpose based on different philosophies, of which three major criteria are of interest. One is convex geometry-based methods that include finding extreme points of convexity via orthogonal projection such as PPI, VCA ATGP. Another is finding a simplex with the minimal volume that embraces *all* data samples such as MVT, CCA, or a simplex with the maximal volume that includes as *many* data samples as possible such as N-FINDR. A third one is least squares error-based constrained spectral unmixing methods such as iterative error analysis (IEA) (Neville et al., 1999) and fully constrained least squares method (FCLS) (Heniz and Chang, 2001). An interesting issue of which criterion is more appropriate for finding true endmembers has been never investigated. According to the endmember definition (Schowengerdt, 1997) an endmember is considered as a pure, idealized signature for a spectral class. So, if there are *p* endmembers are assumed to be present in the data, all data samples linearly mixed by these *p* endmembers should be embraced inside a *p*-vertex simplex with its vertices completely specified by these *p* endmembers. Such a *p* endmember-vertex simplex will produce the maximal volume among all possible *p*-vertex simplexes. In other words, a simplex with all vertices specified by endmembers has its volume greater than or equal to the volume of any simplex with the same number of vertices. So, intuitively, using the maximal volume of a simplex seems to be the most effective measure to determine whether a group of signatures is an endmember set. However, is it true? This section intends to answer this question by investigating other popular criteria and conducting a comparative study via an extensive set of experiments including computer simulations and real data.

Five endmember extraction algorithms (EEAs) are selected and used to conduct experiments to investigate the issue of using maximum simplex volume to find an optimal endemember set. These algorithms are categorized into three classes of EEAs, which are (1) maximal volume-based EEAs, N-FINDR and SGA; (2) OP-based EEAs, PPI, VCA and ATGP; (3) spectral unmixing-based EEA, unsupervised FCLS (UFCLS). Two notes on the selection of these EEAs are worthy being mentioned. In the first class of maximal simplex volume-based EEAs, N-FINDR and SGA are selected. This is because the N-FINDR is designed to find the maximal simplex volume for a given number of endmembers simultaneously, while the SGA is developed to find the maximal simplex volume successively by growing simplexes one at a time starting with two endmembers until it reaches to the given number of endmembers. Interestingly, according to our experiments, it turned out that for a given number of endmembers the maximal simplex volumes produced by the SGA is generally much smaller than that yielded by the N-FINDR, but both extract the same number of endmembers. However, the computational cost saved by the SGA compared to the N-FINDR is not only significant but rather tremendous. This intriguing finding is noteworthy. With very much the same reason for selecting N-FINDR and SGA in the first class of EEAs, three EEAs, PPI, VCA and ATGP are also selected in the second class of OP-based EEAs since they are shown to have very close relationship in Section 11.2.1.

First of all, VD and SSE were used to estimate number of dimensions, q, required for dimensionality reduction. Figures 11.11–11.16 show the endmembers extracted by PPI with 500 skewers, N-FINDR, SGA, VCA, ATGP, and UFCLS. It should be noted that the PPI, N-FINDR, VCA, and SGA required dimensionality reduction (DR) as a preprocessing. Three DR transforms, PCA, MNF, and ICA were used to reduce the original data dimensionality 169 to q with results shown in Figures 11.11(a)–(c)–11.16(a)–(c). Since PPI may generate more than one panel pixel representing the same panel signature, the results shown in Figure 11.11 for the PPI were obtained by those panel pixels that yielded the maximal volumes among all PPI-found pixels for each panel signature.

Table 11.11 also tabulates EEA-extracted pixels corresponding to panel pixels found in Figures 11.11–11.16 where the best results were those obtained by using ICA to perform DR, in which case all five panel signatures could be extracted. If other transforms were used for DR,



Scenario 1 (SSE=3) Scenario 2 (SSE=2) Scenario 3 (SSE=5) Scenario 4 (SSE=1) Scenario 5 (SSE=2)

PCA

Scenario 1 (SSE=3) Scenario 2 (SSE=2) Scenario 3 (SSE=5) Scenario 4 (SSE=1) Scenario 5 (SSE=2)

MNF

Scenario 1 (SSE=3) Scenario 2 (SSE=2) Scenario 3 (SSE=5) Scenario 4 (SSE=1) Scenario 5 (SSE=2)

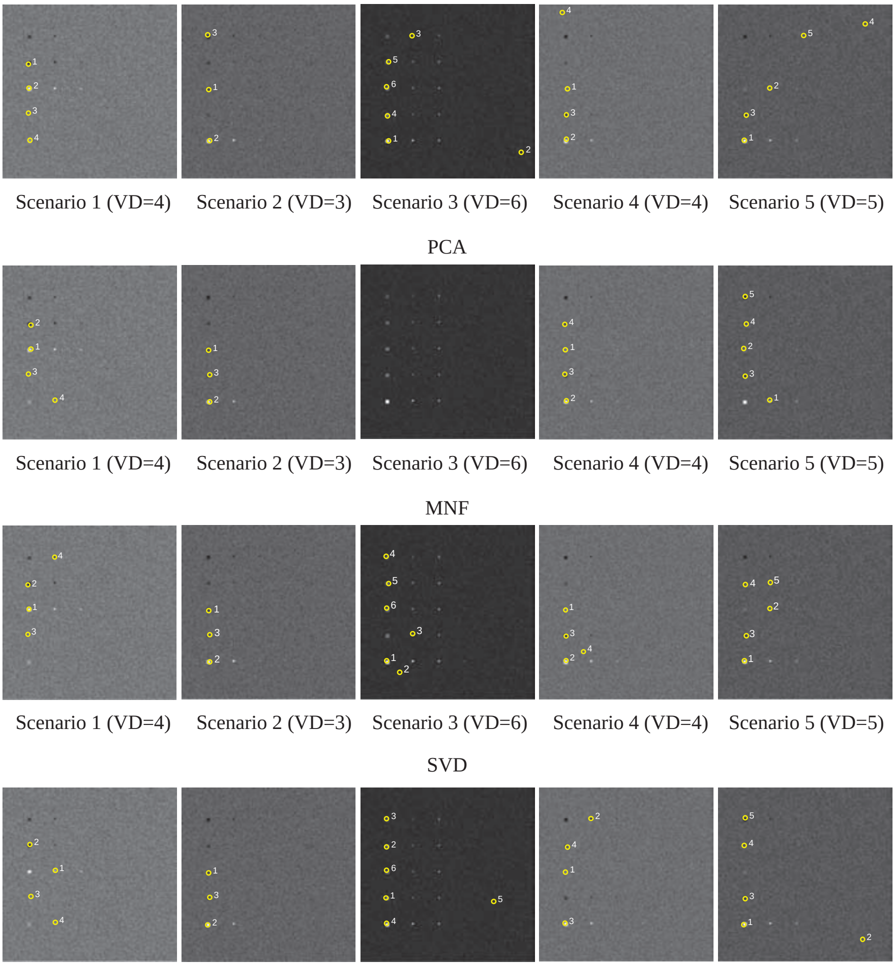
SVD

Scenario 1 (SSE=3) Scenario 2 (SSE=2) Scenario 3 (SSE=5) Scenario 4 (SSE=1) Scenario 5 (SSE=2)

ICA

(a) Values of p estimated by SSE

Figure 11.12 Endmember pixels extracted by PPI using three different DR transforms and 500 skewers.



Scenario 1 (VD=4) Scenario 2 (VD=3) Scenario 3 (VD=6) Scenario 4 (VD=4) Scenario 5 (VD=5)

PCA

Scenario 1 (VD=4) Scenario 2 (VD=3) Scenario 3 (VD=6) Scenario 4 (VD=4) Scenario 5 (VD=5)

MNF

Scenario 1 (VD=4) Scenario 2 (VD=3) Scenario 3 (VD=6) Scenario 4 (VD=4) Scenario 5 (VD=5)

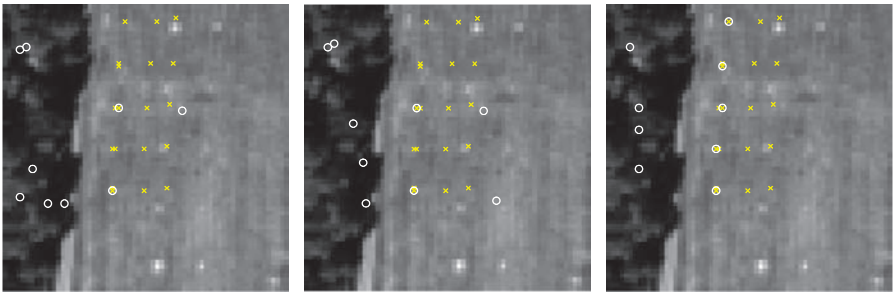
SVD

Scenario 1 (VD=4) Scenario 2 (VD=3) Scenario 3 (VD=6) Scenario 4 (VD=4) Scenario 5 (VD=5)

ICA

(b) Values of *p* estimated by VD

Figure 11.12 (*Continued*)



PCA MNF ICA

Figure 11.13 Endmember pixels extracted by N-FINDR using three different DR transforms.

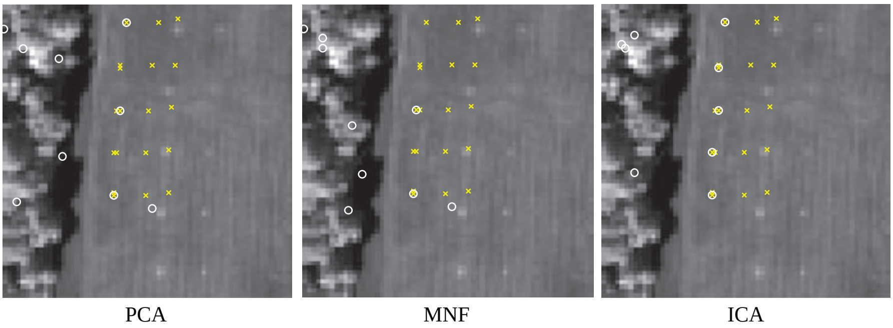


Figure 11.14 Endmember pixels extracted by SGA using three different DR transforms.

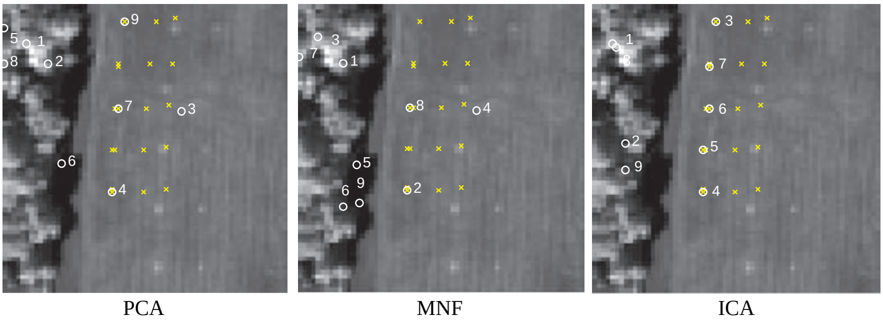


Figure 11.15 Endmember pixels extracted by VCA using three different DR transforms.

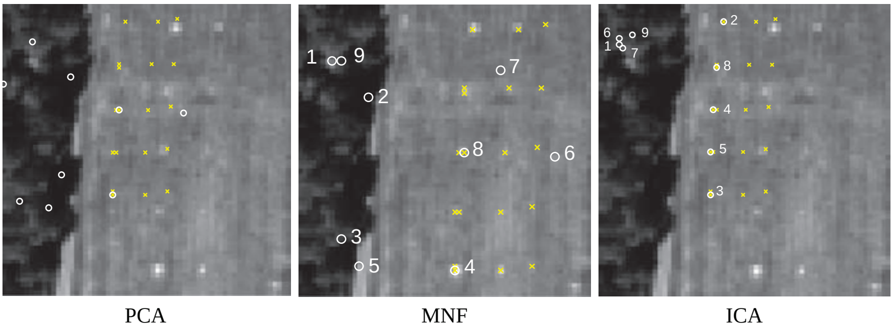


Figure 11.16 Endmember pixels extracted by ATGP and UFCLS.

the best obtained results were only three panel signatures that were extracted by N-FINDR with PCA, SGA with PCA and ATGP.

Table 11.12 calculates the volumes of simplexes formed by extracted pixels for $p = 9$ by six EEAs where the numbers shown in the fourth column are used to rank EEA in descending order of the calculated volumes.

Since N-FINDR is designed to find a maximal volume simplex for a given number of vertices, it is always being ranked "1" shown in Table 11.12 while its performance was also among the best in most cases. On the other hand, UFCLS seemed the one that always produced a simplex with the

Table 11.11 Pixels extracted by six EEAs

		Extracted R panel pixels
PPI	PCA	p_{312}, p_{521}
	MNF	p_{311}, p_{521}
	ICA	$p_{11}, p_{221}, p_{312}, p_{411}, p_{521}$
N-FINDR	PCA	p_{11}, p_{312}, p_{521}
	MNF	p_{311}, p_{521}
	ICA	$p_{11}, p_{221}, p_{312}, p_{411}, p_{521}$
SGA	PCA	p_{11}, p_{312}, p_{521}
	MNF	p_{311}, p_{521}
	ICA	$p_{11}, p_{221}, p_{312}, p_{411}, p_{521}$
VCA	PCA	p_{312}, p_{521}
	MNF	p_{312}, p_{521}
	ICA	$p_{11}, p_{221}, p_{311}, p_{411}, p_{521}$
ATGP		p_{11}, p_{312}, p_{521}
UFCLS		p_{312}, p_{521}

least volume with worst performance. These observations imply that using the criterion of maximal simplex volume was indeed a good and reasonable measure to find endmembers. However, it did not imply that for an EEA to be effective, it must yield maximal volumes. According to Table 11.12, for a given DR transform, there was always at least one EEA that yielded smaller volumes could perform as well as N-FINDR. For example, SGA was always ranked after N-FINDR but performed as well as N-FINDR with much less computational complexity, as demonstrated in Chang et al. (2006). On the other hand, despite the fact that ATGP was one of the two EEAs produced the smallest volumes but it performed as well as or even better than N-FINDR did.

Table 11.12 A comparative analysis on simplex volumes among six EEAs

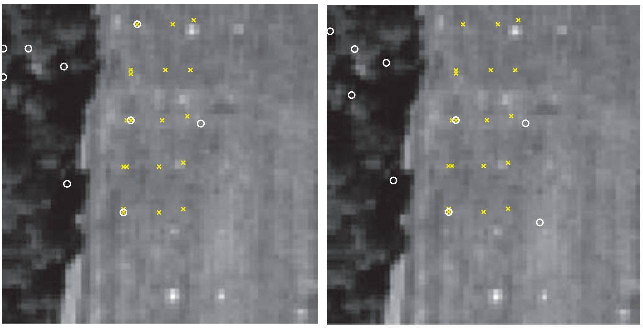
DR	EEA	Volumes	Rank	Number of extracted R panel pixels
PCA	PPI	3.3202×10^{23}	2	2
	N-FINDR	4.3536×10^{23}	1	3
	SGA	1.7306×10^{23}	3	3
	VCA	1.0978×10^{23}	4	2
	ATGP	11.7049×10^{22}	5	3
	UFCLS	3.1084×10^{22}	6	2
MNF	PPI	6.0338×10^{17}	2	2
	N-FINDR	6.8626×10^{17}	1	2
	SGA	4.1013×10^{17}	3	2
	VCA	6.5567×10^{16}	4	2
	ATGP	7.7128×10^{15}	6	3
	UFCLS	6.2375×10^{16}	5	2
ICA	PPI	2.9075×10^{6}	3	5
	N-FINDR	3.7913×10^{7}	1	5
	SGA	2.0113×10^{7}	2	5
	VCA	7.8153×10^{4}	4	5
	ATGP	39.5323	5	3
	UFCLS	0.6267	6	2

Two comments are noteworthy. Although ATGP and UFCLS did not require a DR transform to perform endmember extraction, they did use the same DR transforms performed by other EEAs to calculate the volumes of their found simplexes for a fair comparative analysis. Additionally, according to Table 11.12, the simplexes formed by extracted endmembers using PCA or MNF always produced larger simplex volumes compared to those obtained by ICA, which usually yielded the least simplex volumes. However, the performance in endmember extraction using ICA has been shown to be better than those using PCA or MNF in all cases. This fact demonstrates that for a given EEA, the one used for DR was not always the one that produced simplexes maximal volumes. However, as noted above, for a given DR transform, an EEA that produced simplexes with maximal volumes was always one, but probably not only one that performed the best.

In summary, for a given DR transform, an EEA that produced a maximal simplex volume was always a desired one, but not necessarily the only one. On the other hand, for a given EEA, a DR transform that produced a maximal simplex volume was not always the one that performed the best in endmember extraction. These conclusions imply that the DR transform used by an EEA plays a key role in its performance, while using maximum simplex volume as a criterion to extract endmembers seems always a good measure for an EEA.

11.5 Impact of Dimensionality Reduction on EEAs

Finally, this section investigates the impact of DR on the performance of EEAs. Most endmember extraction algorithms require dimensionality reduction to reduce computation complexity. For example, in order to calculate simplex volumes N-FINDR, SGA, and VCA need DR to reduce data dimensionality of a simplex or convex hull to avoid singularity problems in which case the number of components, q, to be retained after DR is set to the number of extracted endmembers, p. However, several questions may arise: does "$p = q$" always give the best performance? If more components are used to extract the same number of endmembers, that is, $q > p$, will it improve the performance of EEAs? The 15-panel HYDICE data in Figure 1.15(a) and (b) provide an excellent example to explore insights into these issues. The EEAs to be tested for performance evaluation are N-FINDR, SC N-FINDR, and SGA, all of which require simplex volume calculation that is closely related to data dimensionality reduction. Furthermore, because the inconsistency of these algorithms caused by the random initial endmembers might result in biased comparisons, ATGP is used as an EIA to generate the same initial endmembers shown in Figure 11.17 to initialize N-FINDR and SC N-FINDR, while IED-SGA is used to run SGA in the following experiments.



(a) ATGP

(b) UFCLS

Figure 11.17 Nine endmembers generated by ATGP used to initialize the N-FINDR and SC N-FINDR in the following experiments.

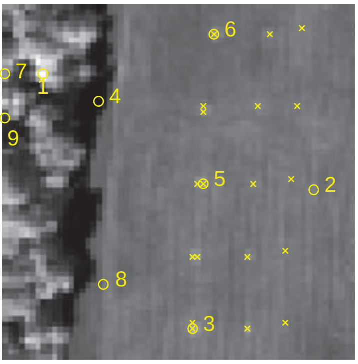
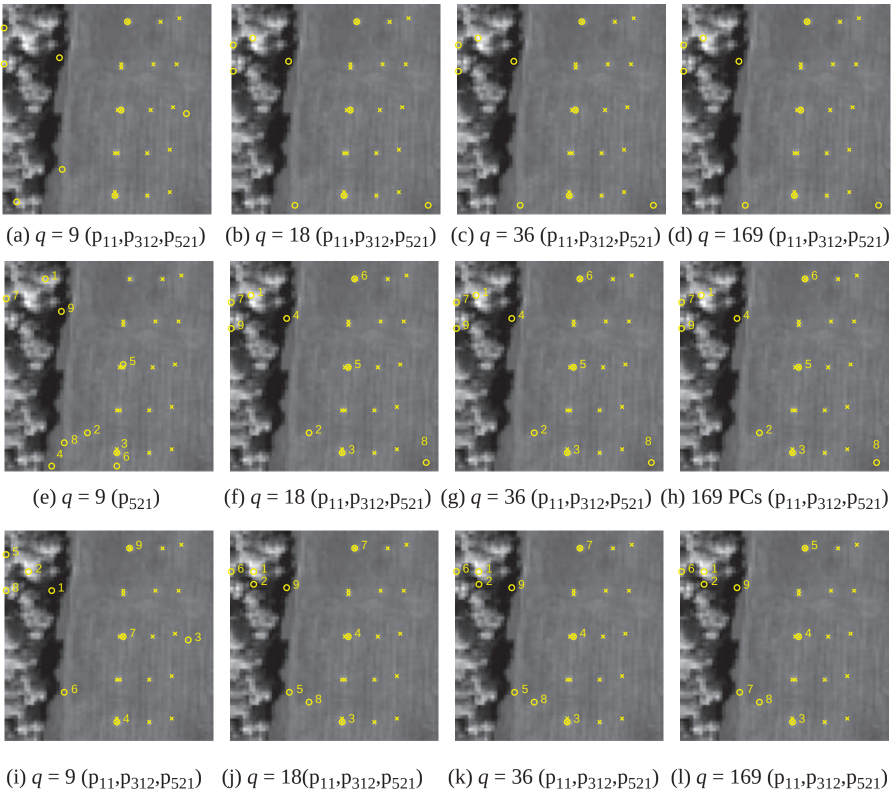


Figure 11.18 Nine endmembers extracted by (a)–(d) N-FINDR; (e)–(h) SC N-FINDR; (i)–(l) SGA on several numbers of components generated by the PCA-DR transform.

Since VD estimated for this HYDICE data was $p = 9$ with $P_F \leq 10^{-3}$, q was set to $q = p$, $2p$, $4p$ and $q =$ full number of bands, 169. Figures 11.18–11.21 show nine endmember extraction results of using q components resulting from four different DR transforms, PCA, MNF, SVD, and ICA, respectively, where extracted panel pixels that correspond to endmembers are listed in parenthesis



(a) $q = 9$ (p_{11}, p_{312}, p_{521}) (b) $q = 18$ (p_{11}, p_{312}, p_{521}) (c) $q = 36$ (p_{11}, p_{312}, p_{521}) (d) $q = 169$ (p_{11}, p_{312}, p_{521})

(e) $q = 9$ (p_{521}) (f) $q = 18$ (p_{11}, p_{312}, p_{521}) (g) $q = 36$ (p_{11}, p_{312}, p_{521}) (h) 169 PCs (p_{11}, p_{312}, p_{521})

(i) $q = 9$ (p_{11}, p_{312}, p_{521}) (j) $q = 18$(p_{11}, p_{312}, p_{521}) (k) $q = 36$ (p_{11}, p_{312}, p_{521}) (l) $q = 169$ (p_{11}, p_{312}, p_{521})

Figure 11.19 Nine endmembers extracted by (a)–(d) N-FINDR; (e)–(h) SC N-FINDR; (i)–(l) SGA on different numbers of components generated by the MNF-DR transform.

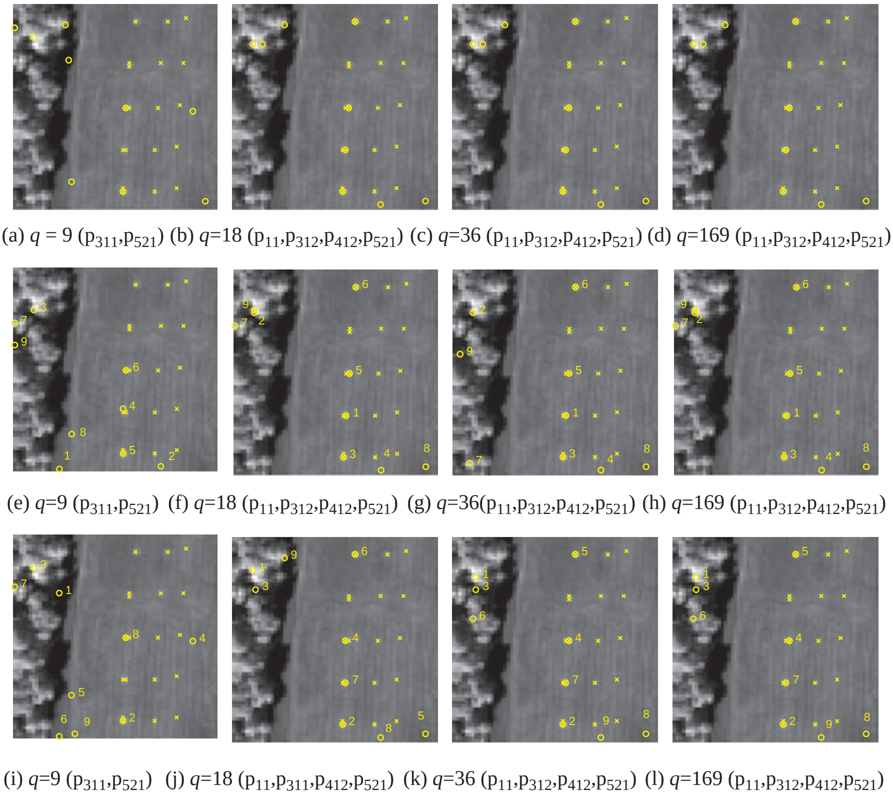
underneath each figure. As shown in Figure 11.18, using $q = 18$ components allows SC N-FINDR to extract two more panel pixels, but it did not improve results after $q > 18$. On the other hand, N-FINDR and SGA did not improve their performance $q > 9$.

Interestingly, the results in Figure 11.19 show that when MNF was used to perform DR all the three EEAs could only extract two panel pixels when $q = 9$ but their performance was significantly improved to extract four panel pixels when $q \geq 18$.

Figure 11.20 shows nine endmembers extracted by the three EEAs using SVD to reduce data dimensionality from 169 to various values of q. The results were similar to those in Figure 11.18 with only difference in the case of $q = 9$.

Comparing results in Figures 11.18–11.20, it is very clear that when DR was required, MNF outperformed PCA and SVD. However, when ICA was used to perform DR, Figure 11.21 clearly shows that $q = 9$ was sufficiently enough for all the three EEAs to be able extract five panel pixels, p_{11}, p_{221}, p_{312}, p_{411}, and p_{521}, corresponding to five endmembers except only one case that SC N-FINDR using nine ICs extracts p_{311} instead of p_{312}.

Comparing results in Figure 11.21 to results in Figures 11.18–11.20, ICA-DR provided the best results for the HYDICE data. So, it is expected that if EEAs performs on the sphered image cube, which uses similar concept as ICA does to retain high-order statistics, the results should be better than the original data without being sphered. Figure 11.22 shows the results by using the full bands

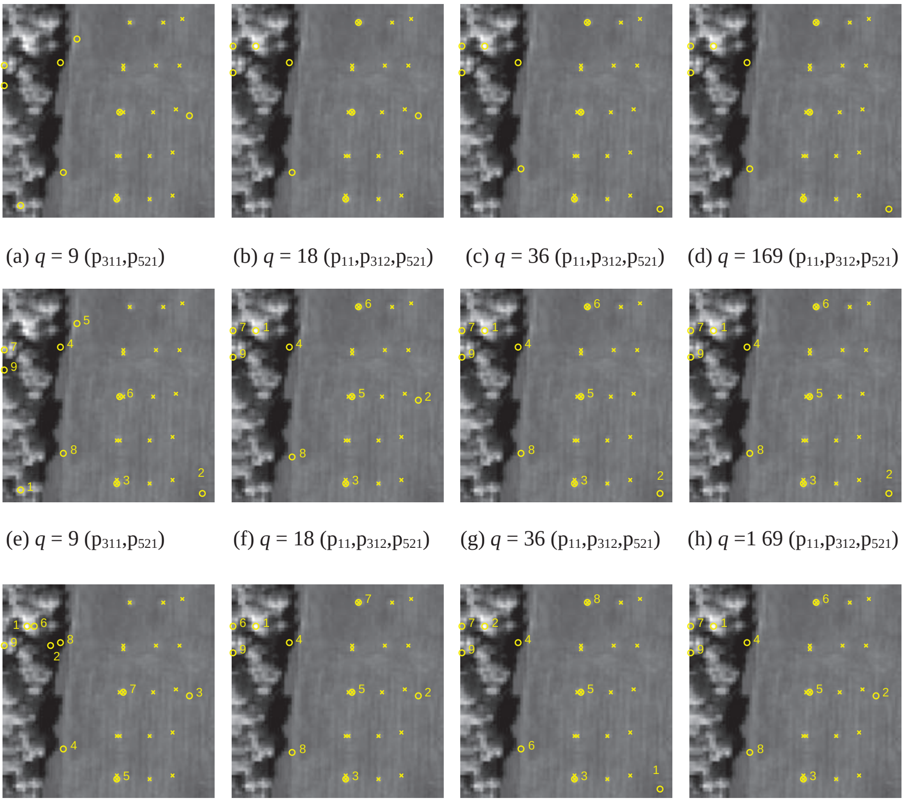


(a) q = 9 $(\mathrm{p}_{311},\mathrm{p}_{521})$ (b) q=18 $(\mathrm{p}_{11},\mathrm{p}_{312},\mathrm{p}_{412},\mathrm{p}_{521})$ (c) q=36 $(\mathrm{p}_{11},\mathrm{p}_{312},\mathrm{p}_{412},\mathrm{p}_{521})$ (d) q=169 $(\mathrm{p}_{11},\mathrm{p}_{312},\mathrm{p}_{412},\mathrm{p}_{521})$

(e) q=9 $(\mathrm{p}_{311},\mathrm{p}_{521})$ (f) q=18 $(\mathrm{p}_{11},\mathrm{p}_{312},\mathrm{p}_{412},\mathrm{p}_{521})$ (g) q=36$(\mathrm{p}_{11},\mathrm{p}_{312},\mathrm{p}_{412},\mathrm{p}_{521})$ (h) q=169 $(\mathrm{p}_{11},\mathrm{p}_{312},\mathrm{p}_{412},\mathrm{p}_{521})$

(i) q=9 $(\mathrm{p}_{311},\mathrm{p}_{521})$ (j) q=18 $(\mathrm{p}_{11},\mathrm{p}_{311},\mathrm{p}_{412},\mathrm{p}_{521})$ (k) q=36 $(\mathrm{p}_{11},\mathrm{p}_{312},\mathrm{p}_{412},\mathrm{p}_{521})$ (l) q=169 $(\mathrm{p}_{11},\mathrm{p}_{312},\mathrm{p}_{412},\mathrm{p}_{521})$

Figure 11.20 Nine endmembers extracted by (a)–(d) N-FINDR; (e)–(h) SC N-FINDR; (i)–(l) SGA on different numbers of components generated by the SVD-DR transform.



(a) q = 9 (p_{311},p_{521}) (b) q = 18 (p_{11},p_{312},p_{521}) (c) q = 36 (p_{11},p_{312},p_{521}) (d) q = 169 (p_{11},p_{312},p_{521})

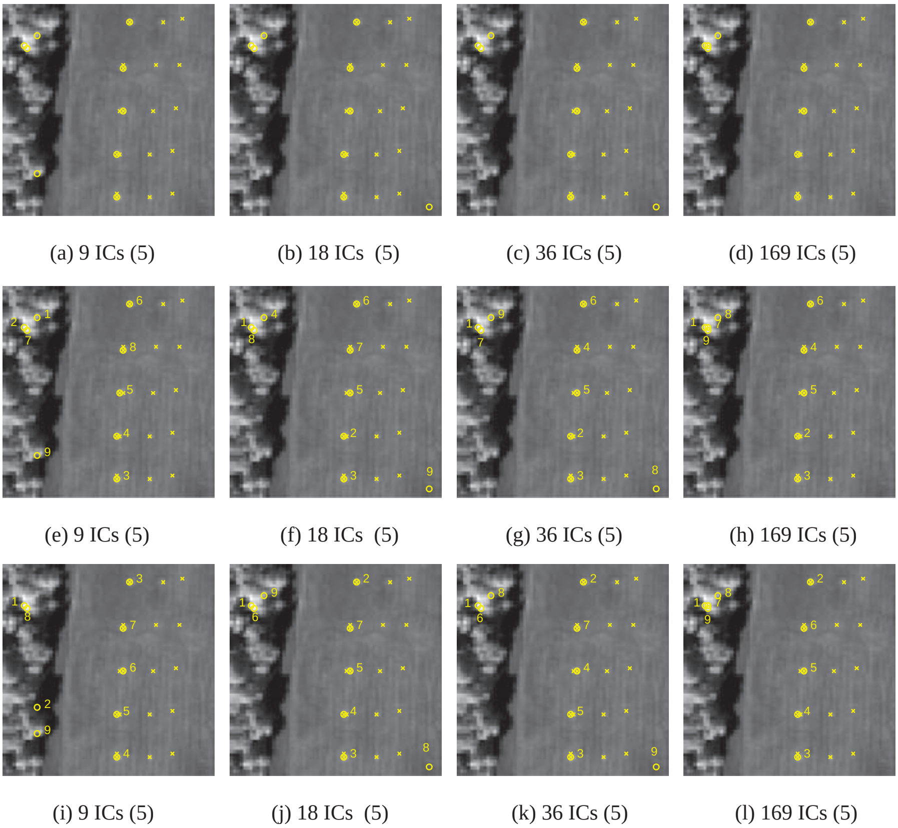
(e) q = 9 (p_{311},p_{521}) (f) q = 18 (p_{11},p_{312},p_{521}) (g) q = 36 (p_{11},p_{312},p_{521}) (h) q =1 69 (p_{11},p_{312},p_{521})

(i) q = 9 (p_{312},p_{521}) (j) q = 18 (p_{11},p_{312},p_{521}) (k) q = 36 (p_{11},p_{312},p_{521}) (l) q = 169 (p_{11},p_{312},p_{521})

Figure 11.21 Nine endmembers extracted by (a)–(d) N-FINDR; (e)–(h) SC N-FINDR; (i)–(l) SGA on different numbers of components generated by the ICA-DR transform.

of the original image cube and sphered image cube where the effect of using sphered data was the same as that of using q independent components in Figure 11.21 in which five panel pixels can be also extracted to be identified as five endmembers.

As a concluding remark on the above-mentioned experiments, an interesting finding from comparing results in Figures 11.18–11.22 is worth mentioning. It seems a common sense that EEAs using original data should perform better than using reduced data since the information in the reduced data is either being lost, such as band selection, or compressed, such as DR. Apparently, this may not be true if we compare the results obtained by using full bands in Figures 11.18–11.20 to the results in Figure 11.21; all the three EEAs performed significantly better using q ICA-DR components than using the original data with 169 bands. This is because DR basically performs data compaction into a low dimensional space while still preserving certain crucial endmember information. More interestingly, according to Figure 11.22 EEAs using sphered data performed better than using the original data. These experiments provide evidence that endmember information can be better characterized by high-order statistics because of its rarity and purity and cannot be effectively captured and preserved by the second-order statistics-based DR transforms such as PCA, MNF, and SVD in fewer DR components. Moreover, Figure 11.22 also suggests that second-order data statistics may also obscure and overwhelm endmember information that can be only



(a) 9 ICs (5) (b) 18 ICs (5) (c) 36 ICs (5) (d) 169 ICs (5)

(e) 9 ICs (5) (f) 18 ICs (5) (g) 36 ICs (5) (h) 169 ICs (5)

(i) 9 ICs (5) (j) 18 ICs (5) (k) 36 ICs (5) (l) 169 ICs (5)

Figure 11.22 Nine endmembers extracted by N-FINDR in (a)–(d), SC N-FINDR (e)–(h), and SGA (i)–(l) from the sphered image cube.

revealed after data being sphered by removing the first two data statistics. This explains why removing first- and second-order statistics in sphered data can enhance subtle information provided by endmembers.

11.6 Conclusions

This chapter is a culmination of previous chapters in PART II and concludes endmember extraction with exploration of insights into relationships among several recently developed popular EEAs, PPI, N-FINDR, VCA, SGA, and ATGP. With appropriate interpretations VCA and SGA can be considered sequential versions of PPI and N-FINDR, respectively, with ATGP bridging the gap between PPI and VCA. On the other hand, the relationship between VCA and SGA is derived from the same idea of growing convex hulls for VCA and growing simplexes for SGA by adding a new endmember at a time as a new vertex in sequence, with the only difference that VCA follows PPI to use maximal orthogonal projection as a criterion as opposed to SGA, which follows N-FINDR to use maximal simplex volume as a criterion. This difference leads to an interesting issue: What is the best criterion to design EEAs? As investigated in this chapter it turns out that using the maximal simplex volume as a criterion is a better measure to design EEAs due to the fact that simplex volume satisfies full abundance constraints, abundance sum-to-one constraint (ASC) and abundance nonnegativity constraint (ANC), while convex hull volume only satisfies ANC. Details on this will be found in Section 33.2 and Chang (2013). Since most EEAs require dimensionality reduction (DR) to reduce data volumes, the impact of DR on endmember extraction is also investigated in this chapter. The experimental results demonstrate that high-order statistics-based DR transforms such as independent component analysis (ICA) or data sphering can significantly improve performance over second-order statistics-based DR transforms such as PCA and MNF.

III

Supervised Linear Hyperspectral Mixture Analysis

Linear spectral mixture analysis (LSMA) is a theory developed for linear spectral unmixing (LSU). It assumes that data sample vectors can be represented by linear mixtures of a finite number of basic component constituent spectra, known as endmembers. More specifically, let $\{\mathbf{m}_j\}_{j=1}^p$ be p such basic component constituent spectra and $\mathbf{r}$ be the spectral signature of a data sample vector. The LSMA models $\mathbf{r}$ as $\mathbf{r} = \sum_{j=1}^p \alpha_j \mathbf{m}_j + \mathbf{n}$ with α_j being the abundance fraction of $\mathbf{m}_j$ resident in the data sample vector $\mathbf{r}$ where the term $\mathbf{n}$ is included to account for a model error or a noise factor. So, according to LSMA, there are two sets of parameters, $\{\mathbf{m}_j\}_{j=1}^p$ and $\{\alpha_j\}_{j=1}^p$ needed to be solved and LSU is developed as a technique to find $\{\alpha_j\}_{j=1}^p$ associated with $\{\mathbf{m}_j\}_{j=1}^p$. As a consequence, in order to carry out LSU effectively, three-stage processes must be performed in sequence. The first-stage process is to estimate the parameter p. As discussed in Chapter 5, this can be accomplished by VD. This is followed by a second-stage process to find the p endmembers $\{\mathbf{m}_j\}_{j=1}^p$. Finally, a third-stage process is to perform LSU after $\{\mathbf{m}_j\}_{j=1}^p$ are known. Techniques developed for LSU have been reported extensively in the literature, for example, Chang (2003a). So, in general, there are two approaches to implementing LSMA to be discussed in PART III and PART IV. One is to assume that $\{\mathbf{m}_j\}_{j=1}^p$ is known *a priori* or provided by ground truth in which case, there is no need of implementing the first two-stage processes. Such LSMA is referred to as supervised LSMA (SLSMA). The main focus of PART III is devoted to this topic. The other is to assume that there is no prior knowledge about the data in which case all the three stage processes discussed above are required. Such LSMA is referred to as unsupervised LSMA (ULSMA). Since ULSMA is more involved and presents great challenges in algorithm design, it will be discussed in great detail in PART IV, specifically, Chapter 17.

The SLSMA in PART III assumes that the prior knowledge of $\{\mathbf{m}_j\}_{j=1}^p$is given. So, LSU implemented by SLSMA is nothing more than unmixing the spectral signature $\mathbf{r}$ of a data sample vector into a set of abundance fractions, $\{\alpha_j(\mathbf{r})\}_{j=1}^p$ represented by a fractional abundance vector

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

$\boldsymbol{\alpha}(\mathbf{r}) = (\alpha_1(\mathbf{r}), \alpha_2(\mathbf{r}), \ldots, \alpha_p(\mathbf{r}))^T$ where each component of $\boldsymbol{\alpha}(\mathbf{r})$, $\alpha_j(\mathbf{r})$ represents an appropriate abundance fraction associated with the jth endmember $\mathbf{m}_j$. In other words, by means of LSU the $\mathbf{r}$ can be represented by a p-dimensional unmixed abundance fractional vector $\boldsymbol{\alpha}(\mathbf{r})$. Consequently, the entire data set is further represented by a set of p fractional abundance maps, each of which is a abundance fraction map of one particular endmember. Unlike a hard decisions-made classifier which performs class membership labeling to produce classification maps the LSU-produced abundance fraction maps are real-valued and can be considered as soft decisions yielded by an abundance estimator.

In Chang (2003a) four techniques are developed for LSMA to perform LSU: the signal-to-noise ratio (SNR)-derived orthogonal subspace projection (OSP) (Harsanyi and Chang, 1994) and three least squares (LS)-based techniques, least squares orthogonal subspace projection (LSOSP) (Tu et al., 1997), non-negativity constrained least squares (NCLS) (Chang and Heinz, 2000), and fully constrained least squares (FCLS) (Heinz and Chang, 2001). Since it has been shown in Chang (1998, 2003a) that OSP can be also derived by the least squares error (LSE) criterion, it can be also considered as an unconstrained least squares-based LSU technique. Consequently, SLSMA using OSP, LSOSP, NCLS, and FCLS is referred to as LS-SLSMA, which will further be investigated in several different aspects in four chapters, Chapters 12–15.

Despite the fact that OSP has been discussed in great detail in Chapter 3 in Chang (2003a), many different insights into OSP are yet to be explored. Chapter 12 revisits and re-derives OSP from three signal processing perspectives in context of *a priori* and *a posteriori* information. It shows that when complete prior knowledge is provided, LS-SLSMA and Gaussian maximum likelihood classification are indeed equivalent to OSP in the sense of their functionality. Moreover, another widely used technique, constrained energy minimization (CEM) developed by Harsanyi in his dissertation (1993) and a commonly used anomaly detection algorithm, RX detector (RXD) developed by Reed and Yu (1990), can be also interpreted as variants of OSP using partial and no prior knowledge, respectively. Interestingly, an alterative interpretation of OSP from an information process point of view can be also found in Chang (2007c, i.e., Chapter 3 in Chang (2007a) and Section 33.3.1.

While OSP is derived using SNR as a criterion for optimality, Settle (1996) and Chang (1998) also showed that OSP and the Gaussian maximum likelihood classifier (GMLC) were essentially the same and both operated the same function forms subject to a constant that actually accounts for unmixed error. In other words, OSP can be further re-derived by LSE as Least Squares OSP (LSOSP) that was shown to be identical to GMLC by Tu et al. (1997) along with its various forms also derived in Chang et al. (1998) and Chang (2003a). With this interpretation LSE was further used to derive least squares abundance constrained LSU methods such as partially abundance-constrained method, called non-negativity constrained least squares (NCLS) derived by Chang and Heinz (2000), and fully abundance-constrained method, called fully constrained least squares (FCLS) developed by Heinz and Chang (2001).

Although SNR and LSE are shown to derive the same operator in the sense of LSU with SNR for detection and LSE for estimation, they are apparently not designed for classification. So, from a view point of pattern classification, the four LS-SLSMA techniques, OSP, LSOSP, NCLS, and FCLS, may not be best in terms of classification. It is known that one of best pattern classification techniques is Fisher's linear discriminant analysis (FLDA) designed on Fisher's ratio. Chapter 13 extends FLDA to an LSU technique, called Fisher's LSMA (FLSMA). One earlier attempt was made by Du and Chang (2000) where an extended version of Fisher's linear discriminant analysis (FLDA), called linearly constrained discriminant analysis (LCDA), was used to perform mixed pixel classification. However, the criterion used in LCDA is still a distance-based measure, not Fisher's ratio. So, technically

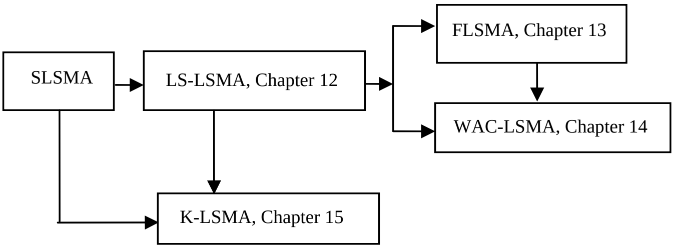


Figure III.1 Relationships among LS-LSMA, FLSMA, WAC-LSMA, and K-LSMA

speaking, LCDA is not a Fisher ratio-based mixed pixel classification technique. Nevertheless, it can be further shown in Chapter 13 that it is indeed an alternative form of FLSMA.

In a comparison between LS-LSMA and FLSMA there is a key difference in terms of how LSE is weighted in abundance fraction estimation. In order to take care of classification error FLSMA introduced a within-class scatter matrix obtained from Fisher's ratio to account for errors resulting from unmixing, whereas LS-LSMA weighs LSE on all abundance fractions equally likely without taking into account difficulty levels of unmixing different endmembers. Chapter 14 generalizes LS-LSMA and FLSMA to weighted abundance-constrained LSMA (WAC-LSMA) by including a weighting matrix $\mathbf{A}$ into LSE that includes LS-LSMA and FLSMA as its special cases. For example, when $\mathbf{A} =$ the identity matrix, WAC-LSMMA is simplified to LS-LSMA. On the other hand, when $\mathbf{A} =$ within-class scatter matrix, WAC-LSMA is reduced to FLSMA.

Since the linear mixing model used by LSMA may not be effective in solving linear nonseparable problems, a feasible solution is to introduce a nonlinear kernel into LSU in a similar manner that a kernel-based support vector machine (SVM) is derived in Section 2.3.1.2.1 in Chapter 2. To accomplish this goal Chapter 15 extends four LS-SLSMA techniques, OSP, LSOSP, NCLS, and FCLS, to their kernel counterparts, KOSP, KLSOSP, KNCKLS, and KFCLS. Figure III.1 depicts the relationships among all the four versions of SLSMA.

12

Orthogonal Subspace Projection Revisited

The orthogonal subspace projection (OSP) approach has received considerable interests in hyperspectral image classification since it was first developed in 1994 (Harsanyi and Chang, 1994). It has been shown to be a versatile technique for a wide range of applications in subpixel detection (Chang, 2003a), mixed classification (Chang, 2003a), dimensionality reduction in Chapter 6, virtual dimensionality (VD) estimation in Chapter 5, and variable-number variable-band selection (VNVBS) in Chapter 27. Unfortunately, insights into its design rationale have not been explored in the past. In this chapter, we revisit this technique and study OSP from several signal processing perspectives. In particular, we further conduct an in-depth investigation in an issue of how to effectively operate OSP using different levels of *a priori* target knowledge for target detection and classification. Additionally, we also look into various assumptions made on OSP, which result in filters with different forms, some of which turn out to be well-known and popular target detectors and classifiers. Interestingly, we also show how OSP is related to the commonly used least-squares-based linear spectral mixture analysis (LSMA) and how OSP takes advantage of Gaussian noise to arrive at the Gaussian maximum likelihood detector/estimator and likelihood ratio test. Extensive experiments are also conducted to simulate scenarios to illustrate the utility of OSP operating under various assumptions and different degrees of target knowledge.

12.1 Introduction

Hyperspectral imagery provides additional benefits over multispectral imagery in many applications, such as detection, discrimination, classification, quantification, identification, etc. In early days, hyperspectral imagery has been processed and analyzed by multispectral image processing algorithms via preprocessing such as feature extraction, dimensionality reduction, and band selection. Such multispectral-to-hyperspectral approaches have achieved some success and may have led to a brief that hyperspectral imaging is nothing more than a straightforward extension of multispectral image processing. As we will see, this is apparently not the case. When the spectral resolution is low as multispectral images are, the used image processing techniques are generally developed to explore spatial information such as geographical information system (GIS) (Jensen, 1996) for spatial domain analysis. Therefore, as spectral resolution is increased significantly like hyperspectral imagery, such spatial domain-based multispectral imaging techniques

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

may be found to be less effective in certain applications. In particular, if targets of interest only account for a small population with very limited spatial extent, the techniques based on spatial information can easily break down. In some cases where the target size may be even smaller than the pixel resolution, for example, rare minerals in geology, special species in agriculture and ecology, small vehicles in battlefields, etc., the data analysis must rely on spectral information provided by a single pixel. Under certain circumstances, the analysis can be only performed at the subpixel level. In order to address this problem, spectral unmixing has been developed to exploit pixel-level spectral information for image analysis. Its success in both multispectral and hyperspectral image analyses has been demonstrated in many applications (Chang, 2003a).

In order to further facilitate spectral unmixing applications in hyperspectral imagery, Harsanyi and Chang developed a hyperspectral image classification technique, referred to as OSP from a viewpoint of hyperspectral imagery (Harsanyi and Chang, 1994). Their idea is based on two aspects: (1) how to best utilize the target knowledge provided *a priori* and (2) how to effectively make use of hundreds of available contiguous spectral bands. With regard to aspect (1), the prior target knowledge is characterized in accordance with target signatures of interest, referred to as the desired target signature, **d**, and undesired target signature matrix **U** formed by those target signatures that are not wanted in image analysis. We believe that OSP is the first approach proposed to separate **d** from the **U** in a signal detection model, and then eliminate the undesired target signatures in **U** prior to detection of **d** so as to improve signal detectability. As for aspect (2), the issue of how to effectively use available spectral bands can be best explained by the well-known pigeon-hole principle in discrete mathematics (Epp, 1995) as discussed in Section 1.3.2. The following example may help readers understand the concept behind OSP.

Suppose that there are 13 pigeons flying into a dozen of pigeon holes (nests). The pigeon-hole principle says that there must exist at least one pigeon hole that should accommodate at least two or more pigeons. Now, if we interpret target signatures of interest and the number of spectral bands as the pigeons and the number of pigeon holes, respectively, then we can use one spectral band to accommodate a distinct target signature for separation. In order to make sure that no more than one target signature is accommodated in a single spectral band, a spectral band that has been used to accommodate a target signature must be disposed. In doing so, the principle of orthogonality is introduced as a mechanism to separate one spectral band from another so that target signatures accommodated in two separate spectral bands are orthogonal to each other. In this case, one band will not share target information with another band. However, for this approach to be effective, the number of spectral bands must be no less than the number of target signatures of interest. For hyperspectral imagery this requirement seems to be met automatically and the pigeon-hole principle is always valid. Unfortunately, using spectral dimensionality as a means to perform target detection, classification and identification is generally not applicable to multispectral imagery, which usually has fewer spectral bands than the number of target signatures of interest. For instance, a SPOT image data has three spectral bands that can be used for data analysis. If more than three target signatures need to be analyzed, the idea of using spectral bands for target detection and classification may not work effectively (Chang and Brumbley, 1999). To circumvent this difficulty, Ren and Chang developed a generalized OSP that included a dimensionality expansion technique to expand the number of spectral bands nonlinearly for OSP to have sufficient spectral dimensions to carry out orthogonal projection (Ren and Chang, 2000). Its utility was further extended to magnetic resonance (MR) image classification (Wang et al., 2001, 2002; Wang, 2002), Wong (2010) and Chapter 32 in this book.

Many OSP-based algorithms have been developed for various applications (Chang, 2003a) since OSP was introduced in 1994 (Harsanyi and Chang, 1994) and its potential in hyperspectral data exploitation is yet to be explored. For example, the noise assumption is not necessarily

Gaussian as commonly assumed. If the noise is assumed to be Gaussian, it has been shown (Settle, 1996; Chang, 1998; Chang et al., 1998) that Harsanyi and Chang's OSP classifier performed essentially like the Gaussian maximum likelihood estimator (Settle, 1996). Nevertheless, from a technical point of view, the design concepts of these two techniques are different. Harsanyi and Chang's OSP classifier is derived from the signal-to-noise ratio (SNR) using a signal detection approach compared to the Gaussian maximum likelihood estimator that is a parametric estimation-based approach. So, technically speaking, Harsanyi and Chang's OSP classifier is a soft decision-made detector to be used to perform classification, more specifically, unmixing. Interestingly, Harsanyi and Chang's OSP can be further shown to perform as a least-squares estimator by including a scaling constant to account for LS estimation error, which is identical to the least squares solution derived from least squares OSP (LSOSP) (Tu et al., 1997; Chang, 1998; Chang et al., 1998).

When OSP was first developed, it required the full knowledge of endmembers to form a linear mixing model to be used to unmix data sample vectors. Such complete *a priori* information may be difficult to obtain in reality, if not impossible. Two approaches have been developed to mitigate this dilemma. One is to develop unsupervised algorithms to obtain the necessary endmember information directly from the data to be processed (Chang, 2003a: Chapter 5). This type of information is referred to as *a posteriori* information as opposed to *a priori* information provided in advance prior to data processing. Since the accuracy of the *a posteriori* information is closely related to the unsupervised method to be used to generate the information, it may not be always reliable. To avoid this problem, a second approach is to suppress unknown information without actually knowing it. One way to do so is the constrained energy minimization (CEM) developed by Harsanyi in his dissertation (Harsanyi, 1993), which only needs the knowledge of the desired signal source. Other than this desired signal source, no knowledge is required. This approach is particularly useful and attractive in the case that the image background is not unknown or very difficult to characterize. CEM was later extended to the target-constrained interference-minimized filter (TCIMF) (Ren and Chang, 2001), which characterized signal sources into three separate information sources, desired, undesired, and interference. Using this three-source model, TCIMF could detect multiple desired signal sources, annihilate undesired signal sources, while suppressing interference caused by unknown signal sources at the same time. Comparing to OSP that only deals with desired and undesired signal sources and CEM that only considers the desired signal source without taking into account other signal sources, TCIMF combines both OSP and CEM into one filter operation and includes them as its special cases, respectively. Interestingly, as will be shown, CEM and TCIMF can be viewed as various versions of OSP operating different degrees of target knowledge. In other words, OSP can be considered as a spectral correlation-whitened version of TCIMF, while TCIMF can be thought of as OSP-version of CEM that eliminates rather than suppresses the undesired signatures. Specifically, when the sample spectral correlation matrix in TCIMF is whitened (i.e., de-correlated), TCIMF performs as if it was OSP. On the other hand, when CEM operates in the same way that OSP eliminates the undesired target signatures, CEM becomes TCIMF. In either case, both CEM and TCIMF are derived from OSP and can be regarded as variants of OSP based on the knowledge used in their filter design. Various relationships among these approaches have been documented in Chang (2002a), Chang (2003b), and Chang (2005).

With all things considered as above, we investigate two intriguing issues in this chapter, which are "to what extent can OSP be applied?" and "how does OSP operate on prior target knowledge?" The first issue will be addressed by deriving OSP from three signal processing perspectives, signal detection, linear discriminant analysis, and parameter estimation that provide evidence that OSP is indeed a versatile technique for a variety of applications. In doing so, we introduce two new signal models, called (**d**,**U**)-model and OSP model. The former separates a desired signal source **d** from undesired signal sources in **U** based on knowledge provided *a priori* so that these two different types of signal sources can be taken care of separately. The latter annihilates the undesired signal sources in **U** from the (**d**,**U**)-model via an OSP operator to reduce the interference caused by the **U** so that the detectability of **d** can be further enhanced and increased. The second issue will be investigated by looking into how target information is used in OSP. Of particular interest is an issue of "how does CEM perform compared to OSP, provided that the undesired signal sources are also known *a priori* and can be annihilated before CEM is applied?." More specifically, "how does CEM perform compared to OSP if OSP-model is used?" While addressing this issue, many interesting results can be obtained based on such OSP-model. Interestingly, under this circumstance, the commonly used least squares-based linear spectral mixture analysis turns out to be OSP. Additionally, we will also show how OSP can be implemented without prior knowledge where OSP takes advantage of the sample spectral correlation to approximate the information that is supposed to be provided by prior knowledge but is not available at the time of data processing. As a result, OSP operates the same form of the RX algorithm developed by Reed and Yu (1991). Furthermore, the low probability detector developed in (Harsanyi, 1993) can, therefore, also be interpreted as a variant of OSP from this aspect by assuming the unity vector as a desired target signature vector.

12.2 Three Perspectives to Derive OSP

Suppose that L is the number of spectral bands and $\mathbf{r}$ is an L-dimensional image pixel vector. Assume that there are p targets, $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_p$ and $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ denote their corresponding signatures, which are generally referred to as digital numbers (DN). A linear mixture of $\mathbf{r}$ models the spectral signature of $\mathbf{r}$ as a linear combination of $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ with appropriate abundance fractions specified by $\alpha_1, \alpha_2, \ldots, \alpha_p$. More precisely, $\mathbf{r}$ is an $L \times 1$ column vector and $\mathbf{M}$ is an $L \times p$ target spectral signature matrix, denoted by $[\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p]$, where $\mathbf{m}_j$ is an $L \times 1$ column vector represented by the spectral signature of the jth target $\mathbf{t}_j$ resident in the pixel vector $\mathbf{r}$. Let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_p)^T$ be a $p \times 1$ abundance column vector associated with $\mathbf{r}$, where α_j denotes the abundance fraction of the jth target signature $\mathbf{m}_j$ present in the pixel vector $\mathbf{r}$.

A classical approach to solving a mixed pixel classification problem is linear unmixing, which assumes that the spectral signature of the pixel vector $\mathbf{r}$ is linearly mixed by $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$, the spectral signatures of the p targets, $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_p$ as follows:

$$\mathbf{r} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n} \tag{12.1}$$

where $\mathbf{n}$ is noise or can be interpreted as a measurement or model error.

Equation (12.1) represents a standard model for signal detection in noise as $\mathbf{s} + \mathbf{n}$, where $\mathbf{M}\boldsymbol{\alpha}$ is considered as a desired signal vector $\mathbf{s} = \mathbf{M}\boldsymbol{\alpha}$ needed to be detected and $\mathbf{n}$ is a corrupted noise. Since we are interested in detecting one target at a time, we can divide the set of the p targets, $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_p$ into a desired target, say $\mathbf{t}_p$ and a class of undesired targets, $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{p-1}$. In this case, a logical approach is to eliminate the effects caused by the undesired targets $\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{p-1}$ that are considered as interferers to $\mathbf{t}_p$ before the detection of $\mathbf{t}_p$ takes place. With annihilation of the undesired target signatures, the detectability of $\mathbf{t}_p$ can therefore be enhanced. In doing so, we refine the signal detection in noise, $\mathbf{s} + \mathbf{n} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n}$ by first separating $\mathbf{m}_p$ from $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ in $\mathbf{M}$ and rewrite (12.1) as

$$\mathbf{r} = \mathbf{d}\alpha_p + \mathbf{U}\boldsymbol{\gamma} + \mathbf{n} \tag{12.2}$$

where $\mathbf{d} = \mathbf{m}_p$ is the desired spectral signature of $\mathbf{t}_p$ and $\mathbf{U} = \left[\mathbf{m}_1\mathbf{m}_2 \cdots \mathbf{m}_{p-1}\right]$ is the undesired target spectral signature matrix made up of $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{p-1}$ that are the spectral signatures of the remaining $p-1$ undesired targets. Here, without loss of generality we assume that the desired target is a single target $\mathbf{t}_p$ and refer (12.2) to the (**d,U**)-model.

12.2.1 Signal Detection Perspective Derived from (d,U)-Model and OSP-Model

Using the (**d,U**)-model specified by (12.2) we can design an orthogonal subspace projector to annihilate **U** from the pixel vector **r** prior to detection of $\mathbf{t}_p$. One such a desired orthogonal subspace projector is derived in Harsanyi and Chang (1994) and given by

$$P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\mathbf{U}^{\#} \tag{12.3}$$

where $\mathbf{U}^{\#} = \left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T$ is the pseudo-inverse of **U**. The notation $P_{\mathbf{U}}^{\perp}$ indicates that the projector $P_{\mathbf{U}}^{\perp}$ maps the observed pixel vector **r** into the orthogonal complement of $\langle\mathbf{U}\rangle$, denoted by $\langle\mathbf{U}\rangle^{\perp}$.

Applying $P_{\mathbf{U}}^{\perp}$ to (**d,U**)-model results in a new signal detection in noise model

$$P_{\mathbf{U}}^{\perp}\mathbf{r} = P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p + P_{\mathbf{U}}^{\perp}\mathbf{n} \tag{12.4}$$

where the undesired signatures in **U** have been annihilated and the original noise **n** has been also suppressed to $\tilde{\mathbf{n}} = P_{\mathbf{U}}^{\perp}\mathbf{n}$. The model specified by (12.4) will be referred to as the OSP-model thereafter in this chapter.

At this point, it is noteworthy to comment on distinction among the three models specified by (12.1), (12.2), and (12.4). The model in (12.1) is a general signal detection in noise model that only separates a signal source $\mathbf{M}\boldsymbol{\alpha}$ from noise **n**. The (**d**,U)-model is a signal model derived from the general signal detection in noise model by breaking up the considered signal sources into two types of signal sources **d** and **U** provided by prior knowledge. It is a two signal-source (**d**,**U**)-model that allows us to deal with these two types of signal sources, **d**,**U** separately. The OSP-model is a single desired-signal source (**d**) detection in noise model derived from the (**d**,**U**)-model with the **U** in the (**d**,**U**)-model annihilated by $P_{\mathbf{U}}^{\perp}$. Therefore, OSP-model can be considered as a custom-designed signal detection in noise model from (12.1) where the signal and noise sources in (12.1) have been preprocessed by $P_{\mathbf{U}}^{\perp}$ for signal enhancement as well as noise suppression.

If we operate a linear filter specified by a weighting vector **w** on the OSP-model, the filter output is given by $\mathbf{w}^T P_{\mathbf{U}}^{\perp}\mathbf{r} = \mathbf{w}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p + \mathbf{w}^T P_{\mathbf{U}}^{\perp}\mathbf{n}$. One commonly used optimal criterion is maximization of the filter output SNR over the weighting vector **w** defined by

$$\mathrm{SNR}(\mathbf{w}) = \frac{\left[\mathbf{w}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right]^T \alpha_p^2 \left[\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{w}\right]}{\mathbf{w}^T P_{\mathbf{U}}^{\perp} E\left[\mathbf{n}\mathbf{n}^T\right] P_{\mathbf{U}}^{\perp}\mathbf{w}} \tag{12.5}$$

If we further assume that **n** is an additive and zero-mean white noise with variance σ^2, (12.5) can be further reduced to $\mathrm{SNR}(\mathbf{w}) = \left(\alpha_p^2/\sigma^2\right)\frac{\left[\mathbf{w}^T\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)\right]^2}{\mathbf{w}^T P_{\mathbf{U}}^{\perp}\mathbf{w}}$ where $P_{\mathbf{U}}^{\perp}$ is an idempotent projector, that is, $\left(P_{\mathbf{U}}^{\perp}\right)^2 = P_{\mathbf{U}}^{\perp}$ (Scharf, 1991). The maximum of SNR(**w**) in (12.5) over **w** can be obtained by Schwarz's inequality:

$$\left|\mathbf{w}^T\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)\right| \le \|\mathbf{w}\|\left\|P_{\mathbf{U}}^{\perp}\mathbf{d}\right\| \tag{12.6}$$

where $||\mathbf{x}||$ is defined by $||\mathbf{x}|| \equiv (\mathbf{x}^T\mathbf{x})^{1/2}$ and the equality holds if and only if $\mathbf{w} = \kappa P_{\mathbf{U}}^{\perp}\mathbf{d}$ for some constant κ. That is, a linear optimal filter specified by the weighting vector $\mathbf{w}^* = \kappa P_{\mathbf{U}}^{\perp}\mathbf{d}$ produces the maximum filter output SNR given by $\max_{\mathbf{w}}\text{SNR}(\mathbf{w}) = \text{SNR}(\mathbf{w}^*) = \left(\alpha_p^2/\sigma^2\right)\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}$. Such a filter can be realized by a matched filter, $M_{P_{\mathbf{U}}^{\perp}\mathbf{d}}$ defined by

$$M_{P_{\mathbf{U}}^{\perp}\mathbf{d}}(\mathbf{x}) = \kappa\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T\mathbf{x} = \kappa\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{x} \text{ for some nonzero constant } \kappa \tag{12.7}$$

with the matched signal specified by $P_{\mathbf{U}}^{\perp}\mathbf{d}$. Applying the matched filter $M_{P_{\mathbf{U}}^{\perp}\mathbf{d}}$ to the OSP-model results in

$$\mathrm{M}_{P_{\mathbf{U}}^{\perp}\mathbf{d}}\left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right) = \kappa\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p + \kappa\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{n} \tag{12.8}$$

that yields the maximum SNR, $\left(\alpha_p^2/\sigma^2\right)\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}$.

Using (12.8) we can design a linear optimal signal detector for (**d**,**U**)-model, denoted by $\delta^{\text{OSPD}}(\mathbf{r})$ by first implementing an undesired target signature rejecter $P_{\mathbf{U}}^{\perp}$ followed by a matched filter $M_{P_{\mathbf{U}}^{\perp}\mathbf{d}}$ with the matched signal $P_{\mathbf{U}}^{\perp}\mathbf{d}$ as follows:

$$\delta^{\text{OSPD}}(\mathbf{r}) = M_{\mathbf{d}}P_{\mathbf{U}}^{\perp}\mathbf{r} = \kappa\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} \tag{12.9}$$

that is exactly the one derived in Harsanyi and Chang (1994) with $\kappa = 1$.

If $\delta^{\text{OSPD}}(\mathbf{r})$ operates the (**d**,**U**)-model in (12.2), then the result is identical to (12.8). This suggests that if the (**d**,**U**)-model is used, the optimal linear filter in (12.9) requires two filters, $P_{\mathbf{U}}^{\perp}$ and $M_{\mathbf{d}}$ to achieve maximum SNR compared to a single matched filter $M_{P_{\mathbf{U}}^{\perp}\mathbf{d}}$ when OSP-model is used with $P_{\mathbf{U}}^{\perp}$ used as a preprocessing of model (12.1).

12.2.2 Fisher's Linear Discriminant Analysis Perspective from OSP-Model

The OSP-model described by (12.4) can be also interpreted as a two-class classification problem, signal $\tilde{\mathbf{s}} = \alpha_p P_{\mathbf{U}}^{\perp}\mathbf{d}$ and noise $\tilde{\mathbf{n}} = P_{\mathbf{U}}^{\perp}\mathbf{n}$, respectively. Let $\boldsymbol{\mu}_{\tilde{\mathbf{s}}}$ and $\boldsymbol{\Sigma}_{\tilde{\mathbf{s}}}$ be the mean vector and covariance matrix of $\tilde{\mathbf{s}} = \alpha_p P_{\mathbf{U}}^{\perp}\mathbf{d}$, and $\boldsymbol{\mu}_{\tilde{\mathbf{n}}}$ and $\boldsymbol{\Sigma}_{\tilde{\mathbf{n}}}$ be the mean vector and covariance matrix of $\tilde{\mathbf{n}} = P_{\mathbf{U}}^{\perp}\mathbf{n}$. Let a linear discriminant function $\mathbf{y}(\mathbf{x})$ be denoted by a linear form specified by $\mathbf{y}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$. Fisher's ratio criterion as Rayleigh quotient defined in Duda and Hart (1973) is given by

$$J(\mathbf{w}) = \frac{\mathbf{w}^T\left[(\boldsymbol{\mu}_{\tilde{\mathbf{s}}} - \boldsymbol{\mu}_{\tilde{\mathbf{n}}})(\boldsymbol{\mu}_{\tilde{\mathbf{s}}} - \boldsymbol{\mu}_{\tilde{\mathbf{n}}})^T\right]\mathbf{w}}{\mathbf{w}^T[\boldsymbol{\Sigma}_{\tilde{\mathbf{s}}} + \boldsymbol{\Sigma}_{\tilde{\mathbf{n}}}]\mathbf{w}} \tag{12.10}$$

where $(\boldsymbol{\mu}_{\tilde{\mathbf{s}}} - \boldsymbol{\mu}_{\tilde{\mathbf{n}}})(\boldsymbol{\mu}_{\tilde{\mathbf{s}}} - \boldsymbol{\mu}_{\tilde{\mathbf{n}}})^T$ and $\boldsymbol{\Sigma}_{\tilde{\mathbf{s}}} + \boldsymbol{\Sigma}_{\tilde{\mathbf{n}}}$ are called between-class and within-class scatter matrices, respectively. So, finding the Fisher linear discriminant function $\mathbf{y}(\mathbf{x}) = \left(\mathbf{w}^{\text{Fisher}}\right)^T\mathbf{x}$ with the weighting vector specified by $\mathbf{w}^{\text{Fisher}}$ is equivalent to maximizing (12.10) over the **w**, which is in turn to solve the following generalized eigenvalue problem (Stark and Woods, 2002, Theorem 5.5.1, pp. 259–260):

$$[\boldsymbol{\Sigma}_{\tilde{\mathbf{s}}} + \boldsymbol{\Sigma}_{\tilde{\mathbf{n}}}]^{-1}(\boldsymbol{\mu}_{\tilde{\mathbf{s}}} - \boldsymbol{\mu}_{\tilde{\mathbf{n}}})(\boldsymbol{\mu}_{\tilde{\mathbf{s}}} - \boldsymbol{\mu}_{\tilde{\mathbf{n}}})^T\mathbf{w}^{\text{Fisher}} = \lambda\mathbf{w}^{\text{Fisher}} \tag{12.11}$$

If we further assume that the signal $\tilde{\mathbf{s}} = \alpha_p P_{\mathbf{U}}^{\perp}\mathbf{d}$ is deterministic and the noise is zero-mean, $\boldsymbol{\mu}_{\tilde{\mathbf{s}}} = \tilde{\mathbf{s}} = \alpha_p P_{\mathbf{U}}^{\perp}\mathbf{d}$, $\boldsymbol{\mu}_{\tilde{\mathbf{n}}} = \mathbf{0}$, $\boldsymbol{\Sigma}_{\tilde{\mathbf{s}}} = \mathbf{0}$, and $\boldsymbol{\Sigma}_{\tilde{\mathbf{n}}} = E\left[\tilde{\mathbf{n}}\tilde{\mathbf{n}}^T\right] = \sigma^2 P_{\mathbf{U}}^{\perp}$. Equation (12.10) becomes (12.5) and (12.11) is also further reduced to

$$\left[\sigma^2 P_{\mathbf{U}}^{\perp}\right]^{-1}\left[P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p^2\mathbf{d}^T P_{\mathbf{U}}^{\perp}\right]\mathbf{w} = \left(\alpha_p/\sigma\right)^2\left[\mathbf{d}\mathbf{d}^T\right]P_{\mathbf{U}}^{\perp}\mathbf{w} = \lambda\,\mathbf{w} \tag{12.12}$$

Since the rank of the matrix $\mathbf{d}\mathbf{d}^T$ in (12.12) is one, the only nonzero eigenvalue is the maximum eigenvalue $\lambda_{\max}$ that turns out to be the solution to (12.12). Now, we substitute $\tilde{\mathbf{w}} = \kappa\mathbf{d}$ for $\mathbf{w}$ in (12.12), (12.5), and (12.10) and obtain

$$\lambda_{\max} = \left(\alpha_p^2/\sigma^2\right)\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d} \neq 0 \tag{12.13}$$

$$\begin{aligned}\mathrm{SNR}(\tilde{\mathbf{w}}) &= \max_{\mathbf{w}}\{\mathrm{SNR}(\mathbf{w})\} = \left(\alpha_p/\sigma\right)^2\frac{\left[(\kappa\mathbf{d})^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right]^T\left[(\kappa\mathbf{d})^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right]}{(\kappa\mathbf{d})^T P_{\mathbf{U}}^{\perp}(\kappa\mathbf{d})}\\ &= \left(\alpha_p/\sigma\right)^2\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d} \neq 0\end{aligned} \tag{12.14}$$

and

$$\begin{aligned}J(\tilde{\mathbf{w}}) &= \frac{(\kappa\,\mathbf{d})^T\left[\left(\alpha_p P_{\mathbf{U}}^{\perp}\mathbf{d}\right)\left(\alpha_p P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T\right](\kappa\,\mathbf{d})}{(\kappa\,\mathbf{d})^T\sigma^2 P_{\mathbf{U}}^{\perp}(\kappa\,\mathbf{d})} = \left(\alpha_p/\sigma\right)^2\frac{(\kappa\,\mathbf{d})^T\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\mathbf{d}^T P_{\mathbf{U}}^{\perp}\right)(\kappa\mathbf{d})}{(\kappa\mathbf{d})^T P_{\mathbf{U}}^{\perp}(\kappa\mathbf{d})}\\ &= \left(\alpha_p/\sigma\right)^2\frac{\kappa^2\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)}{\kappa^2\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)} = \left(\alpha_p/\sigma\right)^2\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d} \neq 0\end{aligned} \tag{12.15}$$

All of these three equations (i.e., (12.13–12.15)) produce the same result $\left(\alpha_p^2/\sigma^2\right)\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}$. This implies that $\tilde{\mathbf{w}} = \kappa\mathbf{d}$ is a desired eigenvector that yields the maximum eigenvalue $\lambda_{\max}$ and can be used to solve both (12.10) and (12.12), in which case $\tilde{\mathbf{w}}$ becomes $\mathbf{w}^{\mathrm{Fisher}}$, that is, $\tilde{\mathbf{w}} = \mathbf{w}^{\mathrm{Fisher}}$. As a result, Fisher's linear discriminant function for (12.10) or (12.12), denoted by $\delta^{\mathrm{Fisher}}(\mathbf{r})$, can be derived as

$$\delta^{\mathrm{Fisher}}(\mathbf{r}) = \left(\mathbf{w}^{\mathrm{Fisher}}\right)^T\mathbf{r} = \kappa\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} \text{ with } \mathbf{w}^{\mathrm{Fisher}} = \kappa\mathbf{d} \tag{12.16}$$

The above approach to arriving at the Fisher's discriminant function in (12.16) was the same one actually used by Harsanyi and Chang (1994) to derive the OSP classifier, $\delta^{\mathrm{OSP}}(\mathbf{r})$ given by

$$\delta^{\mathrm{OSP}}(\mathbf{r}) = \left(\mathbf{w}^{\mathrm{OSP}}\right)^T P_{\mathbf{U}}^{\perp}\mathbf{r} = \mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} \text{ with } \mathbf{w}^{\mathrm{OSP}} = \mathbf{d} \tag{12.17}$$

Interestingly, the solution $\mathbf{w}^* = \kappa P_{\mathbf{U}}^{\perp}\mathbf{d}$ obtained from the signal detection perspective is different from OSP solution $\mathbf{w}^{\mathrm{OSP}} = \mathbf{d}$ and the solution $\mathbf{w}^{\mathrm{Fisher}} = \kappa\mathbf{d}$ obtained from Fisher's linear discriminant function in that the undesired target signature projector $P_{\mathbf{U}}^{\perp}$ appearing in $\mathbf{w}^*$ is absent in $\mathbf{w}^{\mathrm{OSP}}$ and $\mathbf{w}^{\mathrm{Fisher}}$. However, if we substitute $\mathbf{w}^* = \kappa P_{\mathbf{U}}^{\perp}\mathbf{d}$ for $\mathbf{w}$ in (12.12), (12.13)–(12.15), we still obtain the same result, $\lambda_{\max} = \left(\alpha_p^2/\sigma^2\right)\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}$. This implies that both $\mathbf{w}^{\mathrm{Fisher}} = \kappa P_{\mathbf{U}}^{\perp}\mathbf{d}$ and $\mathbf{w}^* = \kappa P_{\mathbf{U}}^{\perp}\mathbf{d}$ produce the same maximum eigenvalue, $\lambda_{\max} = \left(\alpha_p^2/\sigma^2\right)\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}$. Therefore, OSP-based signal detector, $\delta^{\mathrm{OSPD}}(\mathbf{r})$ specified by (12.9) is actually $\delta^{\mathrm{Fisher}}(\mathbf{r})$ and Harsanyi–Chang's OSP, $\delta^{\mathrm{OSP}}(\mathbf{r})$ specified by (12.16) and (12.17) subject to a constant κ.

It should be also noted that if the between-class scatter matrix and within-class scatter matrix in Fisher's Rayleigh quotient or ratio given in (12.10) are replaced with the data covariance matrix $\boldsymbol{\Sigma}$ and noise covariance matrix as follows:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T[\boldsymbol{\Sigma}]\mathbf{w}}{\mathbf{w}^T[\boldsymbol{\Sigma}_{\tilde{\mathbf{n}}}]\mathbf{w}} \tag{12.18}$$

then using the same assumptions made for (12.12) (i.e., $\boldsymbol{\mu}_{\tilde{\mathbf{s}}} = \tilde{\mathbf{s}} = \alpha_p P_{\mathbf{U}}^{\perp}\mathbf{d}$, $\boldsymbol{\mu}_{\tilde{\mathbf{n}}} = \mathbf{0}$, $\boldsymbol{\Sigma}_{\tilde{\mathbf{s}}} = \mathbf{0}$ and $\boldsymbol{\Sigma}_{\tilde{\mathbf{n}}} = E\left[\tilde{\mathbf{n}}\tilde{\mathbf{n}}^T\right]$) maximizing (12.18) over $\mathbf{w}$ is identical to solving (12.12) for $\mathbf{w}$. In this case, Fisher's Rayleigh quotient or ratio in (12.10) can be interpreted as SNR.

12.2.3 Parameter Estimation Perspective from OSP-Model

In signal detection the primary task is to detect the desired target t_p in noise using (12.1). As shown in the above derivations, using OSP-model specified by (12.4) can improve and increase signal detectability of using (12.1). In pattern classification, the desired target signal t_p is discriminated from noise using a between-class scatter matrix/within-class scatter matrix criterion specified by (12.10). Both of these approaches do not intend to estimate its desired signature abundance fraction α_p. In this subsection, we look into a least squares (LS) approach to estimating the abundance fraction α_p of the desired target signature $\mathbf{d}$. Using OSP-model and least squares error (LSE) as a criterion for optimality, we can show that the LS estimate of α_p, $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ minimizing

$$\min_{\alpha_p}\left\{\left(P_{\mathbf{U}}^{\perp}\mathbf{r} - P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p\right)^T\left(P_{\mathbf{U}}^{\perp}\mathbf{r} - P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p\right)\right\} \tag{12.19}$$

is also the LS solution to LSMA using (12.1) as a model to perform spectral unmixing.

Differentiating (12.19) with respect to α_p and setting it to zero results in

$$\left[P_{\mathbf{U}}^{\perp}\mathbf{r} - P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p\right]_{\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})} = 0 \tag{12.20}$$

that yields the solution to (12.19), denoted by $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ and given by

$$\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r}) = \mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} \Rightarrow \delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r}) = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} \tag{12.21}$$

Comparing $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ to $\delta^{\mathrm{OSP}}(\mathbf{r})$, there is a scaling constant $\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$ appearing in $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$, but absent in $\delta^{\mathrm{OSP}}(\mathbf{r})$. In other words, (12.17) and (12.21) are related by

$$\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r}) = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}\delta^{\mathrm{OSP}}(\mathbf{r}) \tag{12.22}$$

where the scaling constant $\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$ is the consequence of LSE resulting from the estimation problem using the OSP-model in (12.19). This constant is included to account for estimation accuracy, not treated as a normalization constant as commonly assumed.

It should be noted that the approach presented above to re-derive $\delta^{\mathrm{OSP}}(\mathbf{r})$ is different from that developed in (Tu et al., 1997; Chang, 1998, 2003a; Chang et al., 1998), all of which use the oblique subspace projection (Scharf, 1991).

12.2.4 Relationship Between $\delta_{\alpha_p}^{LS}(\mathbf{r})$ and Least-Squares Linear Spectral Mixture Analysis

In order to see how $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ is related to the commonly used LS-LSMA, we minimize the LSE resulting from (12.1) as follows:

$$\min_{\boldsymbol{\alpha}}\left\{(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})^T(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})\right\} \tag{12.23}$$

The LS solution to (12.23), denoted by $\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r})$ is given by Scharf (1991)

$$\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r}) = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{r} \tag{12.24}$$

The major difference between $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ and $\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r})$ is that the former is a scalar parameter estimate of α_p, whereas the latter is a vector parameter estimate of the abundance vector $\boldsymbol{\alpha}$. It has been shown in Settle (1996) that $\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r})$ can be decomposed as $\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r}) = \left(\hat{\boldsymbol{\gamma}}^{\mathrm{LS}}(\mathbf{r}), \hat{\alpha}_p^{\mathrm{LS}}(\mathbf{r})\right)$ with

$$\begin{aligned}
\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r}) &= \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{r} = \begin{pmatrix} \hat{\boldsymbol{\gamma}}^{\mathrm{LS}}(\mathbf{r}) \\ \hat{\alpha}_p^{\mathrm{LS}}(\mathbf{r}) \end{pmatrix} \\
&= \begin{bmatrix} \left(\mathbf{U}^T\mathbf{U}\right)^{-1} + \beta\mathbf{U}^{\#}\mathbf{d}\mathbf{d}^T\left(\mathbf{U}^{\#}\right)^T & -\beta\mathbf{U}^{\#}\mathbf{d} \\ -\beta\mathbf{d}^T\left(\mathbf{U}^{\#}\right)^T & \beta \end{bmatrix} \begin{pmatrix} \mathbf{U}^T \\ \mathbf{d}^T \end{pmatrix}\mathbf{r} \\
&= \begin{pmatrix} \mathbf{U}^{\#} + \beta\mathbf{U}^{\#}\mathbf{d}\mathbf{d}^T\mathbf{U}\mathbf{U}^{\#} - \beta\mathbf{U}^{\#}\mathbf{d}\mathbf{d}^T \\ \beta\mathbf{d}^T - \beta\mathbf{d}^T\mathbf{U}\mathbf{U}^{\#} \end{pmatrix}\mathbf{r} = \begin{pmatrix} \mathbf{U}^{\#} - \beta\mathbf{U}^{\#}\mathbf{d}\mathbf{d}^T P_{\mathbf{U}}^{\perp} \\ \beta\mathbf{d}^T P_{\mathbf{U}}^{\perp} \end{pmatrix}\mathbf{r}
\end{aligned} \tag{12.25}$$

where $\hat{\boldsymbol{\gamma}}^{\mathrm{LS}}(\mathbf{r})$ is the LS estimated abundance vector of $\left(\alpha_1, \alpha_2, \ldots, \alpha_{p-1}\right)^T$ and $\beta = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$. Combining (12.22) and (12.25) results in

$$\hat{\alpha}_p^{\mathrm{LS}}(\mathbf{r}) = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} = \delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r}) \tag{12.26}$$

where $\hat{\alpha}_p^{\mathrm{LS}}(\mathbf{r})$ is the pth component of $\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r})$ in (12.25) and also the LS estimate of α_p in (12.22). The same argument can be carried out for all other abundance fractions, $\alpha_1, \alpha_2, \ldots, \alpha_{p-1}$. If we let $\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r}) = \left(\hat{\alpha}_1^{\mathrm{LS}}(\mathbf{r}), \hat{\alpha}_2^{\mathrm{LS}}(\mathbf{r}), \ldots, \hat{\alpha}_p^{\mathrm{LS}}(\mathbf{r})\right)^T$ and $\delta_{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r}) = \left(\delta_1^{\mathrm{LS}}(\mathbf{r}), \delta_2^{\mathrm{LS}}(\mathbf{r}), \ldots, \delta_p^{\mathrm{LS}}(\mathbf{r})\right)^T$, then

$$\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r}) = \delta_{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r}) \tag{12.27}$$

where $\delta_{\alpha_j}^{\mathrm{LS}}(\mathbf{r}) = \hat{\alpha}_j^{\mathrm{LS}}(\mathbf{r})$ for $j = 1, 2, \ldots, p$.

Now, we further introduce the jth component projection function $\mathbf{1}_j$ defined by

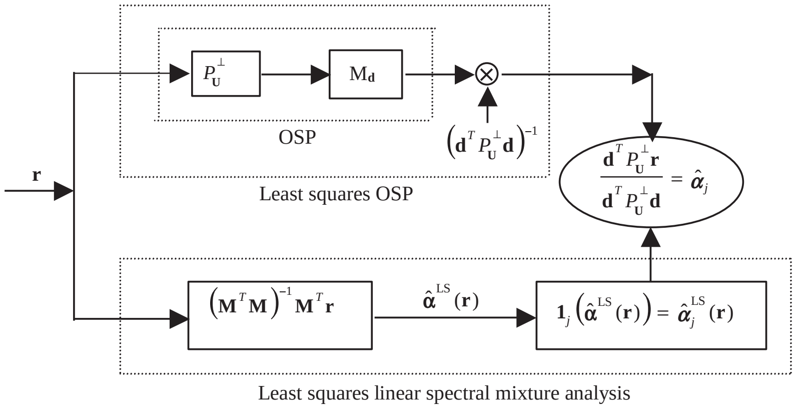
$$\mathbf{1}_j\left(\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r})\right) = \hat{\alpha}_j^{\mathrm{LS}}(\mathbf{r}); \tag{12.28}$$

then we can rewrite (12.27) as

$$\mathbf{1}_j\left(\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r})\right) = \mathbf{1}_j\left(\delta_{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r})\right) \Rightarrow \hat{\alpha}_j^{\mathrm{LS}}(\mathbf{r}) = \delta_{\alpha_j}^{\mathrm{LS}}(\mathbf{r}) \text{ for } j = 1, 2, \ldots, p \tag{12.29}$$

where (12.26) is its particular case.

In light of (12.29), (12.25)–(12.28), if $\delta_{\alpha_j}^{\mathrm{LS}}(\mathbf{r})$ operates on every individual signature with $\mathbf{m}$ being the jth signature in $\mathbf{M}$, it becomes the commonly used linear spectral unmixing solution, $\hat{\alpha}_j^{\mathrm{LS}}(\mathbf{r})$. Compared to $\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r})$ that solves for all p abundance fractions as a vector, the advantage of using $\delta_{\alpha_j}^{\mathrm{LS}}(\mathbf{r})$ over $\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r})$ is conceptually easy to understand and mathematically simple to implement. In other words, if we are interested in detection or estimation of a target signature of particular interest, all we have to do is (1) to designate this target signature as $\mathbf{d}$, (2) to annihilate all signatures other than $\mathbf{d}$ in $\mathbf{U}$ by $P_{\mathbf{U}}^{\perp}$, and (3) to extract $\mathbf{d}$ using a matched filter with the matched



$P_{\mathbf{U}}^{\perp}$
$\mathbf{M_d}$
OSP
$(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d})^{-1}$
Least squares OSP
$\frac{\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r}}{\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}} = \hat{\alpha}_j$
$\mathbf{r}$
$(\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{r}$
$\hat{\boldsymbol{\alpha}}^{\text{LS}}(\mathbf{r})$
$\mathbf{1}_j(\hat{\boldsymbol{\alpha}}^{\text{LS}}(\mathbf{r})) = \hat{\alpha}_j^{\text{LS}}(\mathbf{r})$
Least squares linear spectral mixture analysis

Figure 12.1 Diagram of relationships among OSP, least-squares OSP and least-squares LSMA.

signature specified by $\mathbf{d}$. This is equivalent to using OSP-model to estimate the abundance fraction of $\mathbf{d}$ after the undesired target signatures have been annihilated by $P_{\mathbf{U}}^{\perp}$ rather than using $\hat{\boldsymbol{\alpha}}^{\text{LS}}(\mathbf{r})$ to directly estimate the entire abundance fractions $\alpha_1, \alpha_2, \ldots, \alpha_p$ via (12.24). More specifically, if the LS estimation is performed for (12.24) using OSP-model, then (12.24) is reduced to

$$(\hat{\alpha}_j)^{\text{LS}}(P_{\mathbf{U}}^{\perp}\mathbf{r}) = \mathbf{1}_j(\hat{\boldsymbol{\alpha}}^{\text{LS}}(\mathbf{r})) = \hat{\alpha}_j^{\text{LS}}(\mathbf{r}) = \delta_{\alpha_j}^{\text{LS}}(\mathbf{r}) \text{ for } j = 1, 2, \ldots, p \tag{12.30}$$

where $(\hat{\alpha}_j)^{\text{LS}}$ is the LS estimate of α_j based on the ($\mathbf{d}$,$\mathbf{U}$)-model in (12.2) with $\mathbf{d}$ replaced by $\mathbf{m}_j$ and $\mathbf{U}$ set by $\mathbf{U} = [\mathbf{m}_1 \cdots \mathbf{m}_{j-1}\mathbf{m}_{j+1} \cdots \mathbf{m}_p]$. As a consequence, (12.30) is exactly identical to (12.29). Both (12.29) and (12.30) suggest two different ways to estimate the abundance fraction α_j for $j = 1, 2, \ldots, p$. In other words, (12.30) first projects the data to the space that is orthogonal to the space linearly spanned by the undesired target signatures in $\mathbf{U}$ using $P_{\mathbf{U}}^{\perp}$, then LS estimates the abundance fraction of the desired target signature, $\mathbf{d}$. This is actually the approach taken by OSP in (12.17). In contrast, (12.29) is the commonly used LS-LSMA that performs a vector parameter estimation, then uses a projection function defined by (12.28) to yield the abundance fraction of the desired target signature, $\mathbf{d}$. The relationship between these two equations is delivered by (12.28) and (12.30), which were overlooked in the past. This is very important because many subspace-based vector parameter estimation methods can be interpreted by OSP via (12.25), (12.25)–(12.30). A diagram to illustrate relationships among OSP, the least squares OSP and the LS-LSMA is depicted in Figure 12.1 where the abundance fraction α_j is estimated.

As a concluding remark, it is worth noting that the idea of using OSP-model to re-derive OSP provides new insights into OSP, particularly, the approaches to linear discriminant analysis and parameter estimation, and the relationship between OSP and the LS-LSMA via OSP-model.

12.3 Gaussian Noise in OSP

The noise assumed in (12.1) is nothing more than additive, zero-mean, and white. More precisely, the noise assumed in (12.1) is *uncorrelated* with target signatures in $\mathbf{M}$ and is a zero-mean *de-correlated* (i.e., the noise covariance matrix is an identity matrix) random process. These two assumptions are not crucial and can be relaxed by data preprocessing. The assumption of additivity can be achieved by an estimation technique such as LS methods (Tu et al., 1997; Chang et al., 1998; Chang, 2003a) to remove correlation between target signal subspace and noise subspace.

The assumption of zero-mean white noise can be accomplished by a prewhitening process described in Section 6.3.1, a widely used technique in communications and signal processing community (Poor, 1994). Since SNR is generally very high in hyperspectral imagery, the correlation of the noise subspace with the target signature subspace is significantly reduced compared to that in multispectral imagery. This may be one of the major reasons that OSP has been successful even though it violates the additivity assumption and white noise, but the consequence does not cause much performance deterioration. Nevertheless, by taking advantage of the Gaussian assumption many research efforts have produced satisfactory results (Tu et al., 1997; Chang et al., 1998; Chang, 2003a; Manolakis, 2001) as follows.

12.3.1 Signal Detector in Gaussian Noise Using OSP-Model

In this section, we investigate the role of Gaussian noise assumption in OSP. Specifically, when OSP-model is cast as a two-hypothesis (signal and noise) problem, OSP becomes a maximum likelihood detector. Moreover, if OSP is used as a signal estimator, it can further be shown to be equivalent to the maximum likelihood estimator, which includes the two-class Gaussian discriminant function as a special case.

In what follows, we assume that the noise in (12.1) is zero-mean Gaussian with the covariance matrix given by $\boldsymbol{\Sigma}_{\mathbf{n}}$. In this case, the probability distribution of $\mathbf{r}$ in (12.1) is a Gaussian distribution $p(\mathbf{r}|\mathbf{M}\boldsymbol{\alpha}) = N(\mathbf{M}\boldsymbol{\alpha}, \boldsymbol{\Sigma}_{\mathbf{n}})$ with mean vector and covariance matrix given by $\mathbf{M}\boldsymbol{\alpha}$ and $\boldsymbol{\Sigma}_{\mathbf{n}}$, respectively. Similarly, we can obtain the probability distribution for $P_{\mathbf{U}}^{\perp}\mathbf{r}$ in OSP-model specified by (12.4), which is $p(P_{\mathbf{U}}^{\perp}\mathbf{r}|P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p) = N(P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p, \boldsymbol{\Sigma}_{\tilde{\mathbf{n}}})$ with $\boldsymbol{\Sigma}_{\tilde{\mathbf{n}}} = P_{\mathbf{U}}^{\perp}\boldsymbol{\Sigma}_{\mathbf{n}}\left(P_{\mathbf{U}}^{\perp}\right)^T$. Using the OSP-model as a signal detection model, a standard signal detection problem can be formed by the following binary hypothesis test:

$$\begin{aligned} &H_0 : \ \tilde{\mathbf{n}} \approx p_0(\mathbf{z}) = N(\mathbf{0}, \Sigma_{\tilde{\mathbf{n}}}) \\ &\text{versus} \\ &H_1 : \ P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p + \tilde{\mathbf{n}} \approx p_1(\mathbf{z}) = N\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p, \Sigma_{\tilde{\mathbf{n}}}\right) \end{aligned} \tag{12.31}$$

where $\mathbf{z} = P_{\mathbf{U}}^{\perp}\mathbf{r}$ and $\tilde{\mathbf{n}} = P_{\mathbf{U}}^{\perp}\mathbf{n}$. Following a standard derivation in Poor (1994) a likelihood ratio test (LRT) $\Lambda(\mathbf{z})$ resulting from (12.31) can be obtained by

$$\Lambda(\mathbf{z}) = \log\left(\frac{p_1(z)}{p_0(z)}\right) = \left(\alpha_p P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T \Sigma_{\hat{\mathbf{n}}}^{-1}\mathbf{z} = \alpha_p \mathbf{d}^T \Sigma_{\hat{\mathbf{n}}}^{-1}\mathbf{r} \tag{12.32}$$

However, any color Gaussian noise can further be simplified by the whitening process (Poor, 1994, pp. 58–60) or in Section 6.3.1 and reduced to a white Gaussian noise (WGN) with $\boldsymbol{\Sigma}_{\mathbf{n}} = \sigma^2\mathbf{I}$. In this case, the LRT $\Lambda(\mathbf{z})$ in (12.32) becomes $\Lambda(\mathbf{z}) = \alpha_p\sigma^{-2}\mathbf{d}^T\mathbf{z} = \alpha_p\sigma^{-2}\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r}$ that has the same filter structure as $\delta^{\mathrm{OSPD}}(\mathbf{r})$ specified by (12.9) and $\delta^{\mathrm{OSP}}(\mathbf{r})$ specified by (12.17). Let the false alarm probability be denoted by P_F and define $\Phi(x) = \left(1/\sqrt{2\pi}\right)\int_{-\infty}^{x} e^{-t^2/2}dt$, then

$$P_F = \int_{\Lambda(\mathbf{z})\geq\tau} p_0(\mathbf{z})d\mathbf{z} = \frac{1}{\sqrt{2\pi}}\int_{\tau\left(\alpha_p/\sigma\right)^{-1}}^{\infty} e^{-\frac{1}{2}\left(\frac{t}{\sigma\sqrt{\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}}}\right)^2} dt \tag{12.33}$$

that produces the threshold

$$\tau = \sigma^{-1}\alpha_p\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{1/2}\Phi^{-1}(1 - P_F) \tag{12.34}$$

The detection probability

$$\begin{aligned} P_D &= \int_{\Lambda(\mathbf{z})\geq\tau} p_1(\mathbf{z})d\mathbf{z} \\ &= 1 - \Phi^{-1}\left(\Phi^{-1}(1-P_F) - \sigma^{-1}\alpha_p\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{1/2}\right) \end{aligned} \tag{12.35}$$

turns out to be identical to the one derived in Tu et al. (1997), Chang et al. (1998), and Chang (2003a).

12.3.2 Gaussian Maximum Likelihood Classifier Using OSP-Model

Once again, OSP-model is used with $\mathbf{z} = P_{\mathbf{U}}^{\perp}\mathbf{r}$. Let C_0 and C_1 represent two classes corresponding to noise and signal, respectively. Their discriminant functions can be specified by their corresponding *a posteriori* probability distributions given by $\mathbf{y}_i(\mathbf{z}) = P(C_i|\mathbf{z})$ for $i = 0, 1$. In other words, $\mathbf{z}$ is assigned to class C_1, that is, $\mathbf{z} \in C_1$ if $P(C_1|\mathbf{z}) = P(C_0|\mathbf{z})$, and $\mathbf{z} \in C_0$, otherwise. In this case, we can derive

$$\begin{aligned} P(C_1|\mathbf{z}) > P(C_0|\mathbf{z}) &\Rightarrow P(\mathbf{z}|C_1)P(C_1) > P(\mathbf{z}|C_0)P(C_0) \\ &\Rightarrow \frac{P(\mathbf{z}|C_1)}{P(\mathbf{z}|C_0)} > \frac{P(C_0)}{P(C_1)} \\ &\Rightarrow \Lambda(\mathbf{z}) \equiv \log\left(\frac{p_1(\mathbf{z})}{p_0(\mathbf{z})}\right) > \log\left(\frac{P(C_0)}{P(C_1)}\right) \end{aligned} \tag{12.36}$$

where $P(C_1|\mathbf{z}) = P(C_0|\mathbf{z})$ and $p_0(\mathbf{z}) = P(\mathbf{z}|C_0)$, and $P(C_0)$ and $P(C_1)$ are prior probabilities of C_0 and C_1, respectively. Equation (36) turns out to be the LRT $\Lambda(\mathbf{z})$ in (12.31) with the threshold given by $\log(P(C_0)/P(C_1))$. If we further assume that the prior probabilities $P(C_0)$ and $P(C_1)$ are equally likely, the discriminant function described by (12.36) is reduced to the maximum likelihood detector given by

$$P(\mathbf{z}|C_1) > P(\mathbf{z}|C_0) \Leftrightarrow p_1(\mathbf{z}) > p_0(\mathbf{z}) \Leftrightarrow \mathbf{z} \in C_1 \tag{12.37}$$

With the Gaussian noise assumption (12.37) can be calculated and expressed as follows:

$$-\frac{\left(\mathbf{z} - P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p\right)^T \Sigma_{\tilde{\mathbf{n}}}^{-1}\left(\mathbf{z} - P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p\right)}{2\sigma^2} > -\frac{\mathbf{z}^T\Sigma_{\tilde{\mathbf{n}}}^{-1}\mathbf{z}}{2\sigma^2} \Leftrightarrow \mathbf{z} \in C_1 \tag{12.38}$$

$$\mathbf{z}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p = \mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{z}\alpha_p > \mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p^2/2 \Leftrightarrow \mathbf{z} \in C_1 \tag{12.39}$$

$$\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{z} > \alpha_p/2 \Leftrightarrow \mathbf{z} \in C_1 \tag{12.40}$$

Equation (12.40) makes sense since we assume that the noise is zero-mean and the prior probabilities of the noise class C_0 and the signal class C_1 are equally likely. If we substitute $P_{\mathbf{U}}^{\perp}\mathbf{r}$ for $\mathbf{z}$ in (12.40), then (12.40) becomes

$$\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} > \alpha_p/2 \Leftrightarrow P_{\mathbf{U}}^{\perp}\mathbf{r} \in C_1 \tag{12.41}$$

where the left-hand side of (12.41) is exactly $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ given by (12.22). In this case, Equation (12.41) can be considered as Gaussian discriminant function for (**d**,**U**)-model.

12.3.3 Gaussian Maximum Likelihood Estimator

Using the Gaussian noise assumed in OSP-model, the maximum likelihood estimate of the abundance fraction α_p, $\delta_{\alpha_p}^{\mathrm{GML}}(\mathbf{z})$ is then given by

$$\delta_{\alpha_p}^{\mathrm{GML}}(\mathbf{z}) = \max_{\alpha_p}\left\{p(\mathbf{z}|P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p)\right\} \tag{12.42}$$

where $\mathbf{z} = P_{\mathbf{U}}^{\perp}\mathbf{r}$, $\boldsymbol{\Sigma}_{\tilde{\mathbf{n}}} = P_{\mathbf{U}}^{\perp}\boldsymbol{\Sigma}_{\mathbf{n}}\left(P_{\mathbf{U}}^{\perp}\right)^T$, and $p(\mathbf{z}|P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p) = N(P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p, \boldsymbol{\Sigma}_{\tilde{\mathbf{n}}})$ as defined in (12.31). Solving (12.42) is equivalent to minimizing the following the Mahalanobis distance (Fukunaga, 1990):

$$\delta_{\alpha_p}^{\mathrm{GML}}(\mathbf{z}) = \min_{\alpha_p}\left\{\left(\mathbf{z} - P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p\right)^T\Sigma_{\tilde{\mathbf{n}}}^{-1}\left(\mathbf{z} - P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p\right)\right\} \tag{12.43}$$

that yields the solution

$$\delta_{\alpha_p}^{\mathrm{GML}}(\mathbf{z}) = \left(\mathbf{d}^T\Sigma_{\tilde{\mathbf{n}}}^{-1}\mathbf{d}\right)^{-1}\mathbf{d}^T\Sigma_{\tilde{\mathbf{n}}}^{-1}\mathbf{z} \tag{12.44}$$

Substituting $\mathbf{z} = P_{\mathbf{U}}^{\perp}\mathbf{r}$ and using $\left(P_{\mathbf{U}}^{\perp}\right)^2 = P_{\mathbf{U}}^{\perp}$ we obtain

$$\delta_{\alpha_p}^{\mathrm{GML}}(P_{\mathbf{U}}^{\perp}\mathbf{r}) = \left(\mathbf{d}^T\Sigma_{\tilde{\mathbf{n}}}^{-1}\mathbf{d}\right)^{-1}\mathbf{d}^T\Sigma_{\tilde{\mathbf{n}}}^{-1}\mathbf{r} = \left(\mathbf{d}^T\Sigma_{\tilde{\mathbf{n}}}^{-1}\mathbf{d}\right)^{-1}\mathbf{d}^T\Sigma_{\tilde{\mathbf{n}}}^{-1}\mathbf{r} \tag{12.45}$$

Now, if the Gaussian noise is whitened, that is, $\boldsymbol{\Sigma}_{\tilde{\mathbf{n}}} = \sigma^2\mathbf{I}$, (12.45) becomes

$$\delta_{\alpha_p}^{\mathrm{GML}}(P_{\mathbf{U}}^{\perp}\mathbf{r}) = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} = \delta_{\alpha_p}^{\mathrm{GML}}(\mathbf{r}) \tag{12.46}$$

that is exactly the same one derived in Settle (1996), Chang et al. (1998), and Chang (2003a). The abundance fraction of the desired target signature $\mathbf{d}$ is estimated by $\delta_{\alpha_p}^{\mathrm{GML}}(\mathbf{r})$ that can be obtained directly from the Gaussian maximum likelihood estimator $\delta^{\mathrm{GML}}(\mathbf{r}) = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{r}$ and is identical to (12.24). The preprocessing of using $P_{\mathbf{U}}^{\perp}$ to annihilate the undesired target signatures is not necessary for $\delta^{\mathrm{GML}}(\mathbf{r})$ since it has been already taken care of in the LS estimator shown in (12.25). Once again, (12.46) includes a constant $\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$ that accounts for the LS estimation error and is absent in $\delta^{\mathrm{OSP}}(\mathbf{r})$ given by (12.17).

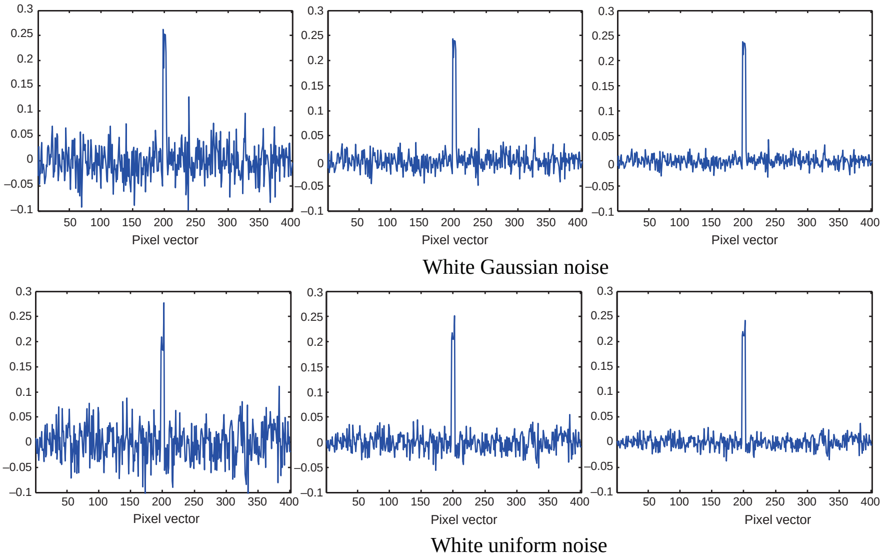
12.3.4 Examples

In what follows, we conduct experiments to examine the noise assumption used in OSP. Two scenarios will be simulated: white Gaussian noise versus white uniform noise (WUN) and color Gaussian noise versus white Gaussian noise.

EXAMPLE 12.1

(White Gaussian Noise vs. White Uniform Noise)

This example demonstrates that the Gaussian noise is an unnecessary assumption for OSP. The set of five reflectance spectra shown in Figure 1.8 is used for illustration and contains five reflectance spectra: dry grass, red soil, creosote leaves, blackbrush, and sagebrush. A signature matrix $\mathbf{M}$ is formed by the dry grass, red soil, and creosote leaves signatures, $\mathbf{M} = [\mathbf{m}_1 \quad \mathbf{m}_2 \quad \mathbf{m}_3]$ with their associated abundance fractions denoted by



(a) SNR = 10:1 (b) SNR = 20:1 (c) SNR = 30:1

Figure 12.2 Detection results of $\delta^{\text{OSP}}(\mathbf{r})$ in white Gaussian noise and white uniform noise with various SNRs.

$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \alpha_3)^T$. The simulation consisted of 401 mixed pixel vectors. We started the first pixel vector with 100% red soil and 0% dry grass, then began to increase 0.25% dry grass and decrease 0.25% red soil every pixel vector until the 401st pixel vector that contained 100% dry grass. We then added creosote leaves to pixel vector numbers 198–202 at abundance fractions 10% while reducing the abundance of red soil and dry grass by multiplying their abundance fractions by 90%. For example, after addition of creosote leaves, the resulting pixel vector 200 contained 10% creosote leaves, 45% red soil, and 45% dry grass. Two types of noise were simulated, white zero-mean Gaussian noise with variance $\sigma^2$and white zero-mean uniform noise with its probability density function defined on $[-a, a]$ and variance $\sigma^2 = a^3/6$. They were added to each band to achieve the signal-to-noise ratio (SNR) defined in Harsanyi and Chang (1994) as 50% reflectance divided by the standard deviation of the noise. Figures 12.2 and 12.3 show the results of OSP detector, $\delta^{\text{OSP}}(\mathbf{r})$ and OSP estimator $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$ in these two types of noise, Gaussian noise and uniform noise with SNR = 30:1, 20:1, and 10:1.

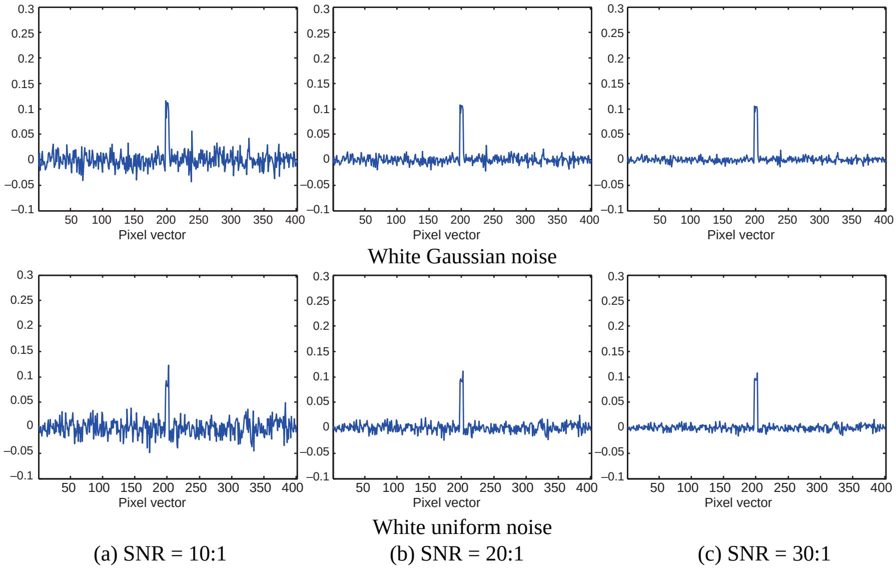
As we can see from these figures, there was no visible difference between Gaussian noise and uniform noise. Table 12.1 also tabulates their corresponding results of $\delta^{\text{OSP}}(\mathbf{r})$ and $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$ in detecting creosote leaves where there was little difference in terms of LSE in abundance fractions between WGN and WUN produced by $\delta^{\text{OSP}}(\mathbf{r})$ and $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$.

Nevertheless, the abundance fractions detected by $\delta^{\text{OSP}}(\mathbf{r})$ and estimated by $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$ were quite different where $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$ produced much more accurate estimates of abundance fractions than does $\delta^{\text{OSP}}(\mathbf{r})$. This was because the scale constant $\left(\mathbf{d}^T P^{\perp}_{\mathbf{U}}\mathbf{d}\right)^{-1}$ in $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$ was included to effectively account for estimation error.

EXAMPLE 12.2

(Gaussian–Markov Noise)

According to the model specified by (12.1) the noise is assumed to be zero-mean and white. More specifically, the noise assumed in the (**d**,**U**)-model is additive and zero-mean. Its covariance matrix is also an identity matrix that implies that the band-to-band noise within a hyperspectral image pixel vector is uncorrelated.



(a) SNR = 10:1 (b) SNR = 20:1 (c) SNR = 30:1

Figure 12.3 Detection results of $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ in white Gaussian noise and white uniform noise with various SNRs.

Interestingly, to the author's best knowledge, most of OSP-based techniques do not include a whitening process, but still successfully achieve their goals. These also include Harsanyi and Chang's OSP (Harsanyi and Chang, 1994). The reason is that due to high spectral resolution the signal-to-noise ratio is generally very high in hyperspectral imagery. In this case, the noise has little impact on OSP performance. Whether or not the noise is white becomes immaterial. This evidence is shown in the following experiments where the whitening process does improve the performance, but the gain is very small.

According to the OSP-model, the whitening was only performed on interband spectral correlation at the pixel level. In this case, a first-order zero-mean Gaussian–Markov noise (GMN) with the between-band correlation coefficient (CC) specified by ρ was added to each pixel vector simulated in Example 12.1 to achieve various levels of SNRs. The covariance matrix of such Gaussian–Markov noise has the form given by $\boldsymbol{\Sigma}_{\mathbf{n}} = \left[\rho^{|i-j|}\right]$, that is:

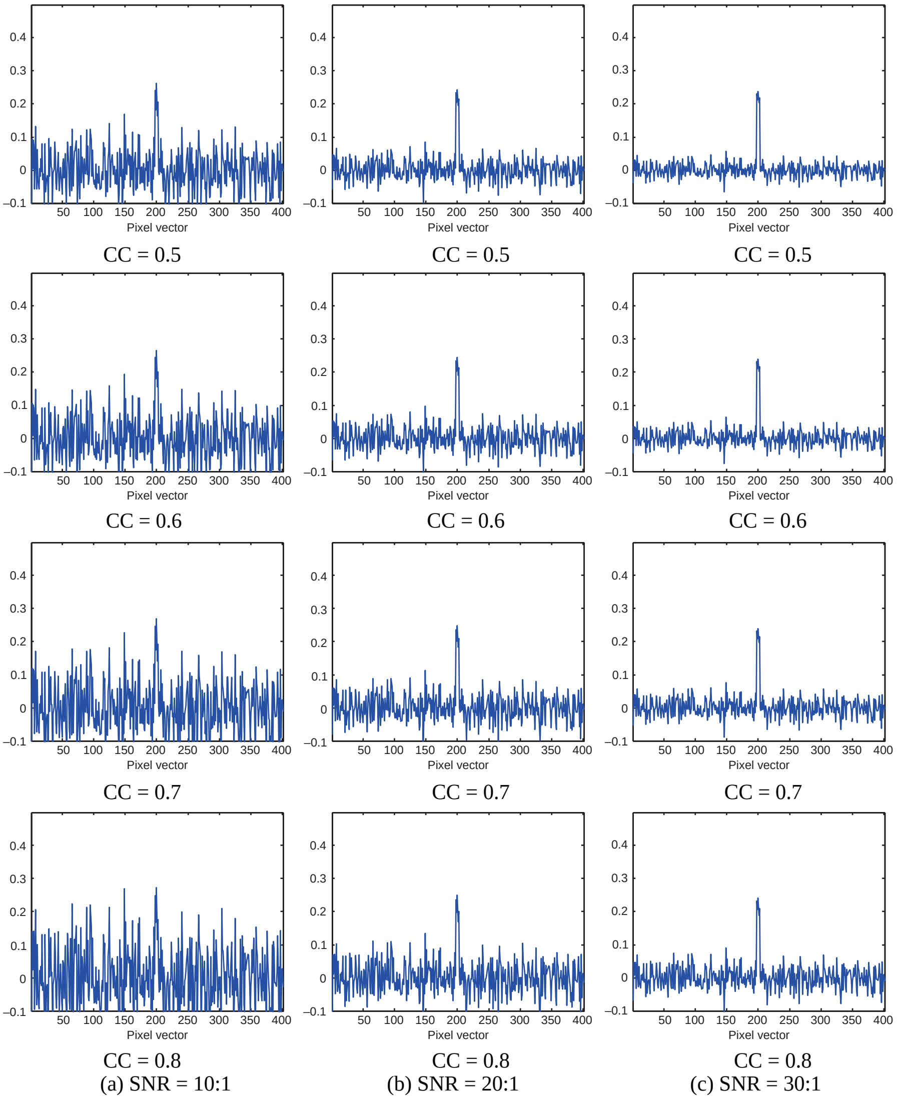
Table 12.1 Detected abundance fractions of "creosote leaves" at 198–202 pixels by $\delta^{\mathrm{OSP}}(\mathrm{r})$ and $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$

	SNR	198	199	200	201	202	LSE
$\delta^{\mathrm{OSP}}(\mathbf{r})$ (WGN)	10:1	0.2616	0.1850	0.2531	0.2518	0.2167	0.093,424
	20:1	0.2436	0.2053	0.2393	0.2386	0.2211	0.084,988
	30:1	0.2375	0.2120	0.2347	0.2343	0.2226	0.082,672
$\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ (WGN)	10:1	0.1160	0.0820	0.1122	0.1116	0.0961	0.00,087,834
	20:1	0.1080	0.0910	0.1061	0.1058	0.0980	0.00,021,959
	30:1	0.1053	0.0940	0.1041	0.1039	0.0987	0.000,097,594
$\delta^{\mathrm{OSP}}(\mathbf{r})$ (WUN)	10:1	0.1851	0.2088	0.1842	0.1828	0.2769	0.064,323
	20:1	0.2053	0.2172	0.2049	0.2042	0.2512	0.069,532
	30:1	0.2121	0.2200	0.2117	0.2113	0.2427	0.072,167
$\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ (WUN)	10:1	0.0821	0.0926	0.0817	0.0811	0.1228	0.0015,905
	20:1	0.0910	0.0963	0.0908	0.0905	0.1114	0.00,039,763
	30:1	0.0940	0.0975	0.0939	0.0937	0.1076	0.00,017,672

$$\Sigma_{\mathbf{n}} = \begin{bmatrix} 1 & \rho & \cdots & \rho^{L-1} \\ \rho & 1 & \rho & \rho^{L-2} \\ \vdots & \rho & \ddots & \rho \\ \rho^{L-1} & \cdots & \rho & 1 \end{bmatrix} \tag{12.47}$$

Figures 12.4 and 12.5 show the results of $\delta^{\text{OSP}}(\mathbf{r})$ and $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$ operating in first-order Gaussian–Markov noise with different correlation coefficients specified by ρ in (12.47) and SNRs.

Table 12.2 also tabulates the detection results of creosote leaves by $\delta^{\text{OSP}}(\mathbf{r})$ and $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$ along with their respective LSEs. In addition, compared to Figures 12.2 and 12.3 and Table 12.1, $\delta^{\text{OSP}}(\mathbf{r})$ and $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$ performed



CC = 0.5 CC = 0.5 CC = 0.5

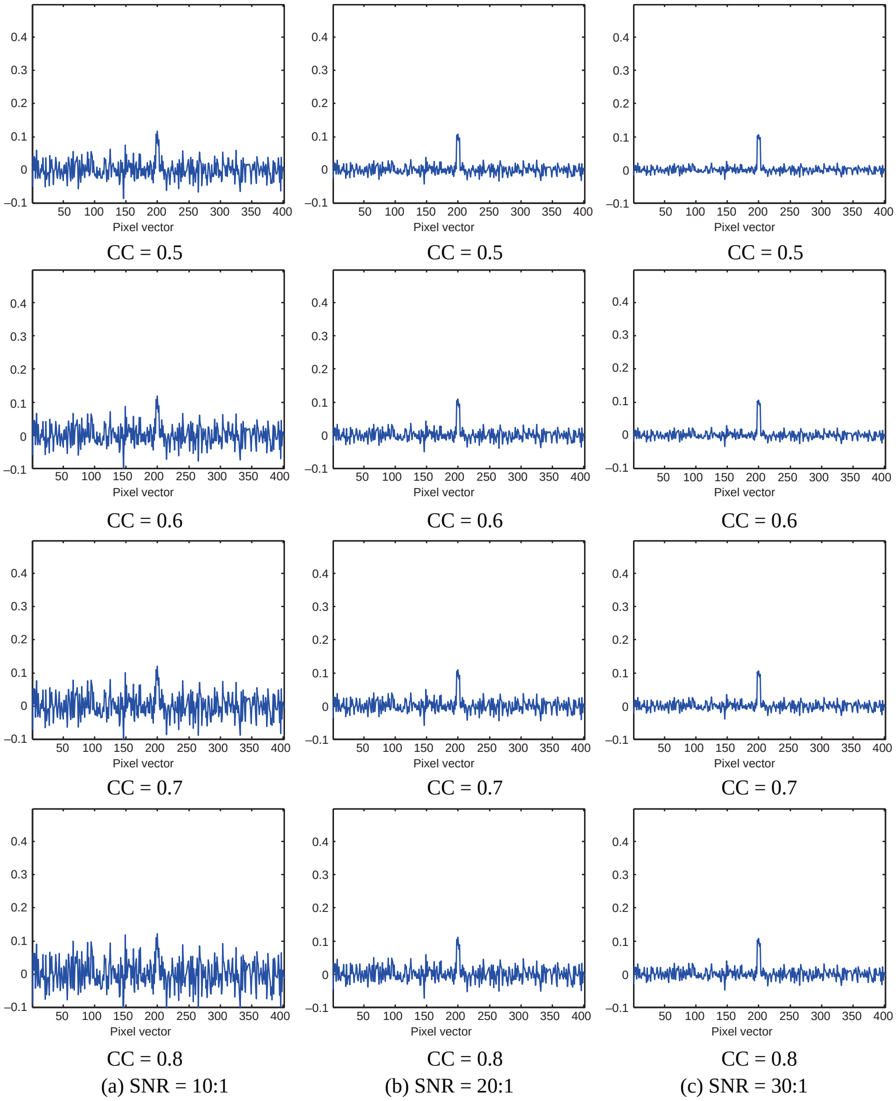
CC = 0.6 CC = 0.6 CC = 0.6

CC = 0.7 CC = 0.7 CC = 0.7

CC = 0.8 CC = 0.8 CC = 0.8

(a) SNR = 10:1 (b) SNR = 20:1 (c) SNR = 30:1

Figure 12.4 Detection results of $\delta^{\text{OSP}}(\mathbf{r})$ in GMN with different correlation coefficients and SNRs.



CC = 0.5 CC = 0.5 CC = 0.5

CC = 0.6 CC = 0.6 CC = 0.6

CC = 0.7 CC = 0.7 CC = 0.7

CC = 0.8 CC = 0.8 CC = 0.8

(a) SNR = 10:1 (b) SNR = 20:1 (c) SNR = 30:1

Figure 12.5 Detection results of $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ in GMN with different correlation coefficients and SNRs.

slightly better in white nose than they did in color noise, but the improvements were very limited. The results also demonstrated that the performance of $\delta^{\mathrm{OSP}}(\mathbf{r})$ and $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ was deteriorated as the CC was increased.

Furthermore, in order to see the effect of noise whitening, Figure 12.6 shows the results of $\delta^{\mathrm{OSP}}(\mathbf{r})$ and $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ where the Gaussian–Markov noise with CC = 0.8 was not whitened and also whitened by using the square-root matrix of $\mathbf{\Sigma}_{\tilde{\mathbf{n}}}^{-1}$, $\mathbf{\Sigma}_{\tilde{\mathbf{n}}}^{-1/2}$ analytically (Poor, 1994, p. 60).

As shown in Figure 12.6, the whitening had slight impact on the performance of $\delta^{\mathrm{OSP}}(\mathbf{r})$ and $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ in the sense that the abundance fractions of creosote leaves and background signatures were detected more accurately. This was particularly visible for $\delta^{\mathrm{OSP}}(\mathbf{r})$. These simple experiments also demonstrated that OSP performance could be improved by a whitening process, but might not be significant. So, the pay-off may not be great given that a reliable estimation of noise covariance may be difficult to obtain.

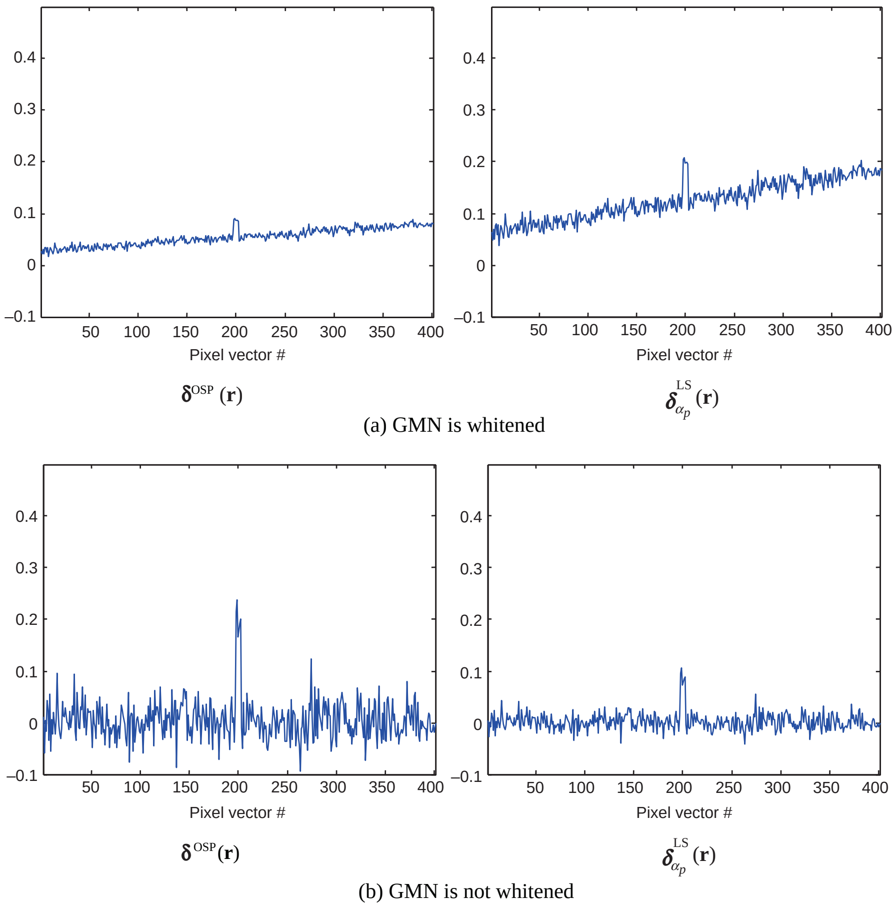
Table 12.2 Detected abundance fractions of “creosote leaves” at 198–202 pixels by $\delta^{\mathrm{OSP}}(\mathrm{r})$ and $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ with LSEs

	SNR	CC	198	199	200	201	202	LSE
$\delta^{\mathrm{OSP}}(\mathbf{r})$ (GMN)	10:1	0.5	0.2419	0.1819	0.2617	0.1642	0.2054	0.068,228
		0.6	0.2439	0.1790	0.2655	0.1550	0.2012	0.067,641
		0.7	0.2462	0.1749	0.2697	0.1405	0.1924	0.065,953
		0.8	0.2477	0.1683	0.2737	0.1141	0.1759	0.062,597
	20:1	0.5	0.2337	0.2037	0.2436	0.1949	0.2155	0.071,598
		0.6	0.2347	0.2023	0.2455	0.1903	0.2134	0.070,798
		0.7	0.2359	0.2002	0.2476	0.1830	0.2090	0.069,052
		0.8	0.2366	0.1968	0.2496	0.1698	0.2007	0.06,545
	30:1	0.5	0.2310	0.2110	0.2376	0.2051	0.2188	0.07,357
		0.6	0.2317	0.2100	0.2389	0.2020	0.2174	0.072,923
		0.7	0.2324	0.2087	0.2402	0.1972	0.2145	0.071,559
		0.8	0.2329	0.2064	0.2416	0.1884	0.2090	0.06,873
$\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ (GMN)	10:1	0.5	0.1072	0.0807	0.1160	0.0728	0.0911	0.0015,011
		0.6	0.1082	0.0794	0.1177	0.0687	0.0892	0.0018,998
		0.7	0.1092	0.0776	0.1196	0.0623	0.0853	0.0026,091
		0.8	0.1098	0.0746	0.1214	0.0506	0.0780	0.0041,221
	20:1	0.5	0.1036	0.0903	0.1080	0.0864	0.0955	0.0003,752
		0.6	0.1041	0.0897	0.1089	0.0844	0.0946	0.0004,749
		0.7	0.1046	0.0888	0.1098	0.0811	0.0926	0.0006,522
		0.8	0.1049	0.0873	0.1107	0.0753	0.0890	0.0010,305
	30:1	0.5	0.1024	0.0936	0.1053	0.0909	0.0970	0.0001667
		0.6	0.1027	0.0931	0.1059	0.0896	0.0964	0.0002,110
		0.7	0.1031	0.0925	0.1065	0.0874	0.0951	0.0002,899
		0.8	0.1033	0.0915	0.1071	0.0835	0.0927	0.0004,580

12.4 OSP Implemented with Partial Knowledge

In Sections 12.2 and 12.3, OSP assumes the complete knowledge of target signatures, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$. In many practical applications, obtaining such full knowledge is generally very difficult if not impossible, specifically, when the image background is not known. In this section, we investigate an issue of how to implement OSP when there is no full knowledge available, particularly, for the case that we are only interested in specific targets, but not image background or other natural signatures.

In his dissertation (Harsanyi, 1993), Harsanyi relaxed the requirement of complete knowledge for OSP by developing an approach called CEM to circumvent this dilemma. The idea is to constrain the desired target signature, **d** with a specific gain while minimizing interfering effects caused by unknown signal sources. Since the undesired target signatures in **U** used by OSP are assumed to be unknown, the undesired target signatures in **U** are suppressed by minimizing their energies instead of being annihilated by a specific operator such as $P_{\mathbf{U}}^{\perp}$ used in OSP. Despite the fact that the relationship between OSP and CEM was reported in Chang 2003a, 2003b) and Du et al. (2003), this section takes an alternative approach to show that with the same assumptions made for OSP, CEM actually performs exactly as does the least squares OSP, $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$.



$\delta^{\mathrm{OSP}}(\mathbf{r})$

$\delta^{\mathrm{LS}}_{\alpha_p}(\mathbf{r})$

(a) GMN is whitened

$\delta^{\mathrm{OSP}}(\mathbf{r})$

$\delta^{\mathrm{LS}}_{\alpha_p}(\mathbf{r})$

(b) GMN is not whitened

Figure 12.6 Results of $\delta^{\mathrm{OSP}}(\mathbf{r})$ and $\delta^{\mathrm{LS}}_{\alpha_p}(\mathbf{r})$ where the GMN with CC = 0.8 with/without being whitened.

12.4.1 CEM

Despite that CEM has been discussed in Section 2.2.3 it is a good idea for our subsequent treatment to recap its approach again as follows.

Let a remotely sensed image be a collection of image pixel vectors, denoted by $\{\mathbf{r}^1, \mathbf{r}^2, \ldots, \mathbf{r}^N\}$ where $\mathbf{r}^i = \left(r_1^i, r_2^i, \ldots, r_L^i\right)^T$ for $1 \leq i \leq N$ is an L-dimensional pixel vector, N is the total number of pixels in the image, and L is the total number of spectral channels. The goal is to design a finite impulse response (FIR) linear filter with L filter coefficients $\{w_1, w_2, \ldots, w_L\}$, denoted by an L-dimensional weighting vector $\mathbf{w} = (w_1, w_2, \ldots, w_L)^T$ that minimizes the filter output energy subject to the constraint $\mathbf{d}^T\mathbf{w} = \mathbf{w}^T\mathbf{d} = 1$.

More specifically, let y^i denote the output of the designed FIR filter resulting from the input $\mathbf{r}^i$. Then y^i can be expressed by

$$y^i = \sum\nolimits_{l=1}^{L} w_l r_l^i = (\mathbf{w})^T \mathbf{r}^i = \left(\mathbf{r}^i\right)^T \mathbf{w} \tag{12.48}$$

and the average energy of the filter output is given by

$$\begin{aligned}\frac{1}{N}\sum_{i=1}^{N}\left(y^i\right)^2 &= \frac{1}{N}\left[\sum_{i=1}^{N}\left(\left(\mathbf{r}^i\right)^T\mathbf{w}\right)^T\left(\mathbf{r}^i\right)^T\mathbf{w}\right] \\ &= \mathbf{w}^T\left[\frac{1}{N}\sum_{i=1}^{N}\mathbf{r}^i\left(\mathbf{r}^i\right)^T\right]\mathbf{w} = \mathbf{w}^T\mathbf{R}\mathbf{w}\end{aligned} \tag{12.49}$$

where $\mathbf{R} = \frac{1}{N}\left[\sum_{i=1}^{N}\mathbf{r}^i\left(\mathbf{r}^i\right)^T\right]$ is the autocorrelation sample matrix of the image. CEM is developed to solve the following linearly constrained optimization problem:

$$\min_{\mathbf{w}}\left\{\mathbf{w}^T\mathbf{R}\mathbf{w}\right\} \text{ subject to } \mathbf{d}^T\mathbf{w} = \mathbf{w}^T\mathbf{d} = 1 \tag{12.50}$$

The optimal solution to (12.50) can be derived in Harsanyi (1993) and Chang (2002) by

$$\mathbf{w}^{\mathrm{CEM}} = \frac{\mathbf{R}^{-1}\mathbf{d}}{\mathbf{d}^T\mathbf{R}^{-1}\mathbf{d}} \tag{12.51}$$

With the optimal weighting vector $\mathbf{w}^{\mathrm{CEM}}$ specified by (12.51) a CEM filter, $\delta^{\mathrm{CEM}}(\mathbf{r})$ was derived in Harsanyi (1993) and given by

$$\delta^{\mathrm{CEM}}(\mathbf{r}) = \left(\mathbf{w}^{\mathrm{CEM}}\right)^T\mathbf{r} = \left(\frac{\mathbf{R}^{-1}\mathbf{d}}{\mathbf{d}^T\mathbf{R}^{-1}\mathbf{d}}\right)^T\mathbf{r} = \frac{\mathbf{d}^T\mathbf{R}^{-1}\mathbf{r}}{\mathbf{d}^T\mathbf{R}^{-1}\mathbf{d}} \tag{12.52}$$

Four special cases are of interest and described as follows.

12.4.1.1 d Is Orthogonal to U (i.e., $P_{\mathbf{U}}^{\perp}\mathbf{d} = \mathbf{d}$) and R = I (i.e., Spectral Correlation is Whitened)

In this case, the noise in the image data to be processed is whitened and assumed to be zero-mean and de-correlated. So, the sample spectral correlation matrix **R** is reduced to the identity matrix **I** and $\mathbf{d}^T\mathbf{R}\mathbf{d} = \mathbf{d}^T\mathbf{d}$. As a result, CEM becomes a normalized spectral matched filter, that is, $\mathbf{R} = \mathbf{I}$. On the other hand, if the desired target signature **d** is further assumed to be orthogonal to **U**, that is, $P_{\mathbf{U}}^{\perp}\mathbf{d} = \mathbf{d}$, then $\delta^{\mathrm{OSP}}(\mathbf{r}) = \mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} = \left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T\mathbf{r} = \mathbf{d}^T\mathbf{r}$. This implies that CEM is identical to OSP subject to a normalization constant $\kappa = \left(\mathbf{d}^T\mathbf{d}\right)^{-1}$. Thus, both OSP and CEM can be considered as the same detector and reduced to a commonly used matched filter with the designated matched signature specified by **d**.

12.4.1.2 An Alternative Approach to Implementing CEM

Comparing $\delta^{\mathrm{CEM}}(\mathbf{r})$ in (12.52) to $\delta^{\mathrm{OSPD}}(\mathbf{r})$ in (12.9), $\delta^{\mathrm{OSP}}(\mathbf{r})$ in (12.17) and $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ in (12.22), we will discover that there is a very close relationship between $P_{\mathbf{U}}^{\perp}$ and $\mathbf{R}^{-1}$. Since the knowledge of the undesired target signature matrix **U** assumed to be known in $P_{\mathbf{U}}^{\perp}$ in $\delta^{\mathrm{OSPD}}(\mathbf{r})$, $\delta^{\mathrm{OSP}}(\mathbf{r})$, and $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ is not available in $\delta^{\mathrm{CEM}}(\mathbf{r})$, $\delta^{\mathrm{CEM}}(\mathbf{r})$ must estimate $P_{\mathbf{U}}^{\perp}$ directly from the image data. One way of doing so is to approximate the "$P_{\mathbf{U}}^{\perp}$" in the sense of minimum LSE by the inverse of the sample spectral information, $\mathbf{R}^{-1}$ that can be obtained directly from the image data. More specifically, $\delta^{\mathrm{CEM}}(\mathbf{r})$ makes use of the *a posteriori* information, $\mathbf{R}^{-1}$ to approximate the *a priori* information $P_{\mathbf{U}}^{\perp}$ to accomplish what $\delta^{\mathrm{OSPD}}(\mathbf{r})$, $\delta^{\mathrm{OSP}}(\mathbf{r})$ and $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ do. The only difference is the scaling constant κ. Since both $\delta^{\mathrm{OSPD}}(\mathbf{r})$ and $\delta^{\mathrm{OSP}}(\mathbf{r})$ are used only for abundance detection, κ is set to 1. To the contrary, $\alpha_p^{\mathrm{LS}}(\mathbf{r})$ is an abundance estimator and the scaling constant $\kappa = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$ is included to account for estimation error (Settle, 1996; Chang et al., 1998; Chang, 2003a). In this case, if we replace $P_{\mathbf{U}}^{\perp}$ in (12.22) with $\mathbf{R}^{-1}$, $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ becomes $\delta^{\mathrm{CEM}}(\mathbf{r})$. This suggests that $\delta^{\mathrm{CEM}}(\mathbf{r})$ can be considered as $\delta_{\alpha_p}^{\mathrm{LS}}(\mathbf{r})$ with the partial knowledge specified by the desired target signature $\mathbf{d}$ and the sample spectral information provided by $\mathbf{R}^{-1}$ used to replace the unknown signature matrix $P_{\mathbf{U}}^{\perp}$.

As noted above, the *a posteriori* information $\mathbf{R}^{-1}$ used in $\delta^{\mathrm{CEM}}(\mathbf{r})$ is intended to approximate the *a priori* information $P_{\mathbf{U}}^{\perp}$ used in $\delta^{\mathrm{OSP}}(\mathbf{r})$. However, the $P_{\mathbf{U}}^{\perp}$ excludes the information provided by the desired target signature $\mathbf{d}$, which is included in $\mathbf{R}^{-1}$. This observation suggests that a more accurate data sample correlation matrix used by $\delta^{\mathrm{CEM}}(\mathbf{r})$ should be the one that removes all the pixel vectors specified by $\mathbf{d}$ from $\mathbf{R}$. We let $\tilde{\mathbf{R}}$ be such a matrix that excludes all target pixel vectors specified by $\mathbf{d}$ and be defined by

$$\tilde{\mathbf{R}} = \mathbf{R} - \frac{1}{N}\sum\nolimits_i \alpha_p^i \mathbf{d}\mathbf{d}^T \tag{12.53}$$

where the superscript "*i*" runs through all target pixel vectors whose signatures are specified by $\mathbf{d}$ and α_p^i indicates their respective abundance fractions contained in pixel vector $\mathbf{r}^i$. Equation (12.53) allows us to rewrite $\mathbf{w}^T\mathbf{R}\mathbf{w}$ in (12.49) as

$$\mathbf{w}^T\mathbf{R}\mathbf{w} = \mathbf{w}^T\tilde{\mathbf{R}}\mathbf{w} + \frac{1}{N}\mathbf{w}^T\left(\sum\nolimits_i \alpha_p^i \mathbf{d}\mathbf{d}^T\right)\mathbf{w} \tag{12.54}$$

Because of the constraint $\mathbf{w}^T\mathbf{d} = 1$, $(\mathbf{w}^T\mathbf{d})^T(\mathbf{w}^T\mathbf{d}) = 1$. In addition, the second term in the right-hand side of (12.54), $\frac{1}{N}\sum_i \alpha_p^i \mathbf{w}^T\mathbf{d}\mathbf{d}^T\mathbf{w} = \frac{1}{N}\sum_i \alpha_p^i$, is independent of $\mathbf{w}$. Solving (12.50) is equivalent to solving the following optimization problem:

$$\min_{\mathbf{w}}\left\{\mathbf{w}^T\tilde{\mathbf{R}}\mathbf{w}\right\} \text{ subject to } \mathbf{w}^T\mathbf{d} = 1 \tag{12.55}$$

with the optimal solution given by

$$\mathbf{w}^* = \left(\mathbf{d}^T\tilde{\mathbf{R}}^{-1}\mathbf{d}\right)^{-1}\tilde{\mathbf{R}}^{-1}\mathbf{d} \tag{12.56}$$

where the $\mathbf{w}^*$ can be obtained by $\mathbf{w}^{\mathrm{CEM}}$ in (12.51) by simply replacing $\mathbf{R}^{-1}$ with $\tilde{\mathbf{R}}^{-1}$. Therefore, technically speaking, (12.56) should be a more appropriate form for CEM, which is also demonstrated in Chang (2003a, pp. 63–67). Nevertheless, the CEM solution outlined by (12.50), (12.50)–(12.52) is still desirable in two aspects. One is that if the number of target pixel vectors specified by the desired signature $\mathbf{d}$ is small, the impact of without removing these pixel vectors will be little and not be significant. Another is that in many practical applications, finding all pixel vectors that are specified by the desired signature $\mathbf{d}$ may not be realistic, particularly, if the $\mathbf{d}$ is a mixed pixel vector.

12.4.1.3 CEM Implemented in Conjunction with $P_{\mathbf{U}}^{\perp}$

More interestingly, when the $\mathbf{U}$ is actually known, CEM should be able to take advantage of this knowledge to annihilate the undesired signatures via $P_{\mathbf{U}}^{\perp}$ instead of suppressing these signatures. In this case, the resulting $\mathbf{r}$ in (12.48) becomes $P_{\mathbf{U}}^{\perp}\mathbf{r}$ and the desired target signature $\mathbf{d}$ is also projected to $P_{\mathbf{U}}^{\perp}\mathbf{d}$. Consequently, the constraint $\mathbf{w}^T\mathbf{d} = 1$ and the object function $\mathbf{w}^T\mathbf{R}\mathbf{w}$ in (12.50) must be replaced with $\mathbf{w}^T\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right) = 1$ and $\mathbf{w}^T\left[\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T\right]\mathbf{w}$, respectively, which results in

$$\begin{aligned}
&\mathbf{w}^T\left[\left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right)\left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right)^T\right]\mathbf{w} \\
&= \alpha_p^2\mathbf{w}^T\left[\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T\right]\mathbf{w} + \mathbf{w}^T\left[E\left(\left(P_{\mathbf{U}}^{\perp}\mathbf{n}\right)\left(P_{\mathbf{U}}^{\perp}\mathbf{n}\right)^T\right)\right]\mathbf{w} \\
&= \alpha_p^2\mathbf{w}^T\left[\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T\right]\mathbf{w} + \mathbf{w}^T\Sigma_{\tilde{\mathbf{n}}}\mathbf{w} = \alpha_p^2 + \mathbf{w}^T\mathbf{K}_{\tilde{\mathbf{n}}}\mathbf{w}
\end{aligned} \tag{12.57}$$

where $\mathbf{K}_{\tilde{\mathbf{n}}} = E[\tilde{\mathbf{n}}\tilde{\mathbf{n}}^T] = E\left[\left(P_{\mathbf{U}}^{\perp}\mathbf{n}\right)\left(P_{\mathbf{U}}^{\perp}\mathbf{n}\right)^T\right]$ the cross-term $\mathbf{d}^T E[\tilde{\mathbf{n}}]$ vanishes if $\mathbf{n}$ is zero-mean. As a consequence, (12.50) is reduced to

$$\min_{\mathbf{w}}\left\{\mathbf{w}^T\Sigma_{\tilde{\mathbf{n}}}\mathbf{w}\right\} \text{ subject to } \mathbf{w}^T\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right) = 1 \tag{12.58}$$

In order to see the relationship between OSP and CEM, we use (12.5) to obtain the filter output SNR as follows:

$$\begin{aligned} \text{SNR}(\mathbf{w}) &= \frac{\mathbf{w}^T\left[\left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right)\left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right)^T\right]\mathbf{w}}{\mathbf{w}^T\Sigma_{\tilde{\mathbf{n}}}\mathbf{w}} = \frac{\alpha_p^2\mathbf{w}^T\left[\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T\right]\mathbf{w} + \mathbf{w}^T\Sigma_{\tilde{\mathbf{n}}}\mathbf{w}}{\mathbf{w}^T\Sigma_{\tilde{\mathbf{n}}}\mathbf{w}} \\ &= \frac{\alpha_p^2\mathbf{w}^T\left[\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T\right]\mathbf{w}}{\mathbf{w}^T\Sigma_{\tilde{\mathbf{n}}}\mathbf{w}} + 1 \end{aligned} \tag{12.59}$$

So, solving (12.58) is equivalent to finding the solution to the following constrained optimization problem:

$$\max_{\mathbf{w}}\{\text{SNR}(\mathbf{w})\} \text{ subject to } \mathbf{w}^T\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right) = 1 \tag{12.60}$$

Now, we use the Largrangian multiplier method by differentiating the following Largrangian:

$$J(\mathbf{w}) = \mathbf{w}^T\Sigma_{\tilde{\mathbf{n}}}\mathbf{w} + \lambda\left(\mathbf{w}^T P_{\mathbf{U}}^{\perp}\mathbf{d} - 1\right) \tag{12.61}$$

with respective to $\mathbf{w}$ and setting it to 0, and obtain

$$\left.\frac{\partial J(\mathbf{w})}{\partial \mathbf{w}}\right|_{\mathbf{w}^*} = 2\Sigma_{\tilde{\mathbf{n}}}\mathbf{w}^* + \lambda P_{\mathbf{U}}^{\perp}\mathbf{d} = 0 \tag{12.62}$$

and

$$\Sigma_{\tilde{\mathbf{n}}}\mathbf{w}^* = -(1/2)\lambda P_{\mathbf{U}}^{\perp}\mathbf{d} \tag{12.63}$$

with

$$\lambda = -2\left(\mathbf{d}^T\Sigma_{\tilde{\mathbf{n}}}^{-1}\mathbf{d}\right)^{-1} \tag{12.64}$$

derived by the constraint $\mathbf{w}^T\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right) = 1$. Substituting (12.64) for λ into (12.63) yields

$$\mathbf{w}^* = \Sigma_{\tilde{\mathbf{n}}}^{-1}\left(\mathbf{d}^T\Sigma_{\tilde{\mathbf{n}}}^{-1}\mathbf{d}\right)^{-1} P_{\mathbf{U}}^{\perp}\mathbf{d} \tag{12.65}$$

CEM specified by (12.65) is called CEM implemented in conjunction with $P_{\mathbf{U}}^{\perp}$ and denoted by $\delta_{P_{\mathbf{U}}^{\perp}}^{\text{CEM}}(\mathbf{r})$.

12.4.1.4 CEM Implemented in Conjunction with $P_{\mathbf{U}}^{\perp}$ in White Noise

If we further assume that the noise in (12.60) is white and given by $\boldsymbol{\Sigma}_{\tilde{\mathbf{n}}} = \sigma^2\mathbf{I}$, (12.60) is reduced to finding a weighting vector $\mathbf{w}$ with the minimum vector length subject to the constraint $\mathbf{w}^T\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right) = 1$, namely,

$$\min_{\mathbf{w}}\left\{\mathbf{w}^T\mathbf{w}\right\} \text{ subject to } \mathbf{w}^T\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right) = 1 \tag{12.66}$$

and (12.65) becomes

$$\mathbf{w}^* = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1} P_{\mathbf{U}}^{\perp}\mathbf{d} \tag{12.67}$$

Let the filter specified by (12.67) be denoted by $\delta_{P_{\mathbf{U}}^{\perp},\text{white}}^{\text{CEM}}(\mathbf{r})$ in which case, CEM assumes that noise is white and the knowledge of the $\mathbf{U}$ is provided *a priori*. Implementing in conjunction with the undesired signature projector $P_{\mathbf{U}}^{\perp}$, $\delta_{P_{\mathbf{U}}^{\perp},\text{white}}^{\text{CEM}}(\mathbf{r})$ becomes the LS estimator $\delta_{\alpha_p}^{\text{LS}}(\mathbf{r})$ given by (12.22) as well as the Gaussian maximum likelihood estimator $\delta_{\alpha_p}^{\text{GML}}(\mathbf{r})$ given by (12.46), respectively. This implies that if the noise is zero mean and white, and the undesired target signatures are annihilated by $P_{\mathbf{U}}^{\perp}$, then $\delta_{P_{\mathbf{U}}^{\perp},\text{white}}^{\text{CEM}}(\mathbf{r})$ performs as if it is an abundance fraction estimator.

12.4.2 TCIMF

CEM is originally designed to detect a single target signature. If there are multiple targets to be detected, the detection must be carried out one target at a time. In order to extend CEM to a multiple-target detection technique, a TCIMF was recently developed by Ren and Chang (2001) that can be viewed as a generalization of OSP and CEM.

Let $\mathbf{D} = [\mathbf{d}_1\mathbf{d}_2\cdots\mathbf{d}_{n_{\mathbf{D}}}]$ denote the desired target signature matrix and $\mathbf{U} = [\mathbf{u}_1\mathbf{u}_2\cdots\mathbf{u}_{n_{\mathbf{U}}}]$ be the undesired target signature matrix where $n_{\mathbf{D}}$ and $n_{\mathbf{U}}$ are the number of the desired target signatures and the number of the undesired target signatures, respectively. Now, we can develop an FIR filter that passes the desired target signatures in $\mathbf{D}$ using an $n_{\mathbf{D}} \times 1$ unity constraint vector $\mathbf{1}_{n_{\mathbf{D}}\times 1} = \underbrace{(1,1,\ldots,1)^T}_{n_{\mathbf{D}}}$ while annihilating the undesired target signatures in $\mathbf{U}$ using an $n_{\mathbf{U}} \times 1$ zero constraint vector $\mathbf{0}_{n_{\mathbf{U}}\times 1} = \underbrace{(0,0,\ldots,0)^T}_{n_{\mathbf{U}}}$. In doing so, the constraint in (12.50) is replaced by

$$[\mathbf{D}\,\mathbf{U}]^T\mathbf{w} = \begin{bmatrix}\mathbf{1}_{n_{\mathbf{D}}\times 1}\\ \mathbf{0}_{n_{\mathbf{U}}\times 1}\end{bmatrix} \tag{12.68}$$

and the optimization problem in (12.50) becomes the following linearly constrained optimization problem:

$$\min_{\mathbf{w}}\left\{\mathbf{w}^T\mathbf{R}\mathbf{w}\right\} \text{ subject to } [\mathbf{D}\,\mathbf{U}]^T\mathbf{w} = \begin{bmatrix}\mathbf{1}_{n_{\mathbf{D}}\times 1}\\ \mathbf{0}_{n_{\mathbf{U}}\times 1}\end{bmatrix} \tag{12.69}$$

The filter solving (12.69) is called target-constrained interference-minimized filter (TCIMF) (Ren and Chang, 2001) and given by

$$\delta^{\text{TCIMF}}(\mathbf{r}) = \left(\mathbf{w}^{\text{TCIMF}}\right)^T\mathbf{r} \tag{12.70}$$

with the optimal weighting vector, $\mathbf{w}^{\text{TCIMF}}$ given by

$$\mathbf{w}^{\text{TCIMF}} = \mathbf{R}^{-1}\,[\mathbf{D}\,\mathbf{U}]\left([\mathbf{D}\,\mathbf{U}]^T\mathbf{R}^{-1}[\mathbf{D}\,\mathbf{U}]\right)^{-1}\begin{bmatrix}\mathbf{1}_{n_{\mathbf{D}}\times 1}\\ \mathbf{0}_{n_{\mathbf{U}}\times 1}\end{bmatrix} \tag{12.71}$$

A discussion on the relationship between OSP and CEM via TCIMF can be found in Chang 2002, 2003a). In what follows, we describe more details about this relationship by considering three special cases of $\delta^{\text{TCIMF}}(\mathbf{r})$.

12.4.2.1 $\mathbf{D} = \mathbf{m}_p = \mathbf{d}$ with $n_{\mathbf{D}} = 1$ and $\mathbf{U} = \left[\mathbf{m}_1\mathbf{m}_2 \cdots \mathbf{m}_{p-1}\right]$ with $n_{\mathbf{U}} = p - 1$

In this case, $\delta^{\text{TCIMF}}(\mathbf{r})$ performs like $\delta^{\text{CEM}}_{P_{\mathbf{U}}^{\perp}}(\mathbf{r})$. However, there is a difference in the sense of algorithm implementation. The former performs extraction of the desired signature $\mathbf{m}_p$ and annihilation of the undesired signatures $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{p-1}$ simultaneously, whereas the latter performs the undesired signature annihilation using $P_{\mathbf{U}}^{\perp}$ followed by CEM, $\delta^{\text{CEM}}(\mathbf{r})$ in sequence. Therefore, despite the fact that both filters may produce the same results, they should be considered as separate filters. In particular, $\delta^{\text{TCIMF}}(\mathbf{r})$ can be carried out in real time as noted in Chang (2003a) and Chang et al. (2001).

12.4.2.2 $\mathbf{D} = \mathbf{m}_p = \mathbf{d}$ with $n_{\mathbf{D}} = 1$ and $\mathbf{U} = \left[\mathbf{m}_1\ \mathbf{m}_2 \cdots \mathbf{m}_{p-1}\right]$ with $n_{\mathbf{U}} = p - 1$ and $\mathbf{R} = \mathbf{I}$

In this case, $\delta^{\text{TCIMF}}(\mathbf{r})$ performs like $\delta^{\text{OSP}}(\mathbf{r})$, but in the mean time, it also suppresses all signatures other than desired and undesired target signatures, an operation that OSP does not do. Let the resulting weighting vector be denoted by $\mathbf{w}^{\text{TCIMF}}_{\mathbf{R}=\mathbf{I}}$ and its corresponding TCIMF be denoted by $\delta^{\text{TCIMF}}_{\mathbf{R}=\mathbf{I}}(\mathbf{r})$. As derived in Chang (2002) and Chang (2003a), the $\delta^{\text{TCIMF}}_{\mathbf{R}=\mathbf{I}}(\mathbf{r})$ can be shown to be equivalent to $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$ or $\delta^{\text{OSP}}(\mathbf{r})$ as follows:

$$\begin{aligned}
\mathbf{w}^{\text{TCIMF}}_{\mathbf{R}=\mathbf{I}} &= \mathbf{I}^{-1}[\mathbf{d}\mathbf{U}]\left[(\mathbf{d}\mathbf{U})^T\mathbf{I}^{-1}(\mathbf{d}\mathbf{U})\right]^{-1}\begin{pmatrix}1\\ \mathbf{0}_{(p-1)\times 1}\end{pmatrix} = [\mathbf{d}\mathbf{U}]\left(\begin{bmatrix}\mathbf{d}^T\mathbf{d} & \mathbf{d}^T\mathbf{U}\\ \mathbf{U}^T\mathbf{d} & \mathbf{U}^T\mathbf{U}\end{bmatrix}\right)^{-1}\begin{pmatrix}1\\ \mathbf{0}_{(p-1)\times 1}\end{pmatrix}\\
&= [\mathbf{d}\mathbf{U}]\begin{bmatrix}\kappa & \kappa\mathbf{d}^T\left(\mathbf{U}^{\#}\right)^T\\ -\kappa\mathbf{U}^{\#}\mathbf{d} & \left(\mathbf{U}^T\mathbf{U}\right)^{-1} + \kappa\mathbf{U}^{\#}\mathbf{d}\mathbf{d}^T\left(\mathbf{U}^{\#}\right)^T\end{bmatrix}\begin{pmatrix}1\\ \mathbf{0}_{(p-1)\times 1}\end{pmatrix}\\
&= [\mathbf{d}\mathbf{U}]\begin{pmatrix}\kappa\\ -\kappa\mathbf{U}^{\#}\mathbf{d}\end{pmatrix} = \kappa\mathbf{d} - \kappa\mathbf{U}^{\#}\mathbf{U}\mathbf{d} = \kappa\left(I - \mathbf{U}\mathbf{U}^{\#}\right)\mathbf{d} = \kappa P_{\mathbf{U}}^{\perp}\mathbf{d}
\end{aligned} \tag{12.72}$$

where $\kappa = \left(\mathbf{d}^T\mathbf{R}\mathbf{d}\right)^{-1}$, and

$$\delta^{\text{TCIMF}}_{\mathbf{R}=\mathbf{I}}(\mathbf{r}) = \delta^{\text{LS}}_{\alpha_p}(\mathbf{r}) = \kappa\delta^{\text{OSP}}(\mathbf{r}) \tag{12.73}$$

It should be noted that the extra constant κ in (12.73) is a result of interference/noise suppression from TCIMF that OSP does not have.

12.4.2.3 $\mathbf{D} = \mathbf{d}$ and $\mathbf{U} = \varnothing$ (i.e., Only the Desired Signature d is Available)

In this case, TCIMF is further reduced to CEM given by

$$\delta^{\text{TCIMF}}_{\mathbf{D}=\mathbf{d},\mathbf{U}=\varnothing}(\mathbf{r}) = \left(\mathbf{w}^{\text{TCIMF}}_{\mathbf{D}=\mathbf{d},\mathbf{U}=\varnothing}\right)^T\mathbf{r} = \delta^{\text{CEM}}(\mathbf{r}) \tag{12.74}$$

So, on one end, OSP and CEM can be considered as special cases of TCIMF by virtue of (12.71). On the other end, OSP can be also interpreted as a noise-whitened version of TCIMF with interference/noise suppression in (12.72). In addition, CEM can be also considered as an undesired target signature-suppressed version of TCIMF in (12.74). Nevertheless, there is a subtle and substantiate distinction between TCIMF and other filters such as OSP and CEM. TCIMF can be implemented to detect and classify multiple targets, annihilate undesired targets, and suppress unknown signal sources in one-shot operation in real time (Chang et al. 2001), whereas $\delta^{\text{CEM}}(\mathbf{r})$ and $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ must be carried out for detection of a single target at a time.

12.4.3 Examples

In this subsection, we conduct a comparative analysis between OSP and CEM with the undesired signatures annihilated by OSP.

EXAMPLE 12.3

(Comparative Study Between (*d*,*U*)-Model and OSP-Model)

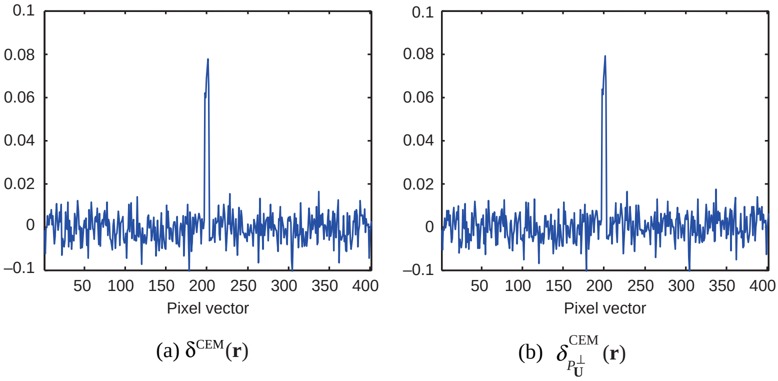
This example assumed that the complete knowledge of target signatures was available. We study how the undesired signatures in $\mathbf{U}$ affect the performance of $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$. The same 401 simulated pixels used in Example 12.1 were also used in this example with added SNR 30:1 white Gaussian noise so that the results of $\delta^{\text{OSP}}(\mathbf{r})$ and $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$ derived in Example 12.1 can be used for comparison. Two scenarios were studied. One was the $\mathbf{U}$ made up of dry grass and red soil. Figure 12.7 shows the detection results of $\delta^{\text{CEM}}(\mathbf{r})$ and $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ with dry grass and red soil annihilated by $P^{\perp}_{\mathbf{U}}$. Table 12.3 tabulates their respective LSEs.

As we can see from Table 12.3, $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ with the dry grass and red soil annihilated performed slightly better than $\delta^{\text{CEM}}(\mathbf{r})$ that only suppressed the dry grass and red soil. However, if we further considered another scenario which included the sagebrush in the $\mathbf{U}$ as another undesired signature even if it was absent in the 401 simulated pixel vectors to repeat the same experiment.

Figure 12.8 shows the detection results of $\delta^{\text{CEM}}(\mathbf{r})$ and $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ with dry grass, red soil, and sagebrush annihilated.

Table 12.3 Comparison of detected abundance fractions between $\delta^{\text{CEM}}(\mathbf{r})$ and $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ with dry grass and red soil annihilated by $P^{\perp}_{\mathbf{U}}$

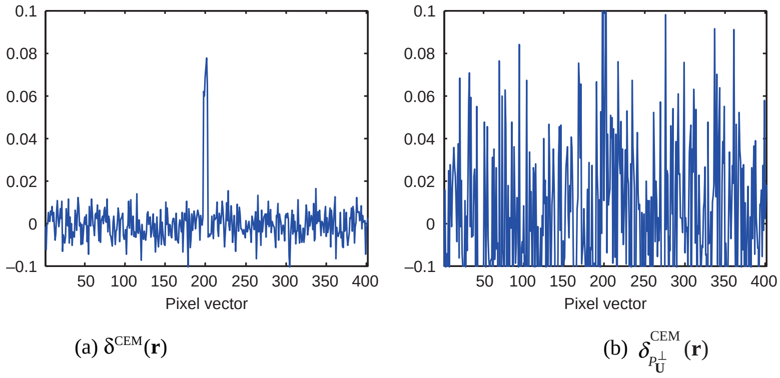
	SNR	198	199	200	201	202	LSE
$\delta^{\text{CEM}}(\mathbf{r})$	30:1	0.0614	0.0597	0.0668	0.0771	0.0633	0.0060,883
$\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$	30:1	0.0630	0.0606	0.0671	0.0782	0.0651	0.0056,946



(a) $\delta^{\text{CEM}}(\mathbf{r})$

(b) $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$

Figure 12.7 Detection results of $\delta^{\text{CEM}}(\mathbf{r})$ and $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ with dry grass and red soil annihilated by $P^{\perp}_{\mathbf{U}}$.



(a) $\delta^{\text{CEM}}(\mathbf{r})$

(b) $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$

Figure 12.8 Detection results of $\delta^{\text{CEM}}(\mathbf{r})$ and $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ with drygrass, redsoil, and sagebrush annihilated by $P^{\perp}_{\mathbf{U}}$.

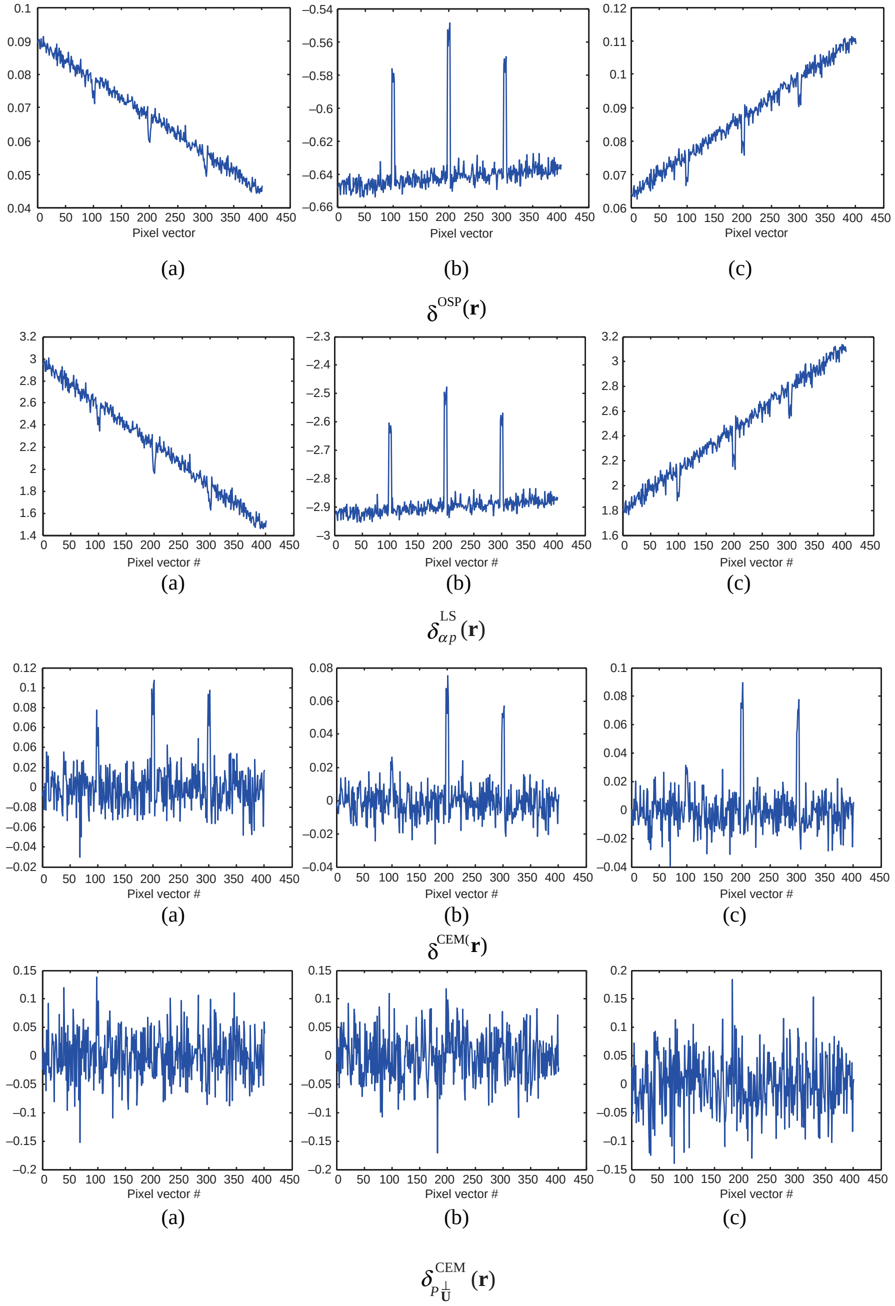
As we can see clearly from Figure 12.8, $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ with the annihilation of **U** performed poorly compared to its counterpart $\delta^{\text{CEM}}(\mathbf{r})$. This was because the signature of the sagebrush was so close to that of creosote leaves as shown in Chang (2000) and Chang (2003a) that the annihilation of sagebrush also eliminated most part of the signature of the creosote leaves, which resulted in significant deterioration of signal detectability of $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$. This example demonstrated a significant difference between annihilation of undesired signatures and suppression of undesired signatures.

EXAMPLE 12.4

(Partial Knowledge)

The same 401 simulated pixel vectors used in Example 12.1 were once again used for the following experiments except that the blackbrush and sagebrush were added to pixel vector numbers 98–102 and pixel numbers 298–302, respectively, at abundance fractions 10% while reducing the abundance of red soil and dry grass by multiplying their abundance fractions by 90%. In this case, there were three target signatures of interest, blackbrush, creosote leaves, and sagebrush with two background signatures, red soil, and dry grass. In this example, the complete knowledge of targets of interest, blackbrush, creosote leaves, and sagebrush was also assumed to be available and the background signatures, soil and dry grass were unknown and considered to be interferers as interference. We also let **d** be the desired target signature and **U** consists of the other two known target signatures. Like Example 12.3, CEM was implemented in conjunction with/without $P^{\perp}_{\mathbf{U}}$, $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ and $\delta^{\text{CEM}}(\mathbf{r})$, respectively. Figure 12.9 shows the detection results of $\delta^{\text{OSP}}(\mathbf{r})$, $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$, $\delta^{\text{CEM}}(\mathbf{r})$, and $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ where figures labeled by (a), (b), and (c) are detection results of blackbrush, creosote leaves and sagebrush, respectively, with the **U** formed by the other two signatures that serve as undesired signatures.

As shown in Chang (2000, 2003a), the three target signatures, blackbrush, creosote leaves, and sagebrush, were very similar. As a consequence, annihilating any two of these three signatures would certainly have tremendous impacts on the detection performance of the third signature. The results of Figures 12.9 confirmed what we expected. That is, detecting one signature also detected the other two signatures. Additionally, it also significantly deteriorated the ability of $\delta^{\text{LS}}_{\alpha_p}(\mathbf{r})$ in estimating abundance fractions as shown in Figure 12.9 where the estimated abundance fractions of each of the three signatures were far more being accurate as tabulated in Table 12.4.



(a) (b) (c)

$\delta^{\text{OSP}}(\mathbf{r})$

(a) (b) (c)

$\delta_{\alpha p}^{\text{LS}}(\mathbf{r})$

(a) (b) (c)

$\delta^{\text{CEM(}}\mathbf{r})$

(a) (b) (c)

$\delta_{P_{\mathbf{U}}^{\perp}}^{\text{CEM}}(\mathbf{r})$

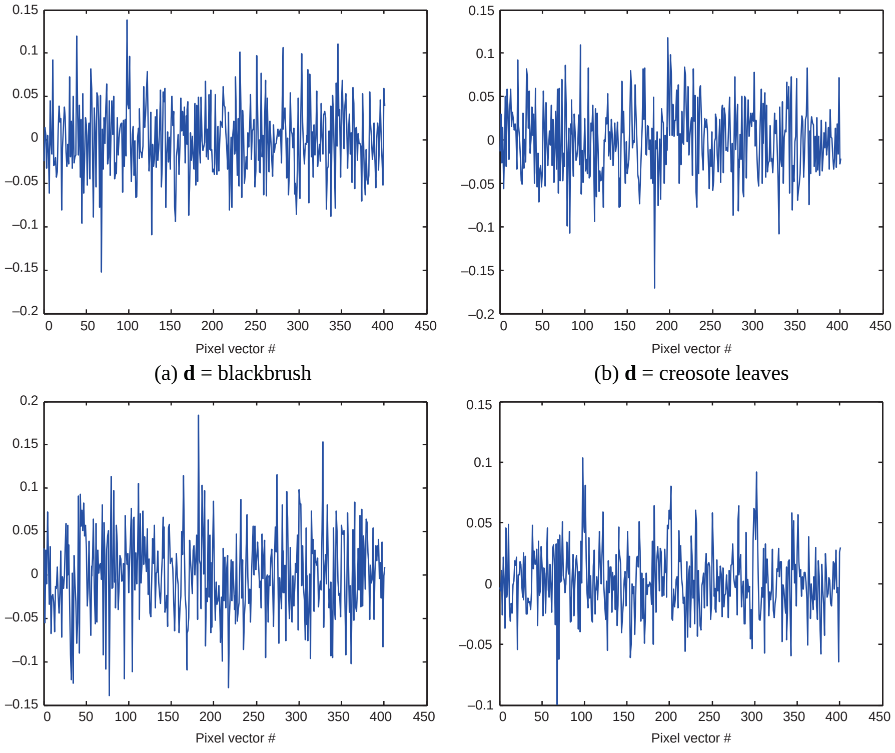
Figure 12.9 Detection results of $\delta^{\text{OSP}}(\mathbf{r})$, $\delta_{\alpha_p}^{\text{LS}}(\mathbf{r})$, $\delta^{\text{CEM}}(\mathbf{r})$, and $\delta_{P_{\mathbf{U}}^{\perp}}^{\text{CEM}}(\mathbf{r})$ with dry grass, red soil, and sagebrush annihilated by $P_{\mathbf{U}}^{\perp}$.

Table 12.4 Comparison of detected abundance fractions by $\delta^{\mathrm{OSP}}(\mathbf{r})$, $\delta^{\mathrm{LS}}_{\alpha_p}(\mathbf{r})$, $\delta^{\mathrm{CEM}}(\mathbf{r})$, and $\delta^{\mathrm{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ with $P^{\perp}_{\mathbf{U}}$-annihilation

	198	199	200	201	202	LSE
$\delta^{\mathrm{OSP}}(\mathbf{r})$	−0.5534	−0.5613	−0.5560	−0.5529	−0.5496	12.143
$\delta^{\mathrm{LS}}_{\alpha_p}(\mathbf{r})$	−12.5008	−12.5366	−12.5124	−12.4986	−12.4837	33.9685
$\delta^{\mathrm{CEM}}(\mathbf{r})$	0.0676	0.0618	0.0551	0.0757	0.0626	0.0065,128
$\delta^{\mathrm{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$	0.1171	0.0678	0.0177	0.0912	0.0903	0.0082,764

Surprisingly, the detection of creosote leaves was quite different from that of blackbrush and sagebrush, as $\delta^{\mathrm{OSP}}(\mathbf{r})$ and $\delta^{\mathrm{LS}}_{\alpha_p}(\mathbf{r})$ were implemented. The detection of creosote leaves also detected significant amounts of blackbrush and sagebrush signatures by suppressing the background signatures even if it was not supposed to do so. The detection of blackbrush and sagebrush showed very similar results and also detected visible amounts of the three signatures except that different amounts of abundance fractions of the background signatures, drygrass, and redsoil were detected. Similar phenomena were also observed from the detection results of $\delta^{\mathrm{CEM}}(\mathbf{r})$ and $\delta^{\mathrm{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ where detection of one signature also picked up the other signatures. Because the spectra of these three signatures were very similar, $P^{\perp}_{\mathbf{U}}$ used in $\delta^{\mathrm{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ also annihilated part of the desired signature before the detection of the desired signature. Consequently, $\delta^{\mathrm{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ performed poorly compared to $\delta^{\mathrm{CEM}}(\mathbf{r})$.

Since the soil and dry grass were used as interference, TCIMF was implemented in two scenarios. One was with $\mathbf{d}=$ a single desired target signature and $\mathbf{U}=$ [drygrass redsoil], and the other was with $\mathbf{D}=$ [blackbrush, creosote leaves, sagebrush] and $\mathbf{U}=$ [drygrass redsoil]. Figure 12.10 shows their



(a) **d** = blackbrush

(b) **d** = creosote leaves

(c) **d** = sagebrush

(d) **D** = [B, C, S]

Figure 12.10 Detection results of $\delta^{\mathrm{TCIMF}}(\mathbf{r})$ with $\mathbf{D}=$ [blackbrush (B), creosote leaves (C), sagebrush (S)].

detection results of $\delta^{\text{TCIMF}}(\mathbf{r})$ where Figure 12.10(a)–(c) was detection results of blackbrush, creosote leaves, and sagebrush, respectively, and Figure 12.10(d) was the simultaneous detection result of the three signatures blackbrush, creosote leaves and sagebrush with $\mathbf{D} = $ [blackbrush (B), creosote leaves (C), sagebrush (S)].

As indicated previously, when TCIMF was implemented with a single target signature designated as the desired signature $\mathbf{D} = \mathbf{d}$, it would perform as it were $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$. This was verified by comparing the results of $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$ in Figure 12.9 to $\delta^{\text{TCIMF}}(\mathbf{r})$ in Figure 12.10(a)–(c). Interestingly, it was not true as shown in Figure 12.10(d) when $\delta^{\text{TCIMF}}(\mathbf{r})$ was implemented with $\mathbf{D} = $ [blackbrush, creosote leaves, sagebrush] and $\mathbf{U} = $ [dry grass, red soil] where it performed significantly better than $\delta^{\text{CEM}}_{P^{\perp}_{\mathbf{U}}}(\mathbf{r})$.

12.5 OSP Implemented Without Knowledge

As OSP was originally developed in Harsanyi and Chang (1994), it required complete endmember knowledge about the image data. Unfortunately, such requirement is seldom satisfied in reality. In order to resolve this issue, two approaches are developed previously. One is to generate desired complete knowledge directly from the image data in an unsupervised means so that the obtained unsupervised knowledge can be used as if it was provided *a priori* (Chang, 2003a, Chang et al., 2001) to make (**d,U**)-model applicable where the undesired target signature projector $P^{\perp}_{\mathbf{U}}$ can be constructed from the generated **U**. Due to the fact that such generated unsupervised knowledge may not be accurate or reliable, an alternative approach is to implement OSP without appealing for unsupervised knowledge. One such approach is CEM described in Section 12.4.1 where only the desired target knowledge, **d**, is required. Instead of trying to find unknown signatures in **U** for annihilation, CEM suppresses all signatures other than the signature of interest. To accomplish that, CEM makes use of the inverse of the sample correlation matrix, $\mathbf{R}^{-1}$ to approximate the complete knowledge provided by $P^{\perp}_{\mathbf{U}}$ in OSP. As a result, OSP and CEM can be related by (12.72)–(12.74) using TCIMF to bridge the gap. Another approach is to implement OSP without prior knowledge. As noted, (**d,U**)-model requires the knowledge of the desired target signature **d** and the undesired target signature matrix **U**. When both the **d** and the **U** are not available, OSP must be implemented whatever it can obtain directly from the data. According to (12.72) and (12.73), when the knowledge about the **U** is not available, the sample spectral correlation matrix $\mathbf{R}^{-1}$ can be used to approximate $P^{\perp}_{\mathbf{U}}$. Moreover, if the knowledge of the **d** is further not provided, the only available information that can be used for OSP is the image pixel vector **r**. In this case, the matched signatures used in OSP must be replaced by the **d**. So, substituting **r** and $\mathbf{R}^{-1}$ for **d** and $P^{\perp}_{\mathbf{U}}$ in (12.9), respectively, results in a new filter that can be used for anomaly detection. Such a filter is referred to as OSP anomaly detector (OSPAD), denoted by $\delta^{\text{OSPAD}}(\mathbf{r})$ and given by

$$\delta^{\text{OSPAD}}(\mathbf{r}) = \mathbf{r}^{T}\mathbf{R}^{-1}\mathbf{r} \tag{12.75}$$

It should be noted that if we replace the **d** in CEM in (12.52) with the image pixel **r**, the resulting form would be the constant 1 for all the image pixel **r**, that is, $\left(\mathbf{r}^{T}\mathbf{R}^{-1}\mathbf{r}\right)^{-1}\mathbf{r}^{T}\mathbf{R}^{-1}\mathbf{r} = 1$. This is because CEM performs as an estimator rather than a detector as OSP does. Since no desired signature **d** needs to be estimated, the quantity of $\left(\mathbf{r}^{T}\mathbf{R}^{-1}\mathbf{r}\right)^{-1}$ that is included in CEM to account for the estimation accuracy varies with the image pixel **r**. Therefore, CEM cannot be used to derive for anomaly detection as we did for OSP in (12.75). That also explains why OSP has better generalization properties than CEM and CEM can be considered as partial-knowledge version of OSP.

Interestingly, if we replace $\mathbf{r}$ and $\mathbf{R}^{-1}$ in $\delta^{\text{OSPAD}}(\mathbf{r})$ with $\mathbf{r} - \boldsymbol{\mu}$ and $\mathbf{K}^{-1}$ where $\boldsymbol{\mu}$ and $\mathbf{K}$ are the sample mean and the sample covariance matrix, the resulting filter $\delta^{\text{OSPAD}}(\mathbf{r})$ turns out to be the well-known anomaly detector, referred to as RX detector (RXD), $\delta^{\text{RXD}}(\mathbf{r})$ developed by Reed and Yu (1990) and also known as Mahalanobis distance (Fukunaga, 1990):

$$\delta^{\text{RXD}}(\mathbf{r}) = (\mathbf{r} - \boldsymbol{\mu})^T \mathbf{K}^{-1} (\mathbf{r} - \boldsymbol{\mu}) \tag{12.76}$$

If we once again replace the matched signature $\mathbf{r}^T$ in (12.75) and (12.76) with the L-dimensional unity vector $\mathbf{1}^T = (\underbrace{1, 1, \ldots, 1}_{L})$, $\delta_{\text{OSPAD}}(\mathbf{r})$ becomes so-called low probability detection (LPD), $\delta^{\text{LPD}}(\mathbf{r})$ in (Harsanyi, 1993; Wang and Chang, 2004) given by

$$\delta^{\text{LPD}}(\mathbf{r}) = \mathbf{1}^T \mathbf{R}^{-1} \mathbf{r} \tag{12.77}$$

that was developed in Harsanyi's dissertation (Harsanyi, 1993) and uniform target detector $\delta^{\text{UTD}}(\mathbf{r})$:

$$\delta^{\text{UTD}}(\mathbf{r}) = (\mathbf{1} - \boldsymbol{\mu})^T \mathbf{K}^{-1} (\mathbf{1} - \boldsymbol{\mu}) \tag{12.78}$$

that was derived in Chang (2003a). More details about $\delta^{\text{LPD}}(\mathbf{r})$ and $\delta^{\text{UTD}}(\mathbf{r})$ can be found in Chang (2003a) and Wang and Chang (2004).

As discussed in Section 12.4, we may sometimes have partial knowledge about target signatures that are not wanted, such as background. In this case, we may think that removing this knowledge prior to anomaly detection could improve anomaly detectability. As will be explained later, this is not necessarily true.

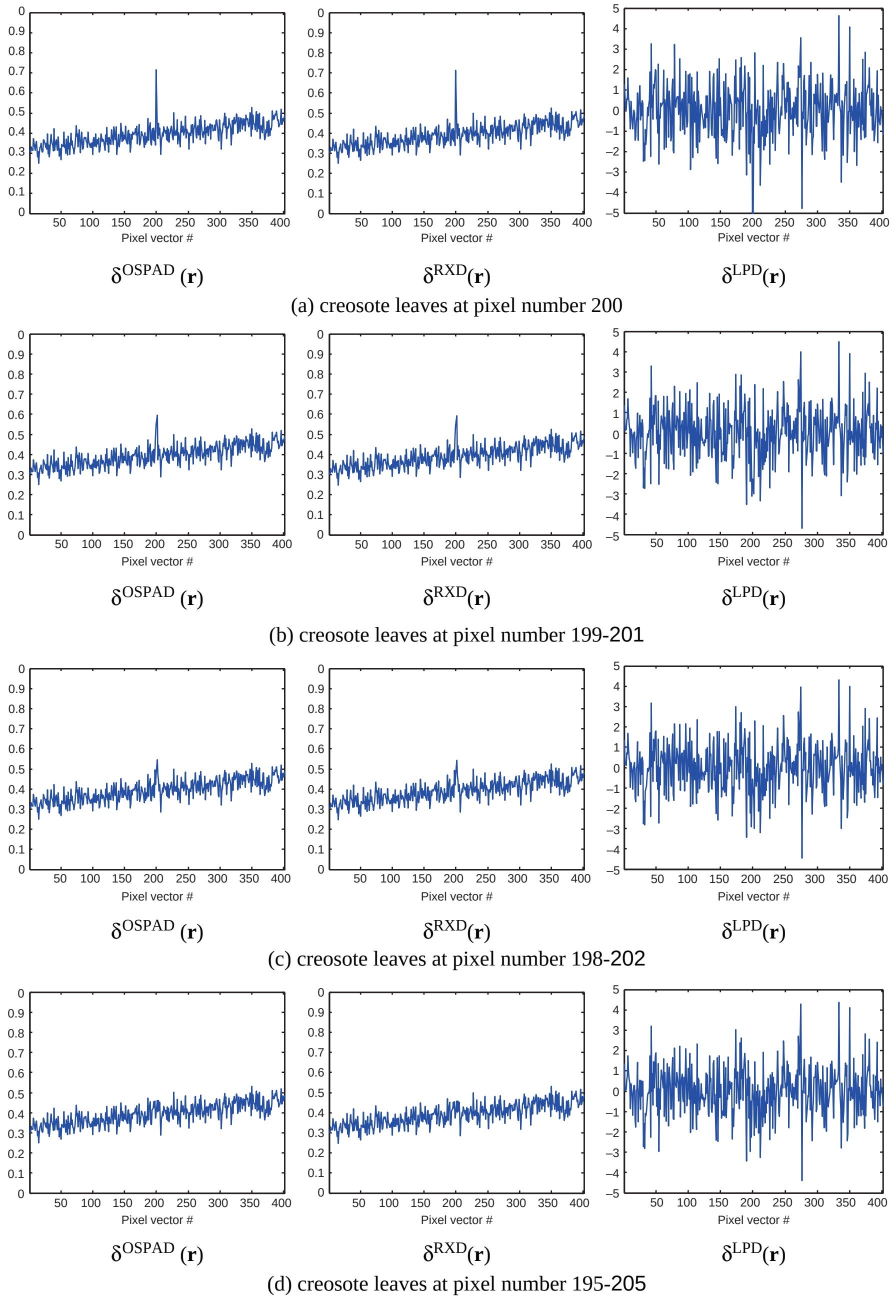
Following a similar treatment in Section 12.4, suppose that the knowledge about $\mathbf{U}$ is provided. We can implement $\delta^{\text{OSPAD}}(\mathbf{r})$ in conjunction with the undesired signature annihilator, $P_{\mathbf{U}}^{\perp}$ in the same way that it is implemented in $\delta^{\text{OSP}}(\mathbf{r})$ to remove the undesired target signatures before anomaly detection. The resulting detector is called $P_{\mathbf{U}}^{\perp}$-OSP anomaly detector ($P_{\mathbf{U}}^{\perp}$-OSPAD), $\delta_{P_{\mathbf{U}}^{\perp}}^{\text{OSPAD}}(\mathbf{r})$ defined by

$$\delta_{P_{\mathbf{U}}^{\perp}}^{\text{OSPAD}}(\mathbf{r}) = \left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right)^T \mathbf{R}^{-1} \left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right) \tag{12.79}$$

Similarly, RXD can be also implemented in conjunction with $P_{\mathbf{U}}^{\perp}$, called $P_{\mathbf{U}}^{\perp}$-RXD and denoted by $\delta_{P_{\mathbf{U}}^{\perp}}^{\text{RXD}}(\mathbf{r})$ as follows:

$$\delta_{P_{\mathbf{U}}^{\perp}}^{\text{RXD}}(\mathbf{r}) = \left(P_{\mathbf{U}}^{\perp}\mathbf{r} - P_{\mathbf{U}}^{\perp}\boldsymbol{\mu}\right)^T \mathbf{K}^{-1} \left(P_{\mathbf{U}}^{\perp}\mathbf{r} - P_{\mathbf{U}}^{\perp}\boldsymbol{\mu}\right) \tag{12.80}$$

Surprisingly, according to the conducted experiments, $\delta^{\text{OSPAD}}(\mathbf{r})$ and $\delta^{\text{RXD}}(\mathbf{r})$ will be shown to perform very closely regardless of whether or not $P_{\mathbf{U}}^{\perp}$ is included in detection. This is because anomaly detectors are generally designed to extract pixels whose signatures spectrally distinct from their



(a) creosote leaves at pixel number 200

(b) creosote leaves at pixel number 199-201

(c) creosote leaves at pixel number 198-202

(d) creosote leaves at pixel number 195-205

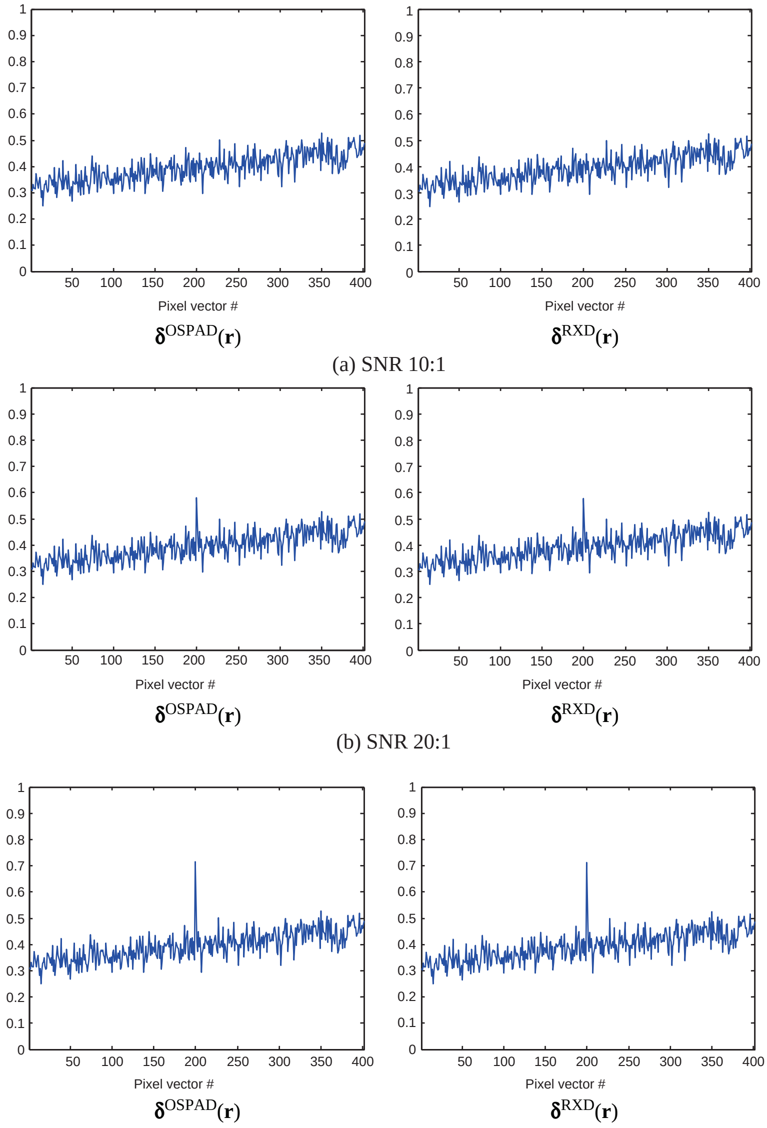
Figure 12.11 Detection results of $\delta^{\text{OSPAD}}(\text{r})$, $\delta^{\text{RXD}}(\text{r})$, $\delta^{\text{LPD}}(\text{r})$ with SNR 30:1 and abundance fraction 10%.

surroundings rather than suppress signatures as OSP does. Another reason is that since $\mathbf{R}^{-1}$ can be viewed as an approximation of $P_{\mathbf{U}}^{\perp}$, an additional inclusion of $P_{\mathbf{U}}^{\perp}$ does not improve the performance of $\delta^{\text{OSPAD}}(\mathbf{r})$ and $\delta^{\text{RXD}}(\mathbf{r})$, both of which already perform a similar task to $P_{\mathbf{U}}^{\perp}$ that is carried out by $\mathbf{R}^{-1}$ in (12.79)–(12.80).

EXAMPLE 12.5

(Anomaly Detection)

In this example, several experiments were conducted to evaluate $\delta^{\text{OSPAD}}(\mathbf{r})$ specified by (12.75), $\delta^{\text{RXD}}(\mathbf{r})$ specified by (12.76) and $\delta^{\text{LPD}}(\mathbf{r})$ specified by (12.77). The same 401 simulated pixel vectors with added SNR 30:1 white Gaussian noise in Example 12.1 were used to detect the creosote leaves as an anomalous target.



$\delta^{\text{OSPAD}}(\mathbf{r})$ $\delta^{\text{RXD}}(\mathbf{r})$

(a) SNR 10:1

$\delta^{\text{OSPAD}}(\mathbf{r})$ $\delta^{\text{RXD}}(\mathbf{r})$

(b) SNR 20:1

$\delta^{\text{OSPAD}}(\mathbf{r})$ $\delta^{\text{RXD}}(\mathbf{r})$

(c) SNR 30:1

Figure 12.12 Detection results of creosote leaves at pixel number 200 by $\delta^{\text{OSPAD}}(\mathbf{r})$ and $\delta^{\text{RXD}}(\mathbf{r})$ with various SNRs.

Figure 12.11(a)–(d) shows the detection results of $\delta^{\text{OSPAD}}(\mathbf{r})$, $\delta^{\text{RXD}}(\mathbf{r})$, and $\delta^{\text{LPD}}(\mathbf{r})$ when the pixels of the creosote leaves with abundance 10% were expanded from one pixel (pixel number 200), three pixels (pixel numbers 199, 200, 201), five pixels (pixel numbers, 198–202) to 11 pixels (pixel numbers 195–205). As we can see from Figure 12.11(a)–(d), the performance of anomaly detection in creosote leaves was deteriorated as the number of creosote leaves was increased.

As a matter of fact, according to our experiments, in order for the creosote leaves to qualify as an anomalous target, the number of pixels should not exceed 3. Figure 12.11 also shows that $\delta^{\text{LPD}}(\mathbf{r})$ could not be used to detect anomalies, but only for background detection. Table 12.5 also tabulates their respective detected abundance fractions where $\delta^{\text{OSPAD}}(\mathbf{r})$ performed slightly better than $\delta^{\text{RXD}}(\mathbf{r})$.

Figures 12.12 and 12.13 also show how SNR (10:1, 20:1, and 30:1) and abundance fractions (10%, 20%, and 30%) affected the performance of anomaly detection for $\delta^{\text{OSPAD}}(\mathbf{r})$ and $\delta^{\text{RXD}}(\mathbf{r})$, respectively.

As shown in Figures 12.12 and 12.13, the higher the SNR, the better the anomaly detection, and the more the abundance fractions of anomaly, the better the anomaly detection. Additionally, all these three figures (i.e., Figs. 12.11–12.13) demonstrated that both $\delta^{\text{OSPAD}}(\text{r})$ and $\delta^{\text{RXD}}(\text{r})$ performed comparably in terms of detected abundance fractions.

The next experiment was designed to see how many anomalies could be detected as distinct targets by $\delta^{\text{OSPAD}}(\mathbf{r})$ and $\delta^{\text{RXD}}(\mathbf{r})$ if the same 401 simulated pixels with added SNR 30:1 white Gaussian noise in Example 12.1 were also used. Figure 12.14 shows the detection results of $\delta^{\text{OSPAD}}(\mathbf{r})$ and $\delta^{\text{RXD}}(\mathbf{r})$ where two of three target signatures, blackbrush, creosote leaves, and sagebrush were selected with same abundance 10% at pixel number 100 and pixel number 300. Table 12.6 tabulates the detected abundance fractions of $\delta^{\text{OSPAD}}(\mathbf{r})$ and $\delta^{\text{RXD}}(\mathbf{r})$ in Figure 12.14 for three target signatures, blackbrush, creosote leaves, and sagebrush with same abundance 10% at pixel number 100 and pixel number 300.

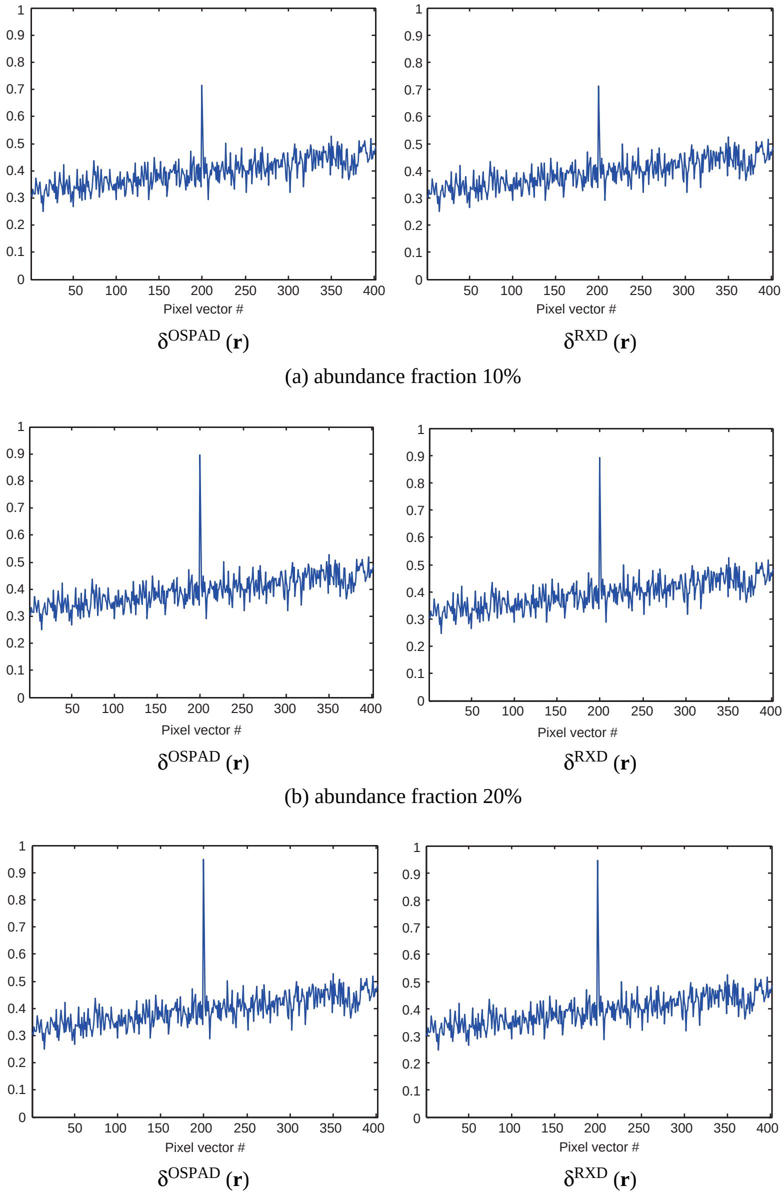
As we can see from Table 12.6, the results were not as good as Figure 12.11(a), but the pixel number 300 was always detected. Interestingly, if the three target signatures, blackbrush, creosote leaves, and sagebrush were present at pixel numbers 100, 200, and 300 with same abundance 10%, Figure 12.15 shows that both $\delta^{\text{OSPAD}}(\mathbf{r})$ and $\delta^{\text{RXD}}(\mathbf{r})$ failed to detect these three target signatures but only the one at pixel number 200.

Table 12.7 tabulates the detection abundance fractions of $\delta^{\text{OSPAD}}(\mathbf{r})$ and $\delta^{\text{RXD}}(\mathbf{r})$ in Figure 12.15 for the three target signatures where the detected amounts of both creosote leaves and sagebrush were close.

However, from a visual inspection of Figure 12.15, only creosote leaves could be detected. This experiment provided evidence that an anomaly could not be simply determined by the detected amount of its abundance fraction. Rather, anomaly detection must be performed by comparing the detected abundance fraction

Table 12.5 Abundance fractions detected by $\delta^{\text{OSPAD}}(\text{r})$ and $\delta^{\text{RXD}}(\text{r})$ in Figure 12.11

	pixel #	197	198	199	200	201	202	203	LSE
δ^{OSPAD}	200				0.714				0.510
	199–201			0.494	0.548	0.594			0.244
	198–202		0.496	0.422	0.473	0.544	0.470		0.246
	195–205	0.425	0.456	0.374	0.396	0.459	0.406	0.460	0.179
δ^{OSP}	200				0.711				0.506
	199–201			0.492	0.546	0.592			0.242
	198–202		0.494	0.419	0.470	0.542	0.468		0.244
	195–205	0.422	0.454	0.371	0.399	0.457	0.403	0.458	0.177
δ^{LPD}	200				−5.609				3782
	199–201			−3.120	−12.260	−3.008			39167
	198-202		−12.921	−1.497	−0.561	−1.471	−3.002		39641
	195-205	−0.482	−12.269	−0.792	0.343	−0.547	−12.181	1.443	0.459



$\delta^{\mathrm{OSPAD}}(\mathbf{r})$ $\delta^{\mathrm{RXD}}(\mathbf{r})$

(a) abundance fraction 10%

$\delta^{\mathrm{OSPAD}}(\mathbf{r})$ $\delta^{\mathrm{RXD}}(\mathbf{r})$

(b) abundance fraction 20%

$\delta^{\mathrm{OSPAD}}(\mathbf{r})$ $\delta^{\mathrm{RXD}}(\mathbf{r})$

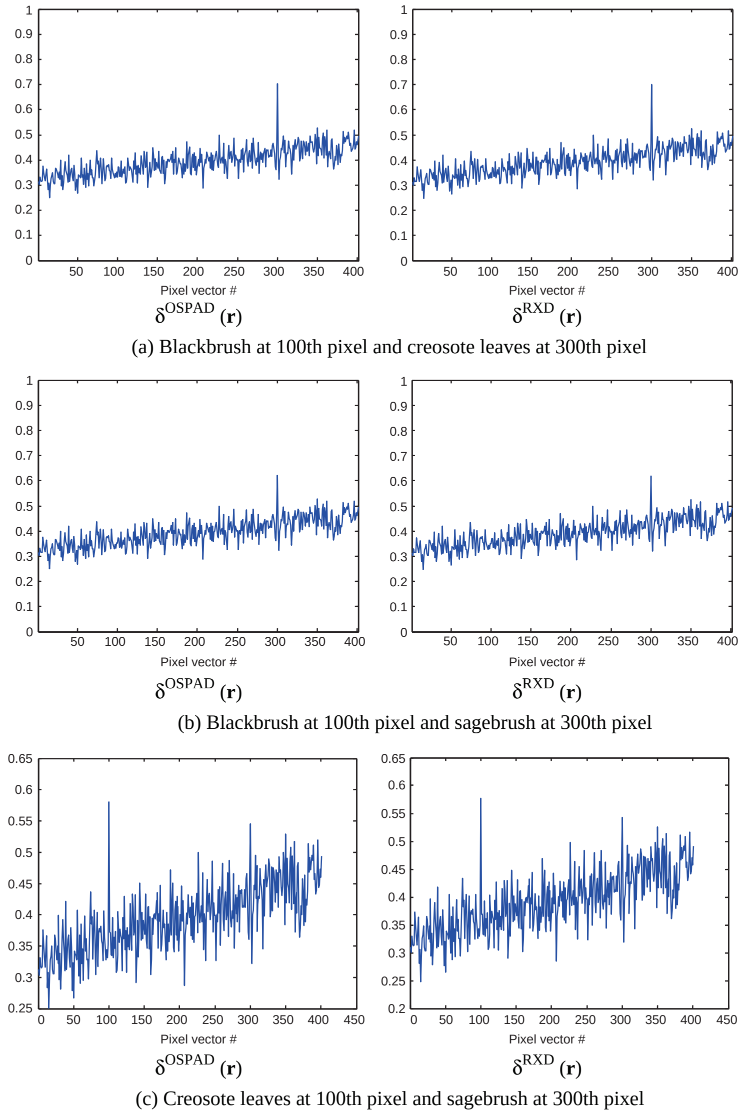
(c) abundance fraction 30%

Figure 12.13 Detection results of creosote leaves at pixel number 200 by $\delta^{\mathrm{OSPAD}}(\mathbf{r})$ and $\delta^{\mathrm{RXD}}(\mathbf{r})$ with various abundance fractions.

of an anomaly against the abundance fractions detected in its surrounding neighborhood. The investigation of this issue was discussed in Hsueh and Chang (2004) and Hsueh (2004) and will be investigated in Chang (2013). Figure 12.15 and Table 12.7 demonstrate that anomaly detection could not be blindly implemented without some extra care.

Table 12.6 Abundance fractions of two signatures detected by $\delta^{\mathrm{OSPAD}}(\mathbf{r})$ and $\delta^{\mathrm{RXD}}(\mathbf{r})$

		Blackbrush	Creosote leaves	Sagebrush
δ^{OSPAD}	Figure 12.14(a)	0.3319	0.7020	
	Figure 12.14(b)	0.3523		0.6220
	Figure 12.14(c)		0.5800	0.5452
δ^{OSP}	Figure 12.14(a)	0.3296	0.6995	
	Figure 12.14(b)	0.3500		0.6195
	Figure 12.14(c)		0.5775	0.5428



δ^{OSPAD} $(\mathbf{r})$ δ^{RXD} $(\mathbf{r})$

(a) Blackbrush at 100th pixel and creosote leaves at 300th pixel

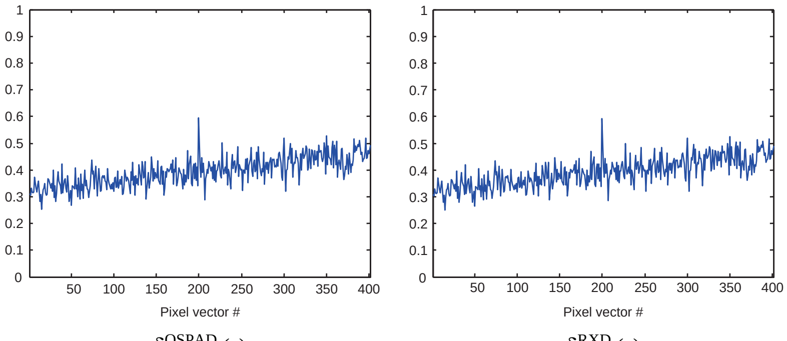
δ^{OSPAD} $(\mathbf{r})$ δ^{RXD} $(\mathbf{r})$

(b) Blackbrush at 100th pixel and sagebrush at 300th pixel

δ^{OSPAD} $(\mathbf{r})$ δ^{RXD} $(\mathbf{r})$

(c) Creosote leaves at 100th pixel and sagebrush at 300th pixel

Figure 12.14 Detection results of $\delta^{\mathrm{OSPAD}}(\mathbf{r})$ and $\delta^{\mathrm{RXD}}(\mathbf{r})$ with various signatures at different pixel numbers.



$\delta^{\text{OSPAD}}(\mathbf{r})$ $\delta^{\text{RXD}}(\mathbf{r})$

Figure 12.15 Detection results of blackbrush at pixel number 100, creosote leaves at pixel number 200, and sagebrush at pixel number 300 by $\delta^{\text{OSPAD}}(\mathbf{r})$ and $\delta^{\text{RXD}}(\mathbf{r})$.

Table 12.7 Abundance fractions of three signatures detected by $\delta^{\text{OSPAD}}(\mathbf{r})$ and $\delta^{\text{RXD}}(\mathbf{r})$ in Figure 12.15

	Blackbrush	Creosote leaves	Sagebrush
δ^{OSPAD}	0.3202	0.5956	0.5204
δ^{RXD}	0.3179	0.5931	0.5179

12.6 Conclusions

OSP has become a standard hyperspectral imaging technique (Schwengerdt, 1997; Chang, 2003a) that can be used in many versatile applications. Despite the fact that various relationships among OSP, CEM, and the RXD have been studied (Chang, 2003a, 2003b; Du et al., 2003), this chapter investigates many interesting issues resulting from OSP that are not explored in Chang (2003a, 2003b) and Du et al. (2003). It revisits OSP from several signal processing perspectives and offers many insights into its design rationales that have not been investigated previously. In particular, it shows that OSP can be derived from various view points of signal detection, linear discriminant analysis, and parameter estimation where the LS OSP is essentially equivalent to LS-LSMA via the proposed OSP-model. It further studies effects of the Gaussian noise and white noise assumptions on the performance of OSP. Finally, it derives various forms of OSP when OSP is provided by different information levels of target knowledge. As shown in this chapter, some well-known and popular filters such as CEM, TCIMF, and the RX anomaly detector can be considered as members of OSP family. Since many experiments conducted based on real hyperspectral images using various forms of OSP have been reported in the literature and can be also found in Chang (2003a), real hyperspectral image experiments are not included here.

13

Fisher's Linear Spectral Mixture Analysis

A commonly used criterion to design techniques for linear spectral mixture analysis (LSMA) is least squares error (LSE) and referred to as least squares (LS)-LSMA. It is also shown in Chapter 12 that the functional form of a matched filter carried out by unconstrained LS-LSMA is essentially identical to that operated by the orthogonal subspace projection (OSP) approach using signal-to-noise ratio (SNR) as a criterion. Unfortunately, it is also known that both criteria are not necessarily optimal for pattern classification. This chapter presents a new and alternative approach to LSMA, called Fisher's LSMA (FLSMA). It extends the well-known pure-pixel-based (i.e., hard decision-based) Fisher's linear discriminant analysis (FLDA) to perform LSMA. Interestingly, what can be derived for LSMA can also be developed for FLSMA. In particular, two types of constrained approaches to LSMA, target signature-constrained mixed pixel classification (TSCMPC) and target abundance-constrained mixed pixel classification (TACMPC) derived in Chang (2002b) and Chang (2003a), can also be developed in parallel for FLSMA, to be called feature vector constrained FLSMA (FVC-FLSMA) and abundance-constrained FLSMA (AC-FLSMA), respectively. Since Fisher's ratio used by FLSMA is a more appropriate criterion than LSE and SNR in classification, both FVC-FLSMA and AC-FLSMA can improve LS-LSMA and SNR-based OSP in mixed pixel classification and abundance fraction quantification.

13.1 Introduction

LSMA has been widely used in subpixel analysis and mixed pixel classification. Many algorithms have been developed for LSMA such as LS-LSMA, SNR-based OSP, and Mahalanobis distance-based Gaussian maximum likelihood estimation (GMLE). However, according to Juang and Katagiri (1992), LSE is not necessarily the best criterion to measure classification error and neither is SNR. Instead, FLDA is one of the major techniques widely used in pattern classification (Duda and Hart, 1973). It makes use of the so-called Fisher's ratio also known as Rayleigh quotient, which is the ratio of between-class scatter matrix to within-class scatter matrix, as a criterion to generate a set of feature vectors that constitute a feature space for better classification. A similar approach to FLDA was developed by Soltanian-Zadeh et al. (1996) who replaced Fisher's ratio with the ratio of interdistance to intradistance and aligned the generated feature vectors along mutual orthogonal directions. This approach has been shown to be successful in magnetic

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

resonance (MR) image classification. Most recently, Soltanian-Zadeh et al.'s approach was further extended to linearly constrained discriminant analysis (LCDA) by Du and Chang for hyperspectral image classification to improve LSMA classification (Du and Chang, 2001a; Chang 2003b). Technically speaking, the feature vectors obtained by Soltanian-Zadeh et al. (1996) as well as those by Du and Chang (2001a) are not actually Fisher's feature vectors because Soltanian-Zadeh et al.'s interdistance to intradistance ratio is not Fisher's ratio.

FLDA is a traditional class membership-labeling technique. When it is used as an LSMA technique, it is implemented in a simple and straightforward manner on a pure pixel basis. Consequently, FLDA produces class maps different from fractional abundance maps generated by LSMA-based techniques that are gray-scale images. This chapter revisits FLDA and presents a new approach, to be called FLSMA. It directly extends pure pixel-based FLDA to a mixed pixel-based technique so as to perform subpixel detection and mixed pixel classification. It constrains FLDA in a way that the Fisher ratio-generated feature vectors are aligned along mutual orthogonal directions in the same way that both Soltanian-Zadeh et al.'s approach and LCDA align the feature vectors generated by the interdistance to intradistance ratio. Analogous to other mixed pixel-based techniques, FLSMA also generates fractional abundance maps with gray scales representing abundance fractions of classes to be classified. As discussed in Chang (2002b) and Chang (2003a), there are two types of constrained approaches, called TSCMPC and TACMPC, developed for LSMA. The TSCMPC constrains target signatures of interest along desired directions to derive a linearly constrained minimum variance (LCMV) approach (Chang, 2002b) that includes constrained energy minimization (CEM) as its special case, whereas TACMPC implements abundance sum-to-one constraint (ASC) and abundance non-negativity constraint (ANC) to derive three least squares abundance-constrained LSMA approaches: sum-to-one constrained least squares (SCLS), non-negativity constrained least squares (NCLS), and fully constrained least squares (FCLS). Interestingly, approaches similar to both TSCMPC and TACMPC can also be developed for FLSMA.

One approach is called FVC-FLSMA derived from TSMPC. It replaces the sample correlation matrix used in LCMV with the within-class scatter matrix. In particular, it can be shown that the classifiers derived by both FVC-FLSMA and LCDA are essentially the same. In addition, because FVC-FLSMA uses Fisher's ratio as a classification criterion as opposed to LCMV that uses LSE as a classification measure, FVC-FLSMA generally performs better than LCMV in classification as expected.

The other approach is abundance constrained least squares FLDA (ACLS-FLDA) derived from TACMPC. It is referred to as AC-FLSMA and implements Fisher's ratio to carry out mixed pixel classification while using the least squares error to perform abundance fraction estimation. Accordingly, in analogy with abundance-constrained LSMA (AC-LSMA), there are also three types of AC-FLSMA that can further be derived: abundance sum-to-one constrained least squares FLSMA (ASCLS-FLSMA), abundance non-negativity constrained least squares FLSMA (ANCLS-FLSMA), and abundance fully constrained least squares FLSMA (AFCLS-FLSMA). As will be demonstrated, AC-FLSMA generally performs better than its counterpart, AC-LSMA, to produce more accurate abundance fractions. It should be noted that the AC-FCLS is the same as FCLS developed in Heinz and Chang (2001) used in other chapters. The inclusion of "AC" in front of FCLS and FLSMA is simply to emphasize the constraints imposed on abundance fractions to distinguish from the FVC-FLSMA which imposes constraints on the feature vectors.

13.2 Feature Vector-Constrained FLSMA (FVC-FLSMA)

In this section, we extend FLDA discussed in Section 2.3.1.1 of Chapter 2 to an LSMA technique using Fisher's ratio as an unmixing criterion, referred to as FLSMA. One difficulty in doing so is that the FLDA-generated feature vectors are not endmembers to form a signature matrix **M** for LSMA Instead, they are discriminant vectors that are used to determine decision boundaries among classes. In particular, the number of such FLDA-generated discriminant feature vectors is one lower than the number of endmembers in **M**.

FLDA finds a set of feature vectors via Fisher's ratio or Rayleigh's quotient defined as

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_\mathrm{B} \mathbf{w}}{\mathbf{w}^T \mathbf{S}_\mathrm{W} \mathbf{w}} \tag{2.35}$$

by solving a generalized eigenvalue problem specified by

$$\mathbf{S}_\mathrm{W}^{-1} \mathbf{S}_\mathrm{B} \mathbf{w} = \lambda \mathbf{w} \tag{13.1}$$

where $\mathbf{S}_B$ and $\mathbf{S}_W$ are referred as between-class and within-class scatter matrices, respectively. Due to the fact that the rank of the between-class scatter matrix $\mathbf{S}_B$ is only $p-1$, there are only $p-1$ nonzero eigenvalues associated with (13.1). However, in order to implement LSMA, we need p feature vectors that can be used to form an endmember matrix **M** rather than discriminant vectors generated by (13.1). One way to mitigate this dilemma was proposed by Soltanian-Zadeh et al. (1996) and Du and Chang (2001a) who replaced Fisher's ratio with the ratio of interdistance to intradistance while constraining the class means along orthogonal directions. As a result, the interdistance was shown to be constant so that it could be removed from consideration in (2.35) and (13.1). Accordingly, the criterion of (2.35) is reduced to the within-class scatter matrix $\mathbf{S}_\mathrm{W}$ that describes the scattering variances centered at each of the p class means. As a consequence, there are p signatures that can be used to form an endmember matrix for LSMA as shown in Du and Chang (2001a). Unfortunately, the criterion used in both Soltanian-Zadeh et al. (1996) and Du and Chang (2001a) is not really Fisher's ratio. Therefore, they cannot be considered as FLDA-based approaches. The approach presented here is indeed derived from Fisher's ratio. It is called FVC-FLSMA that directly extends FLDA in a similar way that LCDA was derived in Du and Chang (2001a) except that FVC-FLSMA constrains Fisher ratio-generated feature vectors along mutual orthogonal directions.

To be more precise, let $\mathbf{w}_j$ be the jth feature vector that maximizes Fisher's ratio subject to a constraint that the jth feature vector must be aligned not only with the jth class mean, $\boldsymbol{\mu}_j$, but also orthogonal to other feature vectors, $\{\boldsymbol{\mu}_k\}_{k=1,k\neq j}^{p}$. In other words, the FVC-FLSMA problem must solve for $1 \leq j, \quad k \leq p$,

$$\max_{\mathbf{w}_j} \left\{ \frac{\mathbf{w}_j^T \mathbf{S}_\mathrm{B} \mathbf{w}_j}{\mathbf{w}_j^T \mathbf{S}_\mathrm{W} \mathbf{w}_j} \right\} \text{subject to the constraint } \mathbf{w}_j^T \boldsymbol{\mu}_k = \delta_{jk} \tag{13.2}$$

Using the same derivation in Du and Chang (2001a), the numerator $\mathbf{w}_j^T \mathbf{S}_\mathrm{B} \mathbf{w}_j$ can be further simplified by

$$\begin{aligned} \mathbf{w}_j^T \mathbf{S}_\mathrm{B} \mathbf{w}_j &= \mathbf{w}_j^T \left\lfloor \sum_{k=1}^{p} n_k (\boldsymbol{\mu}_k - \boldsymbol{\mu})(\boldsymbol{\mu}_k - \boldsymbol{\mu})^T \right\rfloor \mathbf{w}_j \\ &= n_j - 2\sum_{k=1}^{p} n_k \delta_{jk} \mathbf{w}_j^T \boldsymbol{\mu} + \sum_{k=1}^{p} n_k \left(\mathbf{w}_j^T \boldsymbol{\mu}\right)\left(\mathbf{w}_j^T \boldsymbol{\mu}\right)^T \end{aligned} \tag{13.3}$$

with $\boldsymbol{\mu}$ being the global mean of the sample training data. Since

$$2\sum_{k=1}^{p} n_k \delta_{jk} \mathbf{w}_j^T \boldsymbol{\mu} = 2\sum_{k=1}^{p} n_k \delta_{jk} \left\lfloor \mathbf{w}_j^T (1/n_t) \sum_{k=1}^{p} n_k \boldsymbol{\mu}_k \right\rfloor = 2n_j^2/n_t \tag{13.4}$$

and

$$\sum\nolimits_{k=1}^{p} n_k\left(\mathbf{w}_j^T\boldsymbol{\mu}\right)\left(\mathbf{w}_j^T\boldsymbol{\mu}\right)^T = \sum\nolimits_{k=1}^{p} n_k\left(n_j/n_t\right)^2 = n_j^2/n_t \tag{13.5}$$

$\mathbf{w}_j^T\mathbf{S}_{\mathrm{B}}\mathbf{w}_j$ in (2.35) can further be reduced to

$$\mathbf{w}_j^T\mathbf{S}_{\mathrm{B}}\mathbf{w}_j = n_j - 2\left(n_j^2/n_t\right) + \left(n_j^2/n_t\right) = n_j - \left(n_j^2/n_t\right) \tag{13.6}$$

which is independent of $\mathbf{w}_j$. As a consequence, the FVC-FLSMA problem specified by (13.2) is equivalent to the one finding $\mathbf{w}_j^{\text{FVC-FLSMA}}$ that satisfies the following constrained optimization problem, for $1 \leq j,k \leq p$:

$$\min_{\mathbf{w}_j}\mathbf{w}_j^T\mathbf{S}_{\mathrm{W}}\mathbf{w}_j \text{ subject to } \mathbf{w}_j^T\boldsymbol{\mu}_k = \delta_{jk} \tag{13.7}$$

In order to solve the above problem, we define a Lagrangian for each $\mathbf{w}_j$ given by

$$J(\mathbf{w}_j) = \mathbf{w}_j^T\mathbf{S}_{\mathrm{W}}\mathbf{w}_j + \sum\nolimits_{k=1}^{p}\lambda_k^l\left(\mathbf{w}_j^T\boldsymbol{\mu}_k - \delta_{jk}\right) \tag{13.8}$$

where $\left\{\lambda_k^j\right\}_{k=1,j=1}^{p,p}$ are Largrange multipliers. Differentiating (13.8) with respect to $\mathbf{w}_l$ yields

$$\left.\frac{\partial J(\mathbf{w}_j)}{\partial\mathbf{w}_j}\right|_{\mathbf{w}_j^{\text{FVC-FLSMA}}} = 2\mathbf{S}_{\mathrm{W}}\mathbf{w}_j^{\text{FLSMA}} + \sum\nolimits_{k=1}^{p}\lambda_k^j\boldsymbol{\mu}_k = 0 \tag{13.9}$$

which results in

$$\begin{aligned} &2\mathbf{S}_{\mathrm{W}}\mathbf{w}_j^{\text{FVC-FLSMA}} + \textstyle\sum_{k=1}^{p}\lambda_k^j\boldsymbol{\mu}_k = 2\mathbf{S}_W\mathbf{w}_j^{\text{FVC-FLSMA}} + \mathbf{M}\lambda^j = 0 \\ &\Rightarrow \mathbf{w}_j^{\text{FVC-FLSMA}} = -(1/2)\mathbf{S}_{\mathrm{W}}^{-1}\mathbf{M}\lambda^j \end{aligned} \tag{13.10}$$

Using the constraint that $\left(\mathbf{w}_j^{\text{FVC}-\text{FLSMA}}\right)^T\boldsymbol{\mu}_j = \delta_{jk}$ for $1 \leq j,k \leq p$, the Largrange multiplier λ^j can be obtained by

$$\lambda^j = -2\left(\mathbf{M}^T\mathbf{S}_{\mathrm{W}}^{-1}\mathbf{M}\right)^{-1}\mathbf{1}_j \tag{13.11}$$

and the jth weight vector $\mathbf{w}_j^{\text{FVC}-\text{FLSMA}}$in (13.11) becomes

$$\mathbf{w}_j^{\text{FVC}-\text{FLSMA}} = \mathbf{S}_{\mathrm{W}}^{-1}\mathbf{M}\left(\mathbf{M}^T\mathbf{S}_{\mathrm{W}}^{-1}\mathbf{M}\right)^{-1}\mathbf{1}_j \tag{13.12}$$

and

$$\left(\mathbf{w}_j^{\text{FVC}-\text{FLSMA}}\right)^T\mathbf{S}_{\mathrm{W}}\left(\mathbf{w}_j^{\text{FVC}-\text{FLSMA}}\right) = \mathbf{1}_j^T\left(\mathbf{M}^T\mathbf{S}_{\mathrm{W}}^{-1}\mathbf{M}\right)^{-1}\mathbf{1}_j = (-1/2)\mathbf{1}_j^T\lambda^j \tag{13.13}$$

where the last equality in (13.13) is obtained by (13.11).

We can further derive a matrix form for all the optimal solutions $\left\{\mathbf{w}_j^{\text{FVC}-\text{FLSMA}}\right\}_{j=1}^{p}$ for (13.7). Let $\mathbf{W}^{\text{FVC-FLSMA}} = \left[\mathbf{w}_1^{\text{FVC-FLSMA}}\mathbf{w}_2^{\text{FVC-FLSMA}}\cdots\mathbf{w}_p^{\text{FVC-FLSMA}}\right]$ and $\boldsymbol{\Gamma} = \left[\lambda^1\lambda^2\cdots\lambda^p\right]$ the constraints in (13.7) can be expressed in the following matrix form:

$$\left(\mathbf{W}^{\text{FVC}-\text{FLSMA}}\right)^T\mathbf{M} = \mathbf{I} \tag{13.14}$$

and (13.10) becomes

$$\mathbf{W}^{\text{FVC}-\text{FLSMA}} = -(1/2)\mathbf{S}_{\text{W}}^{-1}\mathbf{M}\boldsymbol{\Gamma} \tag{13.15}$$

Using (13.15) and the constraint specified by (13.14) we obtain

$$-(1/2)\boldsymbol{\Gamma}^T\mathbf{M}^T\mathbf{S}_{\text{W}}^{-1}\mathbf{M} = \mathbf{I} \Rightarrow \boldsymbol{\Gamma}^T = -2\left(\mathbf{M}^T\mathbf{S}_{\text{W}}^{-1}\mathbf{M}\right)^{-1} \tag{13.16}$$

Substituting (13.16) into (13.15) we obtain the FVC-FLSMA solution in a matrix form given by

$$\mathbf{W}^{\text{FVC}-\text{FLSMA}} = \mathbf{S}_{\text{W}}^{-1}\mathbf{M}\left(\mathbf{M}^T\mathbf{S}_{\text{W}}^{-1}\mathbf{M}\right)^{-T} = \mathbf{S}_{\text{W}}^{-1}\mathbf{M}\left(\mathbf{M}^T\mathbf{S}_{\text{W}}^{-1}\mathbf{M}\right)^{-1} \tag{13.17}$$

where $\mathbf{X}^{-T}$is defined by $\mathbf{X}^{-T} \equiv \left(\mathbf{X}^{-1}\right)^T$. Applying the $\mathbf{W}^{\text{FVC-FLSMA}}$ to a sample vector $\mathbf{r}$, the abundance vector $\boldsymbol{\alpha}^{\text{FVC-FLSMA}}(\mathbf{r})$ associated with $\mathbf{r}$ can be expressed as

$$\boldsymbol{\alpha}^{\text{FVC}-\text{FLSMA}}(\mathbf{r}) = \left(\mathbf{W}^{\text{FVC}-\text{FLSMA}}\right)^T\mathbf{r} = \left(\mathbf{M}^T\mathbf{S}_{\text{W}}^{-1}\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{S}_{\text{W}}^{-1}\mathbf{r} \tag{13.18}$$

A comment is worthwhile. Since the FVC-FLSMA specified by (13.17) performs mixed pixel classification, it produces a fractional abundance image for each of classes for classification. Therefore, the FVC-generated fractional abundance images generally require a threshold method such as ones in Chang (2003a) to calculate classification rates.

13.3 Relationship Between FVC-FLSMA and LCMV, TCIMF, and CEM

Recalling the LCMV in Chang (2002b) and Chang (2003a), its weighting matrix (11.16) in Chang (2003a) or (6) in Chang (2002b) is given by

$$\mathbf{W}^{\text{LCMV}} = \mathbf{R}^{-1}\mathbf{M}\left(\mathbf{M}^T\mathbf{R}^{-1}\mathbf{M}\right)^{-T}\mathbf{C} = \mathbf{R}^{-1}\mathbf{M}\left(\mathbf{M}^T\mathbf{R}^{-1}\mathbf{M}\right)^{-1}\mathbf{C} \tag{13.19}$$

where the matrix $\mathbf{C}$ is the constraint matrix and $\mathbf{R}$ is the data correlation matrix.

Now, let $\mathbf{I}$ be the $p \times p$ identity matrix and express it as

$$\mathbf{I} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{bmatrix}_{p\times p} = \left[\mathbf{1}_1\mathbf{1}_2\cdots\mathbf{1}_p\right]_{p\times p} \tag{13.20}$$

where $\mathbf{1}_l$ is the lth p-dimensional unit vector specified by $\mathbf{1}_l = (0, \ldots, 0, \underbrace{1}_{l}, 0, \ldots, 0)^T$. By means of (13.20), we can rewrite (13.17) as

$$\mathbf{W}^{\text{FVC}-\text{FLSMA}} = \mathbf{S}_{\text{W}}^{-1}\mathbf{M}\left(\mathbf{M}^T\mathbf{S}_{\text{W}}^{-1}\mathbf{M}\right)^{-T} = \mathbf{S}_{\text{W}}^{-1}\mathbf{M}\left(\mathbf{M}^T\mathbf{S}_{\text{W}}^{-1}\mathbf{M}\right)^{-1}\mathbf{I}_{p\times p} \tag{13.21}$$

Comparing (13.21) to the LCMV-generated weighting matrix specified by (13.19), an immediate finding is that the FVC-FLSMA-generated weighting matrix given by (13.21) has the same form as does (13.19) with the within-class scatter matrix $\mathbf{S}_{\text{W}}$ and $\mathbf{I}$ in the FVC-FLSMA corresponding to $\mathbf{R}$ and the constraint matrix $\mathbf{C}$ in the LCMV, respectively, where the constraint matrix $\mathbf{I}$ used in (13.21) is exactly the same p constraints $\mathbf{w}_l^T\boldsymbol{\mu}_j = \delta_{lj}$ used in (13.8). Similarly, when the constraint matrix $\mathbf{I}$ in (13.21) is replaced with a constraint vector, the lth p-dimensional unit vector $\mathbf{1}_l$, the resulting weighting matrix $\mathbf{W}^{\text{FVC-FLSMA}}$ becomes a weighting vector:

$$\mathbf{w}_l^{\text{FVC-FLSMA}} = \mathbf{S}_{\text{W}}^{-1}\mathbf{M}\left(\mathbf{M}\mathbf{S}_{\text{W}}^{-1}\mathbf{M}\right)^{-T}\mathbf{1}_l = \mathbf{S}_{\text{W}}^{-1}\mathbf{M}\left(\mathbf{M}^T\mathbf{S}_{\text{W}}^{-1}\mathbf{M}\right)^{-1}\mathbf{1}_l \tag{13.22}$$

which corresponds to the target-constrained interference-minimized filter (TCIMF) (Chang, 2002b, 2003a) with the within-class scatter matrix $\mathbf{S}_{\text{W}}$ in (13.22) replaced by $\mathbf{R}$. If there is only one desired target signature $\mathbf{d}$ constrained by $\mathbf{d}^T\mathbf{w} = 1$ via (13.11) and (13.12), (13.12) turns out to be the same functional form implemented by the constrained energy minimization (CEM) in Chang (2002b) and Chang (2003a) where the within-class scatter matrix $\mathbf{S}_{\text{W}}$ and $\boldsymbol{\mu}_l$ are replaced by the sample correlation matrix $\mathbf{R}$ and $\mathbf{d}$ used in the CEM.

13.4 Relationship Between FVC-FLSMA and OSP

If we replace $\mathbf{S}_W^{-1}$ in (13.12) with $P_{\mathbf{U}}^{\perp}$ defined in (2.86), then the resulting weight vector $\mathbf{w}_l^{\text{FVC}-\text{FRLSMA}}$ becomes the least squares OSP (LSOSP) in Tu et al. (1997) and Chang (2003a):

$$\mathbf{w}_l^{\text{LSOSP}} = P_{\mathbf{U}}^{\perp}\mathbf{M}\left(\mathbf{M}^T P_{\mathbf{U}}^{\perp}\mathbf{M}\right)^{-1}\mathbf{1}_l \tag{13.23}$$

With this interpretation, FVC-FLSMA can be considered as a discriminant analysis version of OSP. Additionally, $P_{\mathbf{U}}^{\perp}$ is also idempotent. We can define a linear transformation by $\tilde{\mathbf{r}} = P_{\mathbf{U}}^{\perp}\mathbf{r}$ and $\tilde{\mathbf{M}} = P_{\mathbf{U}}^{\perp}\mathbf{M}$ where $\mathbf{r}$ is an image pixel vector. The resulting image with pixel vectors described by $\tilde{\mathbf{r}}$ is called the $P_{\mathbf{U}}^{\perp}$-whitened hyperspectral image. Let $\tilde{\boldsymbol{\mu}}_l$ be the $P_{\mathbf{U}}^{\perp}$-whitened lth class mean defined by $\tilde{\boldsymbol{\mu}}_l = P_{\mathbf{U}}^{\perp}\boldsymbol{\mu}_l$. Equation (13.23) is reduced to

$$\mathbf{w}_l^{\text{LSOSP}} = \tilde{\mathbf{M}}\left(\tilde{\mathbf{M}}^T\tilde{\mathbf{M}}\right)^{-1}\mathbf{1}_l = \left(\tilde{\boldsymbol{\mu}}_l{}^T P_{\tilde{\mathbf{U}}}^{\perp}\tilde{\boldsymbol{\mu}}_l\right)^{-1} P_{\tilde{\mathbf{U}}}^{\perp}\tilde{\boldsymbol{\mu}}_l \tag{13.24}$$

where $\tilde{\mathbf{U}} = \left[\tilde{\boldsymbol{\mu}}_1 \cdots \tilde{\boldsymbol{\mu}}_{l-1}\,\tilde{\boldsymbol{\mu}}_{l+1} \cdots \tilde{\boldsymbol{\mu}}_p\right]$ and $\tilde{\boldsymbol{\mu}}_j = P_{\tilde{\mathbf{U}}}^{\perp}\boldsymbol{\mu}_j$ for $1 \le j \le p$. If we further let $P_{\tilde{\mathbf{U}}}^{\perp} = \mathbf{I}$, (13.24) is reduced to $\mathbf{w}_l^{\text{LSOSP}} = \left(\tilde{\boldsymbol{\mu}}_l^T\tilde{\boldsymbol{\mu}}_l\right)^{-1}\tilde{\boldsymbol{\mu}}_l$, which is exactly the same matched filter used by LSOSP with the matched signature specified by the desired signature $\tilde{\boldsymbol{\mu}}_l$.

13.5 Relationship Between FVC-FLSMA and LCDA

Recently, a constrained linear discriminant analysis approach, called linearly constrained discriminant analysis (LCDA), was developed by Du and Chang (2001a) where the within-class and between-class scatter matrices were replaced by intradistance and interdistance, respectively, and the class means were also aligned with orthogonal directions. As shown in Du and Chang (2001a), LCDA solution has the same equation specified by (13.21). So, LCDA is essentially FVC-FLSMA. Furthermore, the total scatter matrix $\mathbf{S}_{\mathrm{T}}$ is the sum of within-class scatter matrix $\mathbf{S}_{\mathrm{W}}$ and between-class scatter matrix $\mathbf{S}_{\mathrm{B}}$ in (2.35) and is a constant matrix. The problem specified by (13.7) can be further shown to be equivalent to finding $\mathbf{w}_l$ that satisfies

$$\min_{\mathbf{w}_l} \mathbf{w}_l^T \mathbf{S}_{\mathrm{T}} \mathbf{w}_l \text{ subject to } \mathbf{w}_l^T \boldsymbol{\mu}_j = \delta_{lj} \quad \text{for} \quad 1 \leq j \leq p \tag{13.25}$$

The solution to (13.25) can be obtained by $\mathbf{w}_l^* = \mathbf{S}_{\mathrm{T}}^{-1}\mathbf{M}\left(\mathbf{M}^T\mathbf{S}_{\mathrm{T}}^{-1}\mathbf{M}\right)^{-1}\mathbf{1}_l$ that turns out to be the same as (13.12). As shown in Chang (2003a), the total scatter matrix S_{T} is related to the data training sample covariance matrix $\mathbf{K}_t$ by $\mathbf{S}_{\mathrm{T}} = N\mathbf{K}_t$ where N is total number of training samples. Using this fact, the problem specified by (13.25) is also equivalent to the following problem:

$$\min_{\mathbf{w}_l} \mathbf{w}_l^T \mathbf{K}_t \mathbf{w}_l \text{ subject to } \mathbf{w}_l^T \boldsymbol{\mu}_j = \delta_{lj} \quad \text{for} \quad 1 \leq j \leq p. \tag{13.26}$$

The solution to (13.26) is also $\mathbf{w}_l^* = (\mathbf{K}_t)^{-1}\mathbf{M}\left(\mathbf{M}^T(\mathbf{K}_t)^{-1}\mathbf{M}\right)^{-1}\mathbf{1}_l$ that is also similar to (13.12).

13.6 Abundance-Constrained Least Squares FLDA (ACLS-FLDA)

It should be noted that the FVC-FLDA solution (13.12) is not abundance-constrained in the sense that there is no constraint imposed on the abundance vector $\boldsymbol{\alpha}$. Therefore, the FVC-FLDA solution does not guarantee that $\boldsymbol{\alpha} \geq \mathbf{0}$, that is, $\alpha_j \geq 0$ for all $1 \leq j \leq p$. In order to obtain an abundance-constrained FLDA, we first consider the following unconstrained LSE problem with a weighting matrix given by the within-class scatter matrix $\mathbf{S}_{\mathrm{W}}$ that minimizes the LSE:

$$(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})^T \mathbf{S}_{\mathrm{W}}^{-1} (\mathbf{r} - \mathbf{M}\boldsymbol{\alpha}) \text{ over } \boldsymbol{\alpha} \tag{13.27}$$

Using this square-root form, the LSE in (13.27) can be further expressed as

$$\begin{aligned}
&(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})^T \mathbf{S}_{\mathrm{W}}^{-1/2}\mathbf{S}_{\mathrm{W}}^{-1/2}(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha}) = (\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})^T\left(\mathbf{S}_{\mathrm{W}}^{-1/2}\right)^T\mathbf{S}_{\mathrm{W}}^{-1/2}(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha}) \\
&= \left(\mathbf{S}_{\mathrm{W}}^{-1/2}\mathbf{r} - \mathbf{S}_{\mathrm{W}}^{-1/2}\mathbf{M}\boldsymbol{\alpha}\right)^T\left(\mathbf{S}_{\mathrm{W}}^{-1/2}\mathbf{r} - \mathbf{S}_{\mathrm{W}}^{-1/2}\mathbf{M}\boldsymbol{\alpha}\right)
\end{aligned} \tag{13.28}$$

Now, if we let $\tilde{\mathbf{r}} = \mathbf{S}_{\mathrm{W}}^{-1/2}\mathbf{r}$ and $\tilde{\mathbf{M}} = \mathbf{S}_{\mathrm{W}}^{-1/2}\mathbf{M}$, (13.28) can be further reduced to one that minimizes the following unconstrained LSE:

$$\left(\tilde{\mathbf{r}} - \tilde{\mathbf{M}}\boldsymbol{\alpha}\right)^T\left(\tilde{\mathbf{r}} - \tilde{\mathbf{M}}\boldsymbol{\alpha}\right) \text{ over } \boldsymbol{\alpha} \tag{13.29}$$

which is exactly the same least squares mixing problem considered in LSMA. By virtue of (13.29) we can impose $\boldsymbol{\alpha} \geq \mathbf{0}$ or $\sum_{j=1}^{p} \alpha_j = 1$ on (13.27) to obtain the following three types of abundance-constrained least squares FLDA (ACLS-FLDA) problems: sum-to-one constrained least squares (SCLS), non-negativity constrained least squares (NCLS), and fully constrained least squares (FCLS) problems.

(i) Abundance sum-to-one constrained least squares FLDA (ASCLS-FLDA) problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)^T\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \sum\nolimits_{j=1}^{p}\alpha_j=1 \tag{13.30}$$

(ii) Abundance non-negativity-constrained least squares FLDA (ANCLS-FLDA) problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)^T\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \boldsymbol{\alpha}\geq 0 \tag{13.31}$$

(iii) Abundance fully constrained least squares FLDA (AFCLS-FLDA) problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)^T\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \boldsymbol{\alpha}\geq 0 \text{ and } \sum\nolimits_{j=1}^{p}\alpha_j=1 \tag{13.32}$$

The solutions to (13.30)–(13.32) can be obtained exactly by the same methods that solve the SCLS, NCLS, and FCLS mixing problems in Chang (2003a).

13.7 Synthetic Image Experiments

This section conducts two sets of experiments, synthetic image and real image experiments, to demonstrate the utility of FLSMA in mixed pixel classification and quantification. For AC-FLSMA, only AFCLS-FLSMA that is abundance fully constrained FLSMA is conducted to compare its counterpart of LSMA, AC-FCLS which is actually the FCLS developed by Heinz and Chang (2001).

In order to substantiate FLSMA, a synthetic image similar to the real scene in Figure 1.15(a) was simulated. It has the size of 64 × 64 pixel vectors and 20 panels with various sizes arranged in a 5 × 4 matrix and located at the center of the scene shown in Figure 13.1(a). The five panel signatures in Figure 1.16 were used to simulate these 20 panels.

For row i, the panel signature $\mathbf{p}_i$ was used to simulate four panels in each of columns where the panels are a 2 × 2-pixel panel, $\{\mathrm{p}^i_{1,11},\mathrm{p}^i_{1,12},\mathrm{p}^i_{1,21},\mathrm{p}^i_{1,22}\}$ in the first column, a 1 × 2-pixel panel, $\{\mathrm{p}^i_{2,11},\mathrm{p}^i_{2,12}\}$ in the second column, a one-pixel panel, $\mathrm{p}^i_{3,1}$ in the third column, and a one-pixel panel, $\mathrm{p}^i_{4,1}$ in the fourth column. While the pixels in all the 2 × 2-pixel panels and the 1 × 2-pixel panels were pure pixels simulated by 100% panel signature $\mathbf{p}_i$, the

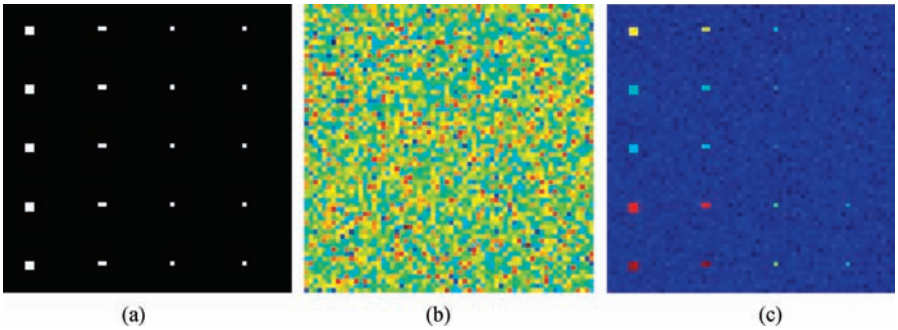


Figure 13.1 A synthetic image, (a) 20 simulated panels; (b) background simulated by a grass signature corrupted by an additive Gaussian noise with SNR 20:1, (c) a synthetic image with the 20 simulated panels in (a) implanted in the background simulated in (b).

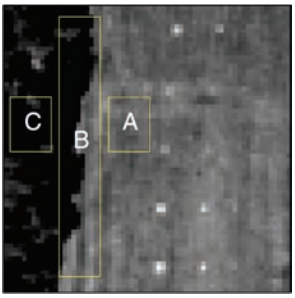


Figure 13.2 HYDICE image scene with training samples marked by areas, A, B, and C.

panels, $p^i_{3,1}$ and $p^i_{4,1}$ were subpixel panels simulated by (50%$\mathbf{p}_i$, 50%$\mathbf{b}_A$) and (25%$\mathbf{p}_i$, 75%$\mathbf{b}_A$) where the background signature, $\mathbf{b}_A$ was a grass signature obtained by averaging all the pixels in the area located at the right and marked by "A" in Figure 13.2.

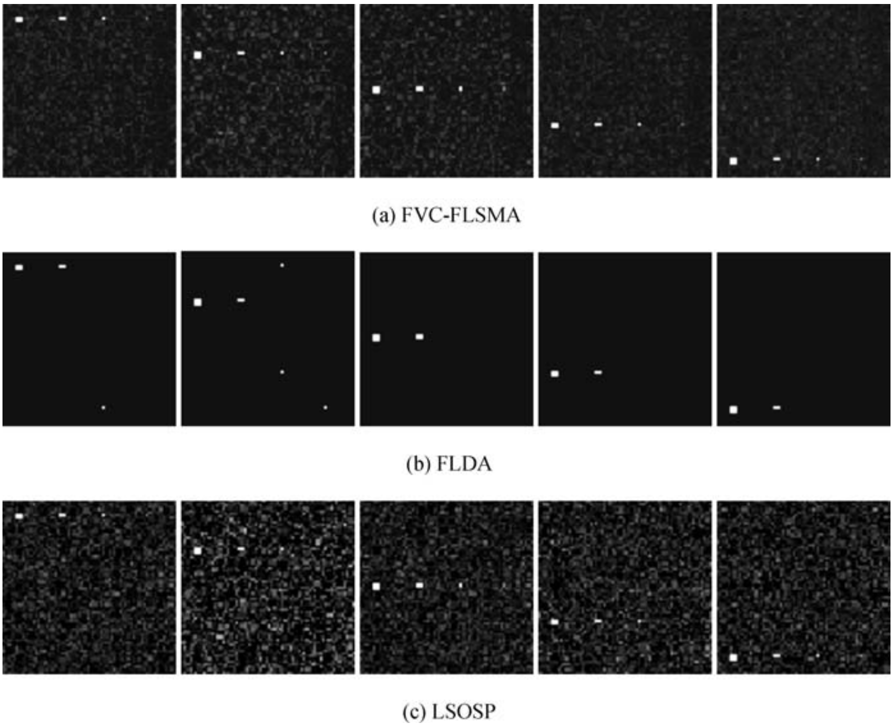
This simulated synthetic image is particularly designed to mimic the image scene in Figure 1.15(a) with the size for comparative analysis. Figure 13.1(b) is the image background simulated by the grass signature $\mathbf{b}_A$ plus an additive white Gaussian noise with the signal-to-noise ratio 20:1 defined in Harsanyi and Chang (1994). Figure 13.1(c) is a synthetic image with these 20 panels in Figure 13.1(a) implanted in the image background in Figure 13.1(b). Three experiments were conducted to compare FLSMA to LSMA in mixed pixel classification and quantification.

EXAMPLE 13.1

(FVC-FLSMA vs. FLDA, LSOSP)

The purpose of this example is to demonstrate the improvement achieved by FVC-FLSMA over the commonly used unconstrained classifiers, FLDA and LSOSP, where FLDA is the best-known classical pure-pixel classifier and LSOSP is a well-known least squares linear spectral mixture analysis. Since both FVC-FLSMA and FLDA required training samples for classification, the set of panels, $\left\{p^i_{1,11}, p^i_{1,12}, p^i_{1,21}, p^i_{1,22}\right\}_{i=1}^{5}$ in the first column in Figure 13.1(a) were used as training data. For LSOSP, the used target signature matrix was formed by $\mathbf{M}_A = [\mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3\ \mathbf{p}_4\ \mathbf{p}_5\ \mathbf{b}_A]$ with the panel signatures $\{\mathbf{p}_i\}_{i=1}^{5}$ in Figure 1.16. Figure 13.3(a)–(c) shows the classification results of the 20 panels in Figure 13.1(c) produced by FVC-FLSMA, FLDA, and LSOSP, respectively.

Since both FVC-FLSMA and LSOSP are mixed pixel classifiers, they produced gray-scale fractional abundance images for each of five panel classes in Figure 13.3(a) and Figure 13.3(c), respectively. On the contrary, FLDA is a pure pixel-based class-labeling classifier. So, the images shown in Figure 13.3(b) are five classification maps, one for each of panel classes. As shown in Figure 13.3(a)–(c), FVC-FLSMA was the best among the three classifiers where all the 20 panels were detected with small detected abundance fractions of other pixel vectors, specifically in detection of panels in row 2. However, if an appropriate threshold value was selected to threshold the gray-scale images, the small abundance fractions detected for other pixel vector would be cleaned out. Compared to FVC-FLSMA, FLDA detected all pure pixel vectors but missed all subpixels. Additionally, it also generated a few falsely alarmed pixels. This shows that FLDA had difficulty with the subpixel detection. Despite the fact that the LSOSP generated more noisy classification images than those produced by the FVC-FLSMA, it still detected most of the 20 panels including the subpixel panels that were not detected by FLDA. This example shows that a pure pixel-based classifier may work well for pure pixels but not subpixels.



(a) FVC-FLSMA

(b) FLDA

(c) LSOSP

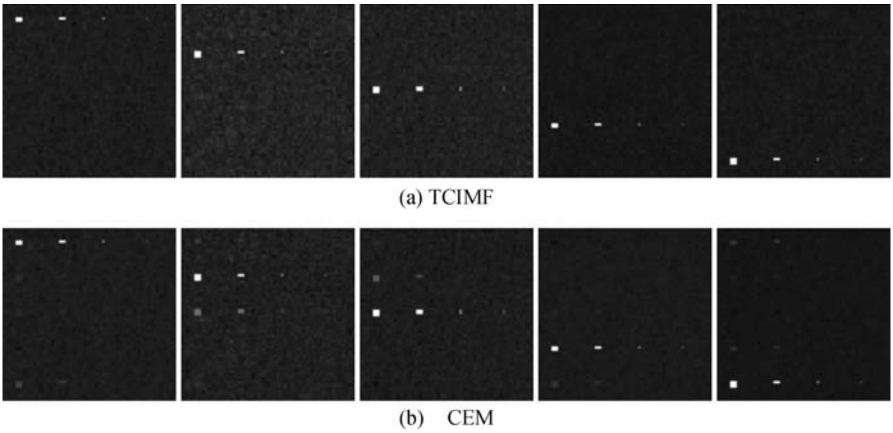
Figure 13.3 Classification results of the 20 panels in Figure 13.1(c) produced by FVC-FLSMA, FLDA, and LSOSP, respectively.

EXAMPLE 13.2

(FVC-FLSMA vs. TCIMF and CEM)

Unlike Example 13.1 that compared FVC-FLSMA to unconstrained classifiers, this example was designed to compare FVC-FLSMA against two target signature-constrained classifiers, TCIMF and CEM (Chang, 2002b, 2003b). The FVC-FLSMA implemented in this example was the same as that in Example 13.1. TCIMF was implemented in a similar manner that LSOSP was implemented in Example 13.1, where TCIMF also used $\mathbf{M}_{\text{A}}$ as its target signature matrices with $p=1$, that is, the desired signature $\mathbf{d}$ was a single target signature. CEM was also implemented by considering each of panel signatures, $\{\mathbf{p}_i\}_{i=1}^{5}$ as a desired target signature $\mathbf{d}$. Figure 13.4(a) and (b) shows the classification results of TCIMF and CEM, respectively, where all the 20 panels were detected with a small number of false alarms for CEM.

Compared to the classification of FVC-FLSMA in Figure 13.3(a), TCIMF performed slightly better than did FVC-FLSMA in terms of more clean background due to TCIMF's ability in noise suppression. Nevertheless, both performed comparably. However, the CEM-generated fractional abundance images had exhibited interfering effects resulting from other panel signatures since CEM could only suppress, but could not eliminate interference caused by signal sources other than the desired signal source. This is particularly evident in detection of the panels in rows 2 and 3 due to the fact that both signatures, $\mathbf{p}_2$ and $\mathbf{p}_3$, are close. Compared to CEM such effects were significantly reduced by TCIMF since TCIMF eliminated rather than suppressed other five signatures as did CEM.


(a) TCIMF

(b) CEM

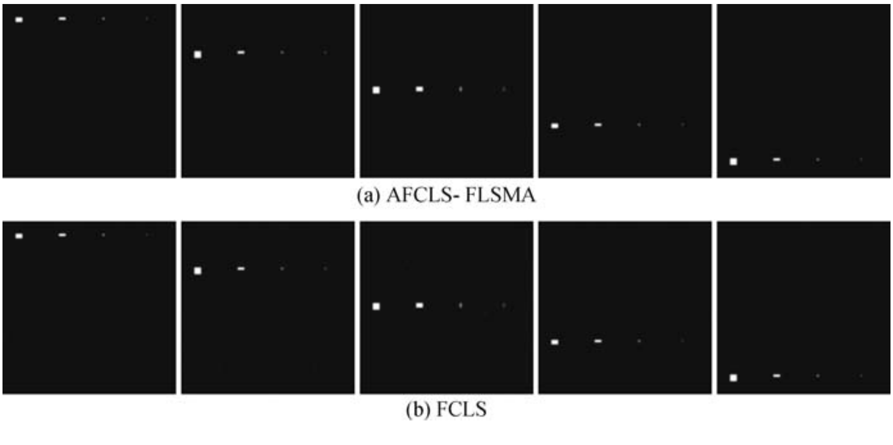
Figure 13.4 Classification results of the 20 panels in Figure 13.1(c) produced by the TCIMF and CEM, respectively.

EXAMPLE 13.3

(AFCLS-FLSMA vs. FCLS)

In Example 13.2, a comparative analysis between FVC-FLSMA and two target signature-constrained classifiers, TCIMF and CEM was performed. This example considers another type of constrained classifiers, which are target abundance-constrained classifiers. FLSMA implemented in this example was AFCLS-FLSMA and compared to the target abundance-constrained classifier, FCLS (Heinz and Chang, 2001; Chang, 2003a). Figure 13.5(a) and (b) shows the classification results of the 20 panels in Figure 13.1(c) produced by the AFCLS-FLSMA and FCLS, respectively.

Surprisingly, comparing Figures 13.5(b) and 13.3(c), FCLS significantly improved LSOSP where all the 20 panels were detected. From visual inspection of Figure 13.5(a) and (b), both AFCLS-FLSMA and FCLS performed very similarly. However, if we tabulate the abundance fractions obtained for images in Figure 13.5


(a) AFCLS- FLSMA

(b) FCLS

Figure 13.5 Classification results of the 20 panels in Figure 13.1(c) produced by the AFCLS-FLSMA and FCLS.

Table 13.1 Quantitative results produced by the AFCLS-FLSMA and FCLS

First row	$p^1_{1,11}$	$p^1_{1,12}$	$p^1_{1,21}$	$p^1_{1,22}$	$p^1_{2,11}$	$p^1_{2,12}$	$p^1_{3,11}$	$p^1_{4,11}$
AFCLS-FLDA	1	1	1	1	1	1	0.503	0.249
FCLS	1	1	1	1	1	1	0.502	0.212
Second row	$p^2_{1,11}$	$p^2_{1,12}$	$p^2_{1,21}$	$p^2_{1,22}$	$p^2_{2,11}$	$p^2_{2,12}$	$p^2_{3,11}$	$p^2_{4,11}$
AFCLS-FLDA	1	1	1	1	1	1	0.483	0.256
FCLS	1	1	1	1	1	1	0.492	0.268
Third row	$p^3_{1,11}$	$p^3_{1,12}$	$p^3_{1,21}$	$p^3_{1,22}$	$p^3_{2,11}$	$p^3_{2,12}$	$p^3_{3,11}$	$p^3_{4,11}$
AFCLS-FLDA	1	1	1	1	1	1	0.494	0.237
FCLS	1	1	1	1	1	1	0.492	0.24
Fourth row	$p^4_{1,11}$	$p^4_{1,12}$	$p^4_{1,21}$	$p^4_{1,22}$	$p^4_{2,11}$	$p^4_{2,12}$	$p^4_{3,11}$	$p^4_{4,11}$
AFCLS-FLDA	1	1	1	1	1	1	0.485	0.247
FCLS	1	1	1	1	1	1	0.498	0.207
Fifth row	$p^5_{1,11}$	$p^5_{1,12}$	$p^5_{1,21}$	$p^5_{1,22}$	$p^5_{2,11}$	$p^5_{2,12}$	$p^5_{3,11}$	$p^5_{4,11}$
AFCLS-FLDA	1	1	1	1	1	1	0.497	0.244
FCLS	1	1	1	1	1	1	0.485	0.242

(a) and (b) in Table 13.1, all the two abundance-constrained classifiers detected 100% for all pure panel pixels, $\left\{p^i_{1,11}, p^i_{1,12}, p^i_{1,21}, p^i_{1,22}\right\}_{i=1}^5$ and $\left\{\mathrm{p}^i_{2,11}, \mathrm{p}^i_{2,12}\right\}_{i=1}^5$ in the first and second columns.

But quantitatively, AFCLS-FLSMA apparently performed better than FCLS in terms of quantifying the abundance fractions of the subpixel panel pixels, $\left\{p^i_{3,11}\right\}_{i=1}^5$ and $\left\{p^i_{4,11}\right\}_{i=1}^5$ in the third and fourth columns.

13.8 Real Image Experiments

In this section, real hyperspectral image experiments were conducted based on the 15-panel HYDICE image scene in Figure 13.2 (see Figure 1.15(a)). One major difference between the real HYDICE image scene in Figure 13.2 and the simulated synthetic image in Figure 13.1(c) is that there is very little knowledge of the image background in Figure 13.2 compared to the image background in Figure 13.1(b) that was simulated by complete knowledge. As expected, FLSMA may not perform as well as it did for the synthetic image if the image background in Figure 13.1(b) was not well characterized. In order to demonstrate this fact, two scenarios are used to characterize the image background as follows. Additionally, we also assume that the knowledge of the 9 R pixels in the 3 m × 3 m and 5 R pixels in 2 m × 2 m panels in Figure 1.15(b) is available *a priori*. So, the panel signatures in Figure 13.2 and 14 R pixels in both the 3 m × 3 m and 2 m × 2 m panels are considered to be prior knowledge according to the ground truth in Figure 1.15(b).

13.8.1 Image Background Characterized by Supervised Knowledge

By viewing the scene in Figure 13.2, a large portion of the image background is made up of one-fourth of a forest on the left and three-fourth of a large grass field. Using this supervised knowledge, we conducted two experiments to represent the image background. One is to use the area B to characterize the image background. In this case, a single background signature $\mathbf{b}_\mathrm{B}$ used for LSOSP was obtained by averaging all the pixels in the area B and the training samples used for FLDA and FVC-FLSMA were all the pixels in the area B for one background class. In this case, we can compare the results of real image experiments to those of the synthetic image experiments. Another is to use the area marked by A and the area marked by C in Figure 13.2 to represent the image background. In this case, two single background signatures $\mathbf{b}_A$ and $\mathbf{b}_C$ used for LSOSP were obtained by averaging all the pixels in the areas A and C separately and the training samples used for FLDA and FVC-FLSMA were all the pixels in the areas A and C for two background classes, forest and grass.

EXAMPLE 13.4

(FVC-FLSMA vs. FLDA, LSOSP, TCIMF, and CEM) and (AFCLS-FLSMA vs. FCLS)

Like Examples 13.1 and 13.2, we implemented FVC-FLSMA, FLDA, LSOSP, TCIMF, and CEM with the image background considered as a single class. So, the training samples used for FVC-FLSMA and FLDA were the R pixels in the 3 m × 3 m and 2m × 2m panels in Figure 13.2 (see Figure 1.15(b)) and background pixels in the area B. For LSOSP and TCIMF with $p = 1$, the used target signature matrix $\mathbf{M}_1$ is formed by $\mathbf{M}_1 = [\mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3\ \mathbf{p}_4\ \mathbf{p}_5\ \mathbf{b}_B]$ where the panel signatures $\{\mathbf{p}_i\}_{i=1}^{5}$ are shown in Figure 1.16. Figure 13.6(a)–(e) shows the classification results of the 15 panels in Figure 13.2 produced by FVC-FLSMA, FLDA, LSOSP, TCIMF, and CEM, respectively, with a single-class background.

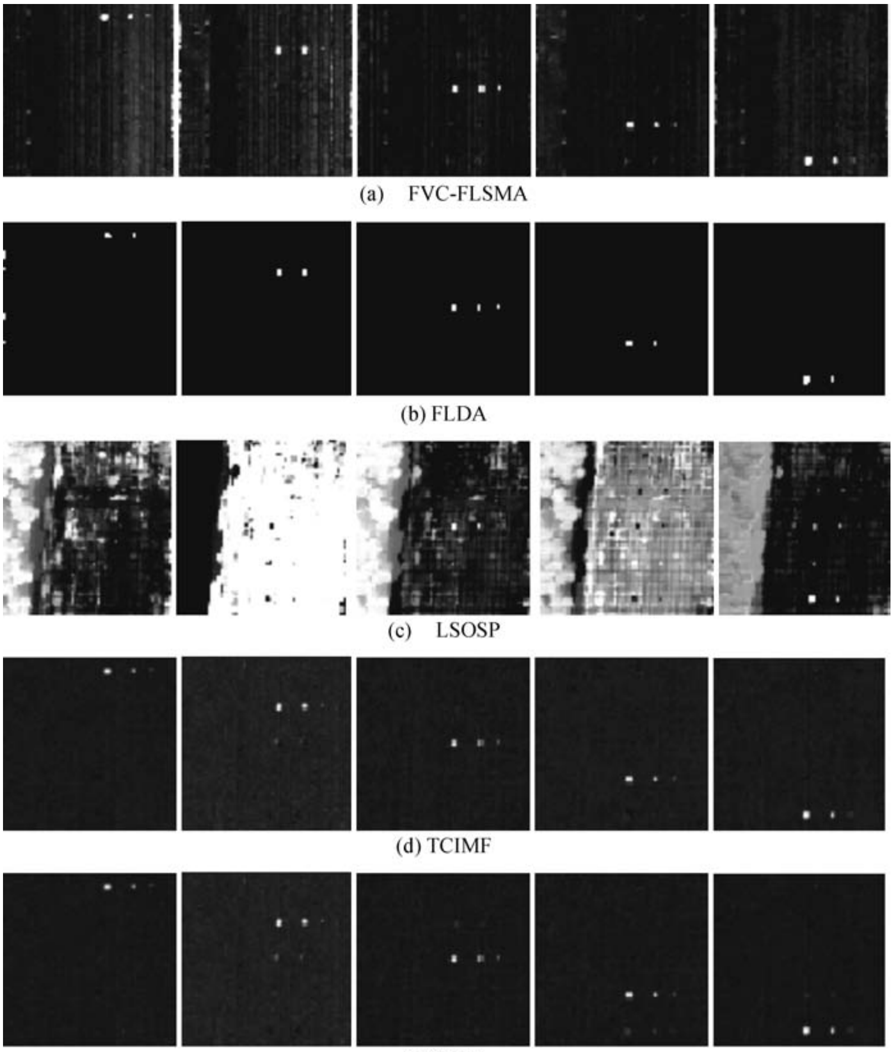
As we can see from Figure 13.6, FLDA performed surprisingly well compared to LSOSP and CEM in detection of pure pixels. For the overall performance, TCIMF seemed to produce the best results in terms of panel pixel detection including the detection of subpixel panels and the LSOSP was the worst.

Figure 13.6 only shows qualitative results for classification. For quantitative analysis AFCLS-FLSMA and FCLS were implemented for comparison. Fig. 13.7 shows their quantification results in gray scales, which are unmixed in Table 13.2 for the abundance fractions of 19 panel pixels. As we can see in Figure 13.7 and Table 13.2, AFCLS-FLSMA performed significantly better than FCLS.

Now, we repeated the same experiments except that two background classes were used. In this case, the training samples used for the background for FVC-FLSMA and FLDA were the pixels in areas A and C, whereas the target signature matrix $\mathbf{M}_2$ used for the LSOSP and TCIMF was formed by $\mathbf{M}_2 = [\mathbf{p}_1\ \mathbf{p}_2\ \mathbf{p}_3\ \mathbf{p}_4\ \mathbf{p}_5\ \mathbf{b}_A\ \mathbf{b}_C]$.

Table 13.2 Quantitative results produced by the AFCLS-FLSMA and FCLS with a one-class background

	AFCLS-FLSMA	FCLS
p_{11}	1	0.76,169
p_{12}	0.85,046	0.58,071
p_{13}	0.29,045	0.07,407
p_{211}	1	0.90,912
p_{221}	1	0.78,833
p_{22}	0.96,217	0.77,286
p_{23}	0.32361	0.75,824
p_{311}	0.98,743	0.94,059
p_{312}	1	0.90,582
p_{32}	0.84,462	0.17,028
p_{33}	0.4375	0
p_{411}	1	0.50,936
p_{412}	0.99,869	0.6555
p_{42}	0.92,917	0.79,783
p_{43}	0.22,415	0.08,977
p_{511}	0.8945	0.82,314
p_{521}	1	1
p_{52}	0.95,603	0.89,426
p_{53}	0.21,056	0



(a) FVC-FLSMA

(b) FLDA

(c) LSOSP

(d) TCIMF

(e) CEM

Figure 13.6 Classification results of the 15 panels in Figure 13.2 produced by FVC-FLSMA, FLDA, LSOSP, TCIMF, and CEM, respectively, with a single-class background.

Figure 13.8(a)–(d) shows the classification results of the 15 panels in Figure 13.2 produced by FVC-FLSMA, FLDA, LSOSP, and TCIMF, respectively, with a two-class background. Once again, AFCLS-FLSMA and FCLS were implemented for quantitative analysis. Fig. 13.9 shows their quantification results in gray scales for the abundance fractions of 19 panel pixels which are unmixed and tabulated in Table 13.3. As shown in Figure 13.9 and Table 13.3 FCLS was significantly improved compared to its counterpart results obtained in Figure 13.7 and Table 13.2 and so was AFCLS-FLSMA. These experiments demonstrate that image back information is very important and crucial for FLSMA work effectively.

It should be noted that CEM only required one of panel signatures, $\mathbf{w}_l^T\boldsymbol{\mu}_j = \delta_{lj}$ as a desired target signature $\mathbf{d}$ while suppressing all signal sources other than the $\mathbf{d}$. So, it had the same performance regardless of how many background classes were used. In this case, the same result produced by CEM

Table 13.3 Quantitative results produced by the AFCLS-FLSMA and FCLS with a two-class background

	AFCLS-FLSMA	FCLS
p_{11}	1	0.76,169
p_{12}	0.95,109	0.57,429
p_{13}	0.22,928	0.08,378
p_{211}	0.99,931	0.90,912
p_{221}	1	0.78,833
p_{22}	0.97,827	0.82,227
p_{23}	0.31,634	0.41,548
p_{311}	0.98,742	0.92,294
p_{312}	1	0.90,582
p_{32}	0.93,698	0.31,199
p_{33}	0.55,418	0.29,653
p_{411}	1	0.50,936
p_{412}	1	0.34,952
p_{42}	0.96,166	0.80,963
p_{43}	0.23,799	0.15656
p_{511}	0.9506	0.8301
p_{521}	1	1
p_{52}	0.96,799	0.91,744
p_{53}	0.27,907	0.11,203

in Figure 13.6(e) was applied to this experiment, but not included in Figure 13.8. The results produced by FLDA in Figure 13.8(b) were pretty much the same as those in Figure 13.6(b). Interestingly, the performance of FVC-FLSMA in Figure 13.8(a) was slightly deteriorated compared to that in Figure 13.6(a). By contrast, TCIMF improved slightly if we compare Figure 13.8(d) to Figure 13.6(d). Once again, LSOSP was still the worst and did not improve very much. This is because there were still not sufficient endmembers to represent the background.

13.8.2 Image Background Characterized by Unsupervised Knowledge

In Section 13.7.1, we have seen that FVC-FLSMA did not perform as well as FLDA. This is mainly due to the fact that the image background in Figure 13.2 cannot be fully characterized by one single class or two classes. Since the image background is not well characterized, the performance of FLSMA was not as good as it did for the synthetic image in Section 13.7.1. In order to improve its performance, we need to find an appropriate set of background pixels that can well represent the image background. According to Chang (2003a) and Chang and Du (2004), the number of spectrally distinct signatures in the scene in Figure 13.2 is estimated to be 18. Apparently, using one or two background classes to describe the image background is far from being complete. This implies that we need at least 13 distinct signatures to characterize the image background in addition to the five panel signatures in Figure 1.16. In this case, we used an algorithm, referred to as automatic target generation process (ATGP), developed for the automatic target detection and classification algorithm (ATDCA) (Chang, 2003a; Ren and Chang, 2001) to generate 13 target pixels, denoted by $\{\mathbf{t}_k\}_{k=1}^{13}$ shown in Figure 13.10 that could be used to find training samples to represent the image background.

Since each of the 13 target pixels represent one single distinct class and they are not sufficient to be used for training samples, SAM was used to find pixels that are similar to each of the 13 target

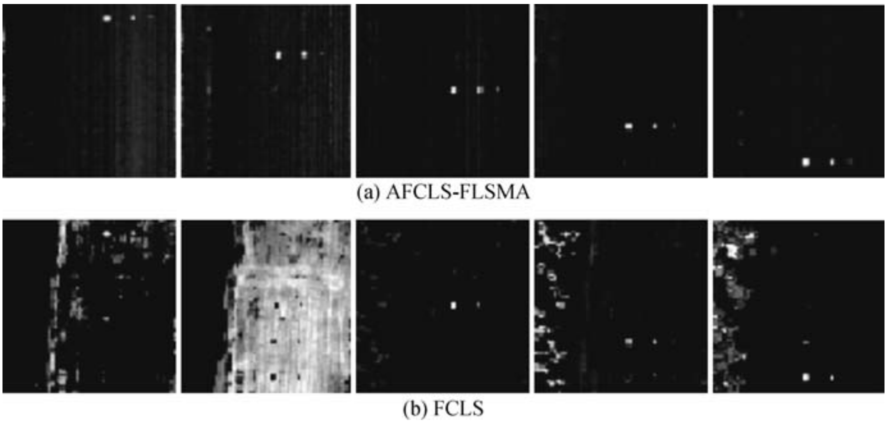


Figure 13.7 Classification results of the 15 panels in Figure 13.2 produced by the FVC-FLSMA, FLDA, LSOSP, and TCIMF, respectively, with a two-class background.

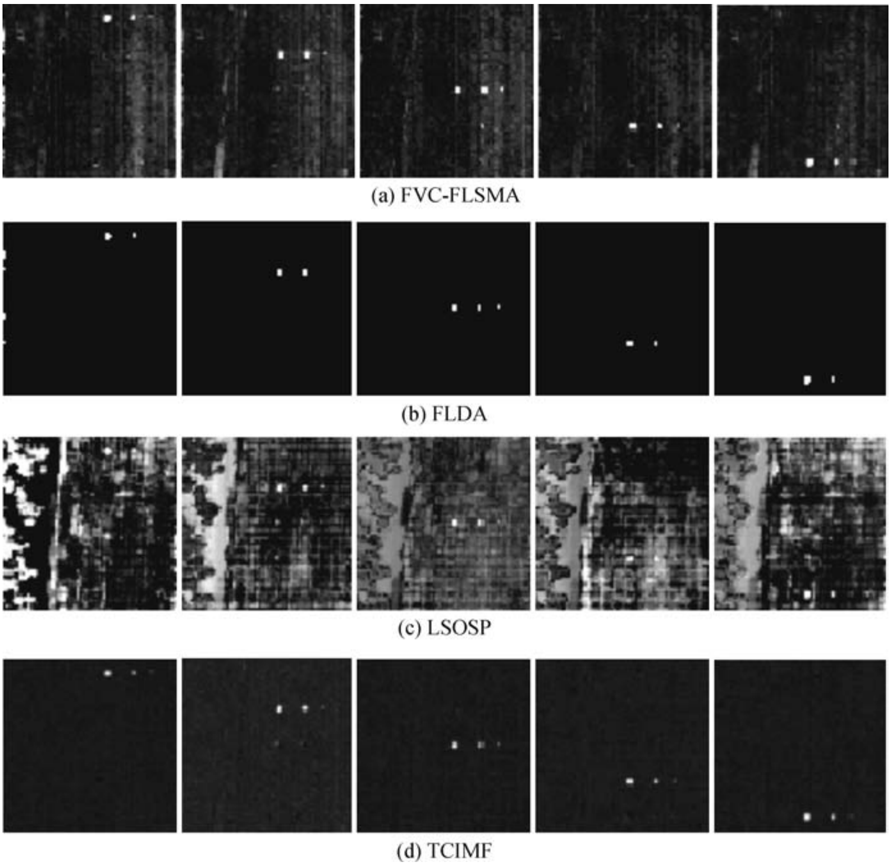
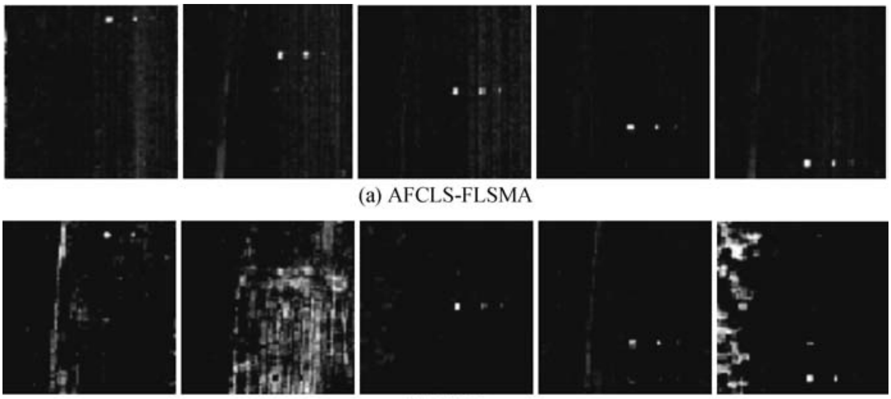


Figure 13.8 Classification results of the 15 panels in Figure 13.2 produced by produced by the AFCLS-FLSMA and FCLS with a one-class background.


(a) AFCLS-FLSMA

(b) FCLS

Figure 13.9 Classification results of the 15 panels in Figure 13.1(b) produced by produced by the AFCLS-FLSMA and FCLS with a two-class background.

pixels to form a set of training data for each of the 13 classes, $\{C_i\}_{i=1}^{13}$. Then the means of each of the 13 classes were further calculated, $\{\boldsymbol{\mu}_i\}_{i=1}^{13}$ to form part of a target signature matrix $\mathbf{M}$ for LSOSP and TCIMF. In our experiments, the threshold for SAM was set to 0.04 and the total number of found training sample was 698 that included the 14 R pixels in both the $3\,\text{m} \times 3\,\text{m}$ and $2\,\text{m} \times 2\,\text{m}$ panels in Figure 13.2.

EXAMPLE 13.5

(FVC-FLSMA vs. FLDA, LSOSP, TCIMF, and CEM)

Now we repeated the same experiments conducted for Example 13.4 where 13 classes are used to characterize the image background. Figures 13.11 and 13.12 show the results of FVC-FLSMA, FLDA, LSOSP, TCIMF, and CEM.

There are several interesting observations. One is that comparing the results in Figure 13.11(a) to those in Figures 13.6(a) and 13.8(a), the performance of FLSMA was not improved; even more background classes were included. Another is that LSOSP was slightly improved and its performance was comparable to that of FVC-FLSMA. But this is not the case as shown in Figures 13.6 and 13.8 where FVC-LSMA performed much better than LSOSP. Additionally, by comparing the results in Figure 13.11 to those in Figures 13.6 and 13.8, we can see that the performance of TCIMF was also slightly improved, while FLDA performed better than the results in Figures 13.6 and 13.8 with a few false

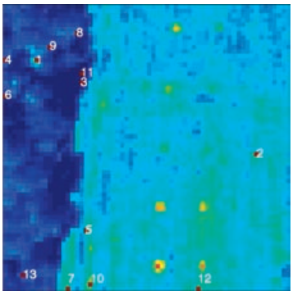
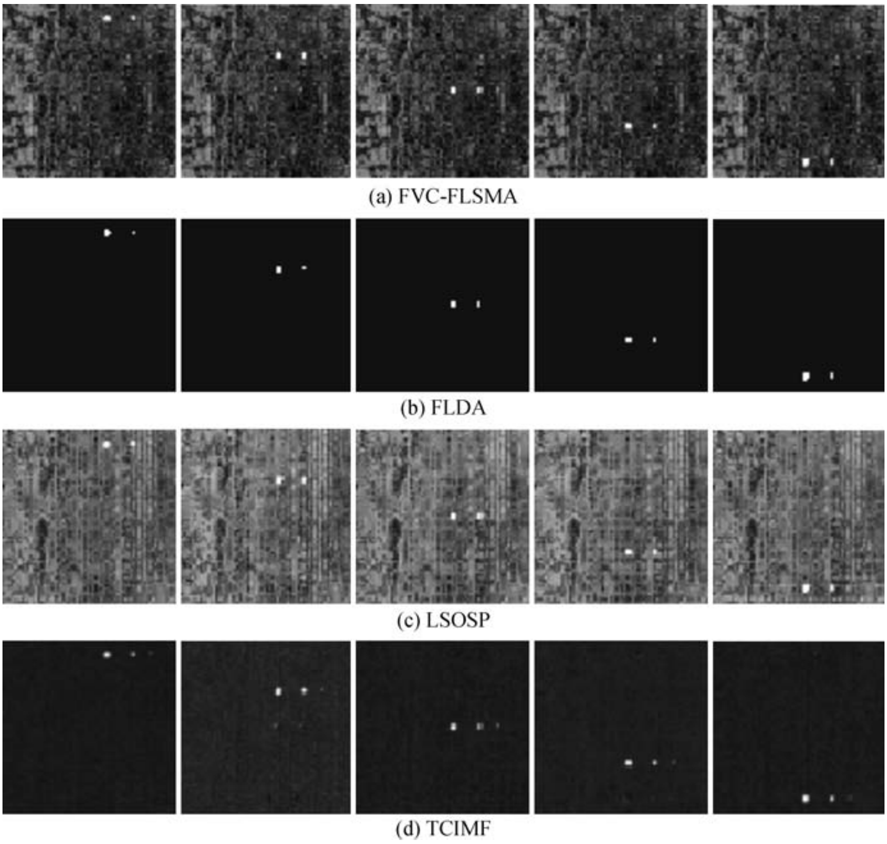


Figure 13.10 13 ATGP-generated target pixels.



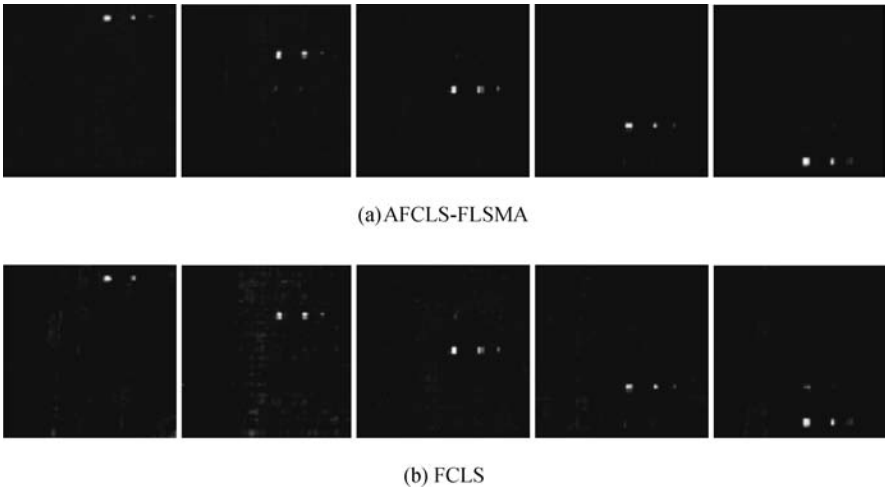
(a) FVC-FLSMA

(b) FLDA

(c) LSOSP

(d) TCIMF

Figure 13.11 Classification results of the 15 panels in Figure 13.2 produced by FVC-FLSMA, FLDA, LSOSP, and TCIMF, respectively, with unsupervised background generated by ATGP.



(a)AFCLS-FLSMA

(b) FCLS

Figure 13.12 Classification results of the 15 panels in Figure 13.2 produced by produced by the AFCLS-FLSMA and FCLS, respectively, with unsupervised background generated by ATGP.

Table 13.4 Quantitative results produced by the AFCLS-FLSMA and FCLS with unsupervised background generated by ATGP

	AFCLS-FLSMA	FCLS
p_{11}	1	0.86,787
p_{12}	0.77,781	0.57,333
p_{13}	0.21,253	0.00,666
p_{211}	1	0.79656
p_{221}	0.99,009	0.52,329
p_{22}	0.96,129	0.8766
p_{23}	0.29,875	0.47,608
p_{311}	0.98,231	0.91,616
p_{312}	1	0.92,114
p_{32}	0.7857	0.61,241
p_{33}	0.42,957	0.406
p_{411}	1	0.67,791
p_{412}	1	0.40,816
p_{42}	0.91,463	0.77,807
p_{43}	0.22,955	0.19,144
p_{511}	0.89,007	0.85,038
p_{521}	1	0.85,472
p_{52}	0.93,611	0.94,691
p_{53}	0.20,823	0.17,809

alarms occurring on the left edge. Interestingly, similar conclusions cannot be drawn if abundance constrained classifiers were used. It should be noted that the poor performance of FVC-FLSMA in Figure 13.11 resulted from the used training samples, which may not well represent the image background, not the technique itself. The issue of finding an appropriate set of training data is a great challenge to unsupervised mixed pixel classification. Now we further implemented AFCLS-FLSMA and FCLS for quantitative analysis. Fig. 13.12 shows their quantification results in gray scales. As we can see, FCLS was significantly improved and its performance was very comparably to AFLCS-FLSMA in Figure 13.12 by visual inspection. However, if we tabulate the unmixed abundance fractions of 19 panel pixels in Table 13.4, AFCLS-FLSMA still outperformed FCLS.

13.9 Conclusions

This chapter presents a new approach to LSMA, referred to as FLSMA, which directly extends the well-known FLDA to LSMA in two different ways. One is called FVC-FLSMA that constrains the Fisher ratio-generated feature vectors to mutual orthogonal directions. Another is called AC-FLSMA that imposes the sum-to-one and non-negativity constraints on abundance fractions in the least squares sense. It has been shown that FVC-FLSMA operates in the same way as LCMV does, with the only difference that the data correlation matrix used in LCMV is replaced by the within-class scatter matrix in FLSMA. Because the within-class scatter matrix is a more effective measure than the data correlation matrix in pattern classification, FVC-FLSMA performs better than LCMV in mixed pixel classification. Additionally, it also shows that LCDA is essentially the same as FVC-FLSMA. There are also three types of AC-FLSMA that can be derived in parallel in the same fashion as three types of constrained least squares methods are developed for LSMA in Chang (2003a). They are called ASCLS-FLSMA, ANCLS-FLSMA, and AFCLS-FLSMA with their respective counterparts in the abundance-constrained least squares LSMA, SCLS, non-negativity constrained least squares (NCLS), and FCLS. Since the mixed pixel classification is performed by AC-FLSMA using Fisher's ratio as a classification measure and least squares error as an abundance estimation criterion, AC-FLSMA also performs better than abundance-constrained least squares-based LSMA (ACLS-LSMA) and abundance-unconstrained FVC-FLSMA. Two concluding remarks are noteworthy. FLSMA presented in this chapter can be extended to unsupervised FLSMA (UFLSMA) in a similar manner as unsupervised knowledge is generated to characterize unknown background in Ji et al. (2004), where UFLSMA performed as well as FLSMA if the unsupervised generated training sample pool provided sufficient representative samples for each of classes. Another remark is that the performance of FLSMA relies heavily on the provided training samples. If the image is ill-represented by a given sample pool, FLSMA may perform poorly.

14

Weighted Abundance-Constrained Linear Spectral Mixture Analysis

Linear spectral mixture analysis (LSMA) has been used in a wide range of applications. It is, in general, implemented without constraints due to mathematical tractability. However, it has been shown that abundance-constrained LSMA (AC-LSMA) can improve abundance-unconstrained LSMA, specifically in quantification when accurate estimates of abundance fractions are necessary. As long as AC-LSMA is considered, two constraints are generally imposed: abundance sum-to-one constraint (ASC) and abundance nonnegativity constraint (ANC). A general and common approach to solving AC-LSMA is to estimate abundance fractions in the sense of least squares error (LSE) while satisfying desired imposed abundance constraints. Since the LSE resulting from each individual band in abundance estimation is not weighted in accordance with significance of each of full bands in the signatures used to unmix data sample vectors, the effect caused by LSE is assumed to be uniform over all bands, which is, in general, not necessarily true in practical applications. This chapter extends the commonly used AC-LSMA to three types of weighted AC-LSMA (WAC-LSMA) from three different perspectives: parameter estimation, pattern classification, and orthogonal subspace projection (OSP). As demonstrated by experiments, WAC-LSMA generally performs better than unweighted AC-LSMA where the latter can be considered a special case of WAC-LSMA with the weighting matrix chosen to be the identity matrix.

14.1 Introduction

LSMA has shown success in solving a variety of problems, such as subpixel detection, mixed pixel classification, quantification, etc. It assumes that there are p image endmembers, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$, and any image pixel vector $\mathbf{r}$ is a linear mixture of these p endmembers with appropriate abundance fractions, $\alpha_1, \alpha_2, \ldots, \alpha_p$, with α_j corresponding to the abundance fraction of the jth endmember $\mathbf{m}_j$ as follows:

$$\mathbf{r} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n} \tag{12.1}$$

where $\mathbf{n}$ is interpreted as a model or measurement error and $\mathbf{M} = \left[\mathbf{m}_1\, \mathbf{m}_2\,, \ldots,\, \mathbf{m}_p\right]$ is the endmember matrix formed by $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$. Because of mathematical tractability, LSMA is widely implemented without imposing any constraint on the abundance fractions $\alpha_1, \alpha_2, \ldots, \alpha_p$ of the

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

image endmembers $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$. However, it has been shown in the literature, for example, Chang (2003a), that AC-LSMA can improve abundance-unconstrained LSMA in many aspects, such as subpixel detection, mixed pixel classification, identification, specifically quantification that requires accurate abundance fraction estimation. As AC-LSMA is considered, two abundance constraints can be imposed on $\alpha_1, \alpha_2, \ldots, \alpha_p$ in (12.1): ASC, that is, $\sum_{j=1}^{p} \alpha_j = 1$ and abundance non-negativity-constraint (ANC), $\boldsymbol{\alpha} \geq \mathbf{0}$, that is, $\alpha_j \geq 0$ for all $1 \leq j \leq p$. A general and common approach to solving AC-LSMA is to estimate abundance fractions in the sense of LSE while satisfying the imposed constraints. Such an approach is referred to as LSE-based AC-LSMA or simply least squares AC-LSMA (LS AC-LSMA). More specifically, by virtue of the model in (12.1) an LSE problem can be described as follows:

$$(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})^T(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha}) \tag{14.1}$$

with $\mathbf{n}$ in (12.1) modeled as LSE, while constraining the abundance fractions $\alpha_1, \alpha_2, \ldots, \alpha_p$ on the model in (12.1) to find LSE solutions. Three types of LSE-based abundance-constrained LSMA are generally considered to solve (14.1) (Heinz and Chang 2001; Chang, 2003a), namely, sum-to-one constrained least squares (SCLS) that implements only ASC, nonnegativity constrained least squares (NCLS) that implements only ANC, and fully constrained least squares (FCLS) that implements both ASC and ANC. Despite the fact that AC-LSMA may require more sophisticated algorithmic implementations, the pay-off is sometimes worthwhile, specifically, for material substance quantification since it generally produces more accurate abundance fraction estimation.

According to (14.1), LSE is equally weighted for all bands by assuming a uniform effect on each band. In general, this may not be necessarily true. To generalize this concept, we consider a weighted LSE approach to (14.1) by introducing a weighting matrix $\mathbf{A}$ that is positive definite into (14.1) so that LSE is weighted by $\mathbf{A}$ via

$$(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})^T\mathbf{A}(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha}). \tag{14.2}$$

So, if $\mathbf{A} = \mathbf{I}$, identity matrix, (14.2) is reduced to (14.1). A key to success in using (14.2) is how to find an appropriate weighting matrix $\mathbf{A}$ that accounts for each of individual bands. As inspired by the three signal processing perspectives studied in Chang (2005) for LSMA, the weighting matrix $\mathbf{A}$ used for (14.2) can also be selected based on the same three signal processing perspectives for LSE AC-LSMA. One is a parameter estimation perspective derived from the well-known Mahalanobis distance or the Gaussian maximum likelihood estimator (GMLE). In this case, the weighting matrix $\mathbf{A}$ is selected to be the inverse of the data sample covariance matrix $\mathbf{K}$, $\mathbf{K}^{-1}$, in which case (14.2) becomes the Mahalanobis distance (MD) (Fukunaga, 1990) or the Gaussian maximum likelihood estimator (Richards and Jia, 1999). The resulting LSE AC-LSMA is called MD-weighted AC-LSMA. As an alternative, if $\mathbf{A}$ in (14.2) is replaced with $\mathbf{R}^{-1}$ (i.e., the inverse of data sample correlation matrix $\mathbf{R}$), (14.2) is reduced to a form of the linearly constrained minimum variance (LCMV) (Chang 2002b, 2003b) that is referred to as LCMV-weighted AC-LSMA. As a second approach, a selection of $\mathbf{A}$ can be derived from a view of pattern classification perspective based on Fisher's linear discriminant analysis (FLDA), as discussed in Chapters 2 and 13 where the within-class scatter matrix $\mathbf{S}_{\mathrm{W}}^{-1}$ is used for the weighting matrix $\mathbf{A}$ in (14.2) that yields weighted abundance-constrained Fisher's LSMA (WAC-FLSMA), referred to as $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA. A third approach is derived from a signal detection perspective. It selects a weighting matrix $\mathbf{A}$ arising from OSP. It is shown in Section 12.4 of this book, Chang (2003a) and Chang (2005), that the undesired signature rejection matrix, $P_{\mathbf{U}}^{\perp}$, used in the OSP detector (12.9) can be approximated by $\mathbf{R}^{-1}$ if prior knowledge of the undesired signatures in $\mathbf{U}$ is not available. Using this interpretation substituting $P_{\mathbf{U}}^{\perp}$ for the weighting matrix $\mathbf{A}$ in (14.2) results in OSP-weighted AC-LSMA. An interesting finding is that if the weighting matrix $\mathbf{A}$ is selected by the signature subspace projection (SSP) matrix (Tu et al., 1997; Chang et al., 1998) formed by the endmember matrix $\mathbf{M}$ in (12.1) or (14.1), the resulting SSP-weighted AC-LSMA can be shown to be identical to the unweighted AC-LSMA in (14.1), in which case $\mathbf{A}$ is the identity matrix due to the fact that both the SSP approach and LSMA are LSE-based methods and the weighted matrix specified by SSP does not provide any additional advantage. Nevertheless, as will be demonstrated by experiments, all these three types of weighted AC-LSMA specified by appropriate selections for the weight matrix $\mathbf{A}$ in (14.2) generally perform better than unweighted AC-LSMA described by (14.1).

14.2 Abundance-Constrained LSMA (AC-LSMA)

When the abundance-unconstrained LSMA specified by (12.1) is considered, its unconstrained LSE solution to (14.2) is given by

$$\hat{\boldsymbol{\alpha}}(\mathbf{r}) = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{r} \tag{14.3}$$

A commonly used least squares method is to minimize the LSE problem specified by (12.1) over the abundance vector $\boldsymbol{\alpha} = \left(\alpha_1, \alpha_2, \ldots, \alpha_p\right)^T$ subject to the constraints imposed by ASC and/or ANC on $\boldsymbol{\alpha}$. If we impose either ASC or ANC or both on (14.1), three LSE problems derived from AC-LSMA can be formulated as follows (Heinz and Chang 2001; Chang, 2003b).

i. Abundance sum-to-one constrained LSMA (ASCLS-LSMA) problem:

$$\min_{\boldsymbol{\alpha}}\left\{(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^T(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})\right\} \text{ subject to } \sum\nolimits_{j=1}^{p}\alpha_j = 1 \tag{14.4}$$

ii. Abundance nonnegativity-constrained LSMA (ANCLS-LSMA) problem:

$$\min_{\boldsymbol{\alpha}}\left\{(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^T(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})\right\} \text{ subject to } \boldsymbol{\alpha} \geq \mathbf{0} \tag{14.5}$$

iii. Abundance fully constrained LSMA (AFCLS-LSMA) problem:

$$\min_{\boldsymbol{\alpha}}\left\{(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^T(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})\right\} \text{ subject to } \boldsymbol{\alpha} \geq \mathbf{0} \text{ and } \sum\nolimits_{j=1}^{p}\alpha_j = 1 \tag{14.6}$$

The solutions to (14.4)–(14.6) were well documented by Heinz and Chang (2001) and Chang (2003a) and will be referred to as SCLS, NCLS, and FCLS solutions, respectively.

14.3 Weighted Least-Squares Abundance-Constrained LSMA

It should be noted that the LSE specified by (14.1) does not include a weighting matrix to account for significance of bands in signatures used to form the $\mathbf{M}$, in which case the identity matrix $\mathbf{I}$ is used in (14.4)–(14.6). However, this is not necessarily an optimal way to impose LSE since it weights LSE caused by each band equally significant. If a weighting matrix $\mathbf{A}$ is included in (14.2) to account for LSEs resulting from different bands (i.e., replacing $\mathbf{I}$ in (14.1) with $\mathbf{A}$), then an $\mathbf{A}$-weighted LSE problem is to find a solution that solves

$$\min_{\boldsymbol{\alpha}}\left\{(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^T\mathbf{A}(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})\right\} \tag{14.7}$$

Suppose that **A** is a positive-definite and symmetric matrix; we can use $\mathbf{A}^{1/2}$ that is, the square-root form of **A** to whiten the LSE in (14.7) as follows:

$$\begin{aligned}&(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^T\mathbf{A}^{1/2}\mathbf{A}^{1/2}(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha}) = (\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^T\left(\mathbf{A}^{1/2}\right)^T\mathbf{A}^{1/2}(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})\\ &= \left(\mathbf{A}^{1/2}\mathbf{r}-\mathbf{A}^{1/2}\mathbf{M}\boldsymbol{\alpha}\right)^T\left(\mathbf{A}^{1/2}\mathbf{r}-\mathbf{A}^{1/2}\mathbf{M}\boldsymbol{\alpha}\right)\end{aligned} \tag{14.8}$$

Using a linear transformation defined by

$$\widehat{\mathbf{r}} = \mathbf{A}^{1/2}\mathbf{r} \text{ and } \widehat{\mathbf{M}} = \mathbf{A}^{1/2}\mathbf{M} \tag{14.9}$$

an **A**-whitened LSE can be obtained by

$$\min_{\boldsymbol{\alpha}}\left\{\left(\widehat{\mathbf{r}}-\widehat{\mathbf{M}}\boldsymbol{\alpha}\right)^T\left(\widehat{\mathbf{r}}-\widehat{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \tag{14.10}$$

which is reduced to minimization of (14.2), except that both the image pixel vector $\hat{\mathbf{r}}$ and the matrix $\widehat{\mathbf{M}}$ have been whitened by the weighting matrix **A** via (14.9). Following the same approach that derives (14.4)–(14.6), we can also consider three types of **A**-weighted AC-LSMA similar to (14.4)–(14.6) as follows.

i. **A**-weighted abundance sum-to-one constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\widehat{\mathbf{r}}-\mathbf{M}\boldsymbol{\alpha}\right)^T\left(\widehat{\mathbf{r}}-\widehat{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \sum\nolimits_{j=1}^{p}\alpha_j = 1 \tag{14.11}$$

ii. **A**-weighted abundance nonnegativity-constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\widehat{\mathbf{r}}-\widehat{\mathbf{M}}\boldsymbol{\alpha}\right)^T\left(\widehat{\mathbf{r}}-\widehat{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \boldsymbol{\alpha} \geq \mathbf{0} \tag{14.12}$$

iii. **A**-weighted abundance fully constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\widehat{\mathbf{r}}-\widehat{\mathbf{M}}\boldsymbol{\alpha}\right)^T\left(\widehat{\mathbf{r}}-\widehat{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \boldsymbol{\alpha} \geq \mathbf{0} \text{ and } \sum\nolimits_{j=1}^{p}\alpha_j = 1 \tag{14.13}$$

As shown in Chapter 12 (Chang 2003a; 2005), LSMA can be interpreted from three different perspectives: signal detection that results in OSP, parameter estimation that results in MD or GMLE, and pattern classification that results in FLDA. In what follows, these three same perspectives can also be used to develop three different approaches for AC LSMA by appropriately selecting a weighted matrix **A** used in (14.2).

14.3.1 Weighting Matrix Derived from a Parameter Estimation Perspective

There are several natural approaches to selection of the weighting matrix **A** in (14.2) that accounts for spectral correlation used in parameter estimation. One is the sample covariance spectral matrix **K** and the other is the sample correlation spectral matrix **R**.

14.3.1.1 MD-Weighted AC-LSMA

One well-known example using a weighted mean squared error is MD, which is also known as GMLE that uses the data covariance matrix $\mathbf{K}^{-1}$ as a weighting matrix that turns out to be a whitening matrix in signal processing and communications. Substituting $\mathbf{K}^{-1}$ for $\mathbf{A}$ in to (14.7) yields

$$\min_{\boldsymbol{\alpha}}\left\{(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^T\mathbf{K}^{-1}(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})\right\} \tag{14.14}$$

Using a linear transformation similar to (14.9), we can define

$$\hat{\mathbf{r}}=\mathbf{K}^{-1/2}\mathbf{r}\ \text{ and }\ \mathbf{M}\,\text{into}\,\hat{\mathbf{M}}=\mathbf{K}^{-1/2}\mathbf{M} \tag{14.15}$$

Then, the resulting $\mathbf{K}^{-1}$-whitened LSE is found by

$$\min_{\boldsymbol{\alpha}}\left\{\left(\hat{\mathbf{r}}-\hat{\mathbf{M}}\boldsymbol{\alpha}\right)^T\left(\hat{\mathbf{r}}-\hat{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \tag{14.16}$$

that is similar to (14.10). By virtue of (14.16), we can also consider three types of MD-weighted AC-LSMA similar to (14.11)–(14.13) as follows.

i. MD-weighted abundance sum-to-one constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\hat{\mathbf{r}}-\hat{\mathbf{M}}\boldsymbol{\alpha}\right)^T\left(\hat{\mathbf{r}}-\hat{\mathbf{M}}\boldsymbol{\alpha}\right)\right\}\ \text{subject to}\ \sum\nolimits_{j=1}^{p}\alpha_j=1 \tag{14.17}$$

ii. MD-weighted abundance nonnegativity-constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\hat{\mathbf{r}}-\hat{\mathbf{M}}\boldsymbol{\alpha}\right)^T\left(\hat{\mathbf{r}}-\hat{\mathbf{M}}\boldsymbol{\alpha}\right)\right\}\ \text{subject to}\ \boldsymbol{\alpha}\geq\mathbf{0} \tag{14.18}$$

iii. MD-weighted abundance fully constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\hat{\mathbf{r}}-\hat{\mathbf{M}}\boldsymbol{\alpha}\right)^T\left(\hat{\mathbf{r}}-\hat{\mathbf{M}}\boldsymbol{\alpha}\right)\right\}\ \text{subject to}\ \boldsymbol{\alpha}\geq\mathbf{0}\ \text{and}\ \sum\nolimits_{j=1}^{p}\alpha_j=1 \tag{14.19}$$

The solutions to (14.17)–(14.19) are referred to as MD-weighted SCLS, MD-weighted NCLS, and MD-weighted FCLS, respectively.

14.3.1.2 LCMV-Weighted AC-LSMA

The LSE in (14.14) is derived from MD or GMLE. If we replace the sample covariance spectral matrix $\mathbf{K}$ in (14.14) with the sample correlation spectral matrix $\mathbf{R}$, we can also derive an LCMV-based abundance-constrained LSE problem given by

$$\min_{\boldsymbol{\alpha}}\left\{(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^T\mathbf{R}^{-1}(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})\right\} \tag{14.20}$$

which uses the data correlation matrix $\mathbf{R}$ as a weighting matrix to replace the data sample covariance matrix $\mathbf{K}$. Once again, using a linear transformation similar to (14.9) by mapping $\mathbf{r}$ into $\bar{\mathbf{r}}=\mathbf{R}^{-1/2}\mathbf{r}$ and $\overline{\mathbf{M}}=\mathbf{R}^{-1/2}\mathbf{M}$, we can also obtain an $\mathbf{R}^{-1}$-whitened LSE problem given by

$$\min_{\boldsymbol{\alpha}}\left\{\left(\bar{\mathbf{r}}-\overline{\mathbf{M}}\boldsymbol{\alpha}\right)^T\left(\bar{\mathbf{r}}-\overline{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \tag{14.21}$$

which is another correlation-based least-squares error problem, referred to as an LCMV-weighted abundance-constrained LSE problem. Similarly to (14.11)–(14.13), we can also consider three types of LCMV-weighted AC-LSMA as follows.

i. LCMV-weighted abundance sum-to-one constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\bar{\mathbf{r}}-\overline{\mathbf{M}}\boldsymbol{\alpha}\right)^{T}\left(\bar{\mathbf{r}}-\overline{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \sum\nolimits_{j=1}^{p}\alpha_j=1 \tag{14.22}$$

ii. LCMV-weighted abundance nonnegativity-constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\bar{\mathbf{r}}-\overline{\mathbf{M}}\boldsymbol{\alpha}\right)^{T}\left(\bar{\mathbf{r}}-\overline{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \boldsymbol{\alpha}\geq\mathbf{0} \tag{14.23}$$

iii. LCMV-weighted abundance fully constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\bar{\mathbf{r}}-\overline{\mathbf{M}}\boldsymbol{\alpha}\right)^{T}\left(\bar{\mathbf{r}}-\overline{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \boldsymbol{\alpha}\geq\mathbf{0} \text{ and } \sum\nolimits_{j=1}^{p}\alpha_j=1 \tag{14.24}$$

The solutions to (14.23)–(14.25) are referred to as LCMV-weighted SCLS, LCMV-weighted NCLS, and LCMV-weighted FCLS, respectively.

14.3.2 Weighting Matrix Derived from Fisher's Linear Discriminant Analysis Perspective

FLDA is one of most widely used pattern classification techniques in pattern recognition and has been considered in Chapters 2 and 13, where a Fisher's ratio-based LSE problem could be formulated as

$$\min_{\boldsymbol{\alpha}}\left\{(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^{T}\mathbf{S}_{\mathrm{W}}^{-1}(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})\right\} \tag{13.27}$$

with the weighting matrix $\mathbf{A}$ in (14.7) being replaced by $\mathbf{S}_{\mathrm{W}}^{-1}$. So, with the transformation defined by $\tilde{\mathbf{r}}=\mathbf{S}_{\mathrm{W}}^{-1/2}\mathbf{r}$ and $\tilde{\mathbf{M}}=\mathbf{S}_{\mathrm{W}}^{-1/2}\mathbf{M}$ via (13.28), (13.27) can be whitened in the sense of classification by $\mathbf{S}_{\mathrm{W}}^{-1}$ and becomes

$$\min_{\boldsymbol{\alpha}}\left\{\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)^{T}\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \tag{13.29}$$

Therefore, we can also obtain three types of $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA, also referred to as abundance-constrained least-squares FLDA (ACLS-FLDA) in Chapter 13, as follows:

i. $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted abundance sum-to-one constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)^{T}\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \sum\nolimits_{j=1}^{p}\alpha_j=1 \tag{14.25}$$

ii. $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted abundance nonnegativity-constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)^{T}\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \boldsymbol{\alpha}\geq\mathbf{0} \tag{14.26}$$

iii. $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted abundance fully constrained least LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)^{T}\left(\tilde{\mathbf{r}}-\tilde{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \boldsymbol{\alpha}\geq\mathbf{0} \text{ and } \sum\nolimits_{j=1}^{p}\alpha_j=1 \tag{14.27}$$

The solutions to (14.25)–(14.27) are referred to as $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted SCLS, $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted NCLS, and $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted FCLS, respectively.

14.3.3 Weighting Matrix Derived from an Orthogonal Subspace Projection Perspective

As we have seen in Sections 14.3.1 and 14.3.2, the weighting matrix **A** can be selected by either the sample spectral covariance/correlation matrix or Fisher's ratio. In this section, we present another selection of the weighting matrix **A** by criteria based on OSP.

14.3.3.1 OSP-Weighted AC-LSMA

According to the signal-decomposed interference-annihilated (SDIA) model in Du and Chang (2004), the signal sources can be decomposed into signal sources and unwanted signal sources. If we let **U** be the unwanted signature matrix formed of such interferers, we can project all image pixels onto the space $\langle\mathbf{U}\rangle^{\perp}$ that is orthogonal to the space linearly spanned by the signal sources in **U** and then perform the LSE problem specified by (14.1) in $\langle\mathbf{U}\rangle^{\perp}$. In doing so, the weighting matrix **A** in (14.7) is designed by the unwanted signature rejector $P_{\mathbf{U}}^{\perp}$ defined by

$$P_{\mathbf{U}}^{\perp}=\mathbf{I}-\mathbf{U}(\mathbf{U}^{T}\mathbf{U})^{-1}\mathbf{U}^{T}. \tag{2.86}$$

The resulting LSE problem is obtained by replacing **A** in (14.7) with $P_{\mathbf{U}}^{\perp}$ in (2.86) and given by

$$\min_{\boldsymbol{\alpha}}\left\{(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^{T}P_{\mathbf{U}}^{\perp}(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})\right\}. \tag{14.28}$$

Since $P_{\mathbf{U}}^{\perp}$ is idempotent, $P_{\mathbf{U}}^{\perp}=\left(P_{\mathbf{U}}^{\perp}\right)^{2}$ and $\left(P_{\mathbf{U}}^{\perp}\right)^{T}=P_{\mathbf{U}}^{\perp}$. This implies that

$$\begin{aligned}(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^{T}P_{\mathbf{U}}^{\perp}(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})&=(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^{T}\left(P_{\mathbf{U}}^{\perp}\right)^{2}(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})\\&=(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^{T}\left(P_{\mathbf{U}}^{\perp}\right)^{T}P_{\mathbf{U}}^{\perp}(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})=\left(P_{\mathbf{U}}^{\perp}\mathbf{r}-P_{\mathbf{U}}^{\perp}\mathbf{M}\boldsymbol{\alpha}\right)^{T}\left(P_{\mathbf{U}}^{\perp}\mathbf{r}-P_{\mathbf{U}}^{\perp}\mathbf{M}\boldsymbol{\alpha}\right).\end{aligned} \tag{14.29}$$

Using a linear transformation similar to (14.9) by mapping **r** into $\breve{\mathbf{r}}=P_{\mathbf{U}}^{\perp}\mathbf{r}$ and **M** into $\breve{\mathbf{M}}=P_{\mathbf{U}}^{\perp}\mathbf{M}$, we can also obtain a similar form to (14.10) given by

$$\min_{\boldsymbol{\alpha}}\left\{\left(\breve{\mathbf{r}}-\breve{\mathbf{M}}\boldsymbol{\alpha}\right)^{T}\left(\breve{\mathbf{r}}-\breve{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \tag{14.30}$$

which is referred to as the OSP-weighted abundance-constrained LSE problem. Consequently, we can also consider three types of OSP-weighted AC-LSMA as follows:

i. OSP-weighted abundance sum-to-one constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\breve{\mathbf{r}}-\breve{\mathbf{M}}\boldsymbol{\alpha}\right)^{T}\left(\breve{\mathbf{r}}-\breve{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \sum\nolimits_{j=1}^{p}\alpha_j=1 \tag{14.31}$$

ii. OSP-weighted abundance nonnegativity-constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\breve{\mathbf{r}}-\breve{\mathbf{M}}\boldsymbol{\alpha}\right)^{T}\left(\breve{\mathbf{r}}-\breve{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \boldsymbol{\alpha}\geq\mathbf{0} \tag{14.32}$$

iii. OSP-weighted abundance fully constrained LSE problem:

$$\min_{\boldsymbol{\alpha}}\left\{\left(\breve{\mathbf{r}}-\breve{\mathbf{M}}\boldsymbol{\alpha}\right)^{T}\left(\breve{\mathbf{r}}-\breve{\mathbf{M}}\boldsymbol{\alpha}\right)\right\} \text{ subject to } \boldsymbol{\alpha}\geq\mathbf{0} \text{ and } \sum\nolimits_{j=1}^{p}\alpha_j=1 \tag{14.33}$$

The solutions to (14.31)–(14.33) are referred to as OSP-weighted SCLS, OSP-weighted NCLS, and OSP-weighted FCLS, respectively. In order for the OSP-weighted ACLSMA to work effectively, the unknown signal sources in the matrix **U** must be found appropriately in an unsupervised means. One such an algorithm is the automatic target generation process (ATGP) developed by Ren and Chang (2003), and Chapter 8 can be used for this purpose.

14.3.3.2 SSP-Weighted AC-LSMA

As an alternative to (14.28), we can also formulate an LSE problem by performing abundance estimation in the space that is linearly spanned by the signal sources in the signature matrix **M** exclusively. This can be done by replacing $P_{\mathbf{U}}^{\perp}$ in (14.28) with a signature subspace projector, $P_{\mathbf{M}}$, defined in Scharf (1991) by

$$P_{\mathbf{M}}=\mathbf{M}(\mathbf{M}^{T}\mathbf{M})^{-1}\mathbf{M}^{T} \tag{14.34}$$

and the resulting LSE problem is referred to as SSP-weighted AC-LSMA that finds a solution to the following optimization problem:

$$\min_{\boldsymbol{\alpha}}\left\{(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})^{T}P_{\mathbf{M}}(\mathbf{r}-\mathbf{M}\boldsymbol{\alpha})\right\} \tag{14.35}$$

Since $P_{\mathbf{M}}$ is idempotent, $(P_{\mathbf{M}})^2=P_{\mathbf{M}}$ and $(P_{\mathbf{M}})^T=P_{\mathbf{M}}$. Using a linear transformation similar to (14.9) by mapping **r** into $\hat{\mathbf{r}}=P_{\mathbf{M}}\mathbf{r}$ and $\hat{\mathbf{M}}=P_{\mathbf{M}}\mathbf{M}=\mathbf{M}$, we can also obtain the following form similar to (14.28):

$$\min_{\boldsymbol{\alpha}}\left\{(\hat{\mathbf{r}}-\mathbf{M}\boldsymbol{\alpha})^{T}(\hat{\mathbf{r}}-\mathbf{M}\boldsymbol{\alpha})\right\} \tag{14.35a}$$

Interestingly, the solution to (14.35) is

$$\begin{aligned}\hat{\boldsymbol{\alpha}}(\hat{\mathbf{r}})&=(\mathbf{M}^{T}\mathbf{M})^{-1}\mathbf{M}^{T}\hat{\mathbf{r}}=(\mathbf{M}^{T}\mathbf{M})^{-1}\mathbf{M}^{T}P_{\mathbf{M}}\mathbf{r}\\&=(\mathbf{M}^{T}\mathbf{M})^{-1}\mathbf{M}^{T}\mathbf{M}(\mathbf{M}^{T}\mathbf{M})^{-1}\mathbf{M}^{T}\mathbf{r}=(\mathbf{M}^{T}\mathbf{M})^{-1}\mathbf{M}^{T}\mathbf{r}=\hat{\boldsymbol{\alpha}}(\mathbf{r})\end{aligned} \tag{14.36}$$

which turns out to be identical to the abundance-unconstrained least-squares LSMA solution given by (14.4). As a result, the three types of SSP-weighted AC-LSMA are essentially the same ASC-LSMA, ANC-LSMA, and AFC-LSMA described by (14.4)–(14.6). This is because the weighted matrix specified by the SSP does not provide any additional advantage, as shown in (14.36) and $P_{\mathbf{M}}\mathbf{M}=\mathbf{M}$.

14.4 Synthetic Image-Based Computer Simulations

To evaluate performance of WAC-LSMA, a synthetic image similar to Figure 13.1 is simulated in Figure 14.1, where an area marked by “A” was used to simulate the image background.

This synthetic image scene to be simulated has size of 64 × 64 pixel vectors and 20 panels with various sizes arranged in a 5 × 4 matrix and located at the center of the scene, as shown in Figure 14.2(a).

The five panel signatures in Figure 1.16 are used to simulate these 20 panels. For row i, the panel signature $\mathbf{p}_i$ was used to simulate four panels with a 2 × 2-pixel panel, $\{p^i_{1,11}, p^i_{1,12}, p^i_{1,21}, p^i_{1,22}\}$ in the first column, a 1 × 2-pixel panel, $\{p^i_{2,11}, p^i_{2,12}\}$ in the second column, a one-pixel panel, $p^i_{3,1}$ in the third column, and a one-pixel panel, $p^i_{4,1}$ in the fourth column, respectively. While the pixels in all the 2 × 2-pixel panels and the 1 × 2-pixel panels are pure pixels simulated by 100% panel signature $\mathbf{p}_i$, the panels, $\mathbf{p}^i_{3,1}$ and $\mathbf{p}^i_{4,1}$, are subpixel panels simulated by (50%$\mathbf{p}_i$, 50%$\mathbf{b}_A$) and (25%$\mathbf{p}_i$, 75%$\mathbf{b}_A$), where the background signature $\mathbf{b}_A$ is a grass signature obtained by averaging all the pixels in the area “A” marked in Figure 14.1.

Figure 14.2(b) is a synthetic image simulated by implanting the 20 panels in Figure 14.2(a) in the grass signature $\mathbf{b}_A$-generated image background. Unlike Figure 13.1(c) that was simulated by implanting panel pixels in the image background in the same way as the scenario TI2 is simulated as described in Chapter 4, Figure 14.2(c) was simulated by adding a Gaussian noise to the synthetic image in Figure 14.2(b) in the same way as the scenario TI3 was simulated as described in Chapter 4 with the signal-to-noise ratio 20:1 defined in Harsanyi and Chang (1994). If we compare Figure 14.2(c) with Figure 13.1(c), the panel pixels in Figure 14.2(c) are invisible because

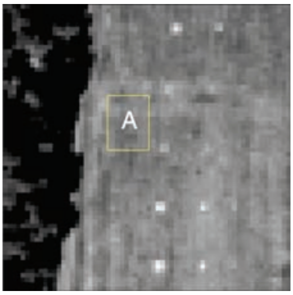
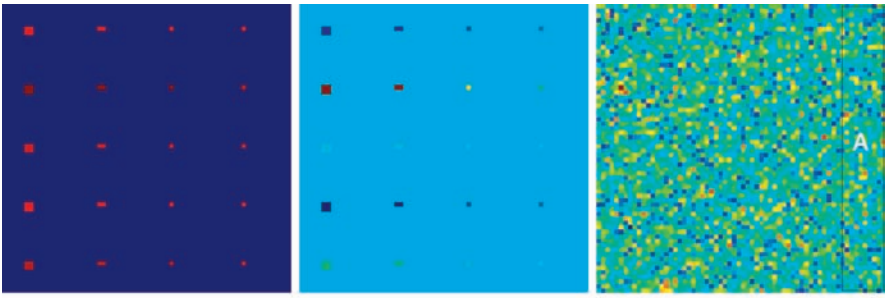


Figure 14.1 A HYDICE image scene with background signature marked by A.



(a) 20 simulated panels (b) panels with background (c) image with noise

Figure 14.2 A synthetic image with 20 implanted panels.

these panel pixels are corrupted by Gaussian noise as compared to those in Figure 13.1(c), which are clean and visible in the noise image background. Thus, estimating abundance fractions of the panel pixels in the synthetic image in Figure 14.2(c) should be more difficult than those in Figure 13.1(c). Since full constraints, that is, ASC + ANC on abundance fractions are of major interest in this section, the algorithms to be evaluated for comparative analysis are weighted fully abundance constrained LSMA that are MD-weighted AC-LSMA, LCMV-weighted AC-LSMA, $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA, and OSP-weighted AC-LSMA plus the FCLS that is the unweighted AC-LSMA (i.e., $\mathbf{A} = \mathbf{I}$, unweighting AC-LSMA) and also turns out to be the SSP-weighted AC-LSMA (i.e., $\mathbf{A}^{-1} = P_{\mathbf{M}}$, unweighting AC-LSMA).

EXAMPLE 14.1

(with Complete Knowledge of Panel Signatures and Background Signature)

This example assumes that complete knowledge of the five panel signatures in Figure 1.16 and the background signature $\mathbf{b}_{\mathrm{A}}$ is given *a priori*. Figure 14.3 shows the abundance fractions of the 20 panels in Figure 14.2(c) unmixed by the following six methods, (a) MD-weighted AC-LSMA, (b) LCMV-weighted AC-LSMA, (c) $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA, (d) OSP-weighted AC-LSMA, (e) FCLS, and (f) unconstrained LSOSP, respectively, with additional results produced by the unconstrained LSOSP being included in Figure 14.3 for comparison.

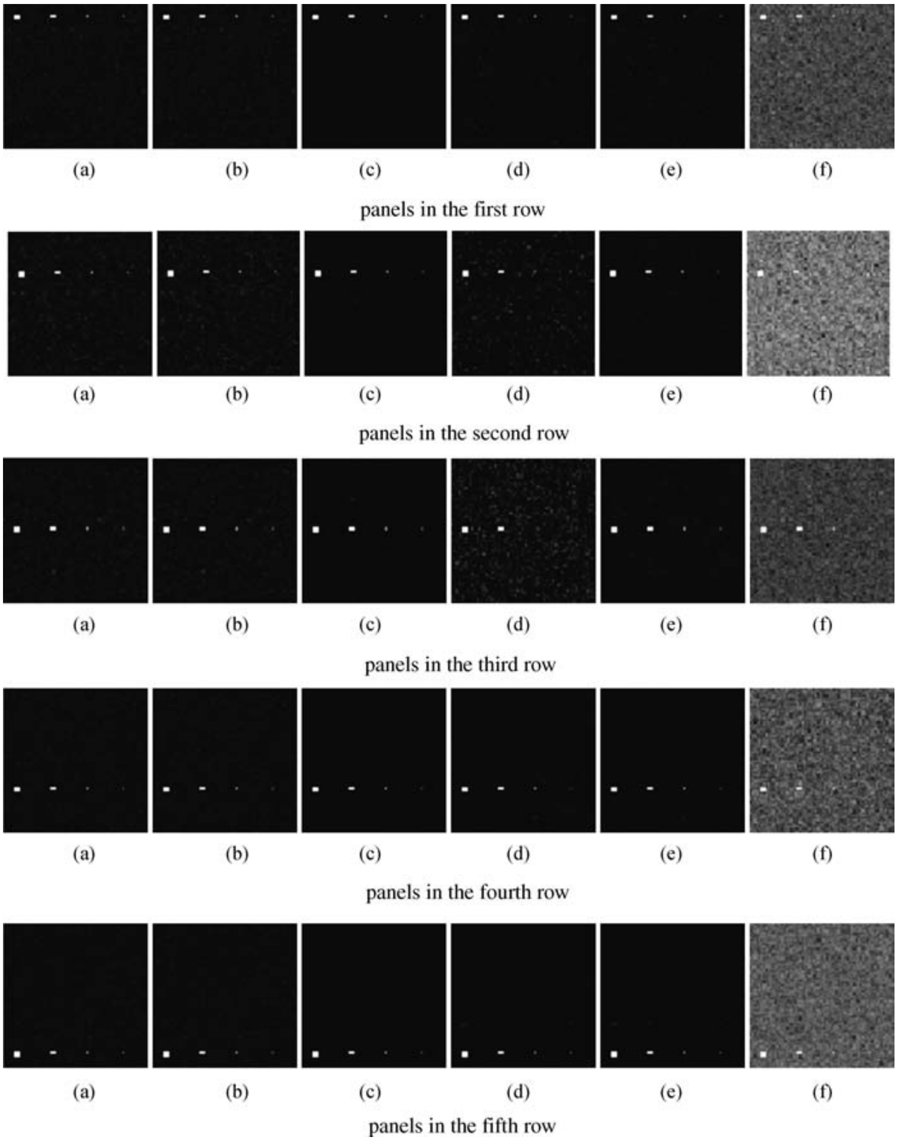
As shown in Figure 14.3, all the weighted AC-LSMA methods produced very comparable results and performed significantly better than the unconstrained LSOSP did. It should be noted that the $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA requires a training set to implement. For our experiments, the training samples were selected and consisted of the pixels in the 2 × 2-pixel panels in the first column of five rows in Figure 14.2(a) and all background pixels in the right strip marked by "A" in Figure 14.2(c). Also, when the OSP-weighted AC-LSMA was implemented, the threshold chosen for the SAM to find interferers was set to 0.03 that resulted in 18 interferers. For quantitative analysis, Table 14.1 tabulates abundance fractions unmixed by the supervised MD-weighted AC-LSMA, LCMV-weighted AC-LSMA, $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA, and OSP-weighted AC-LSMA, FCLS, and unconstrained LSOSP, where the quantification results provide further evidence that the $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA was the best in producing accurate abundance fractions in most of the 20 panels. This is due to the fact that the $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA took advantage of training samples to perform classification based on Fisher's ratio.

Figure 14.4 graphically plots the abundance fractions of the 20 panels in Table 14.1 for visual assessment. According to Figure 14.4, both MD-weighted AC-LSMA and LCMV-weighted AC-LSMA labeled by (a) and (b) performed in a very similar way. Surprisingly, the OSP-weighted AC-LSMA labeled by (d) did not perform as good as did the MD-weighted AC-LSMA and LCMV-weighted AC-LSMA. On the other hand, the FCLS labeled by (e) seemed to perform very well and slightly better than the MD-weighted AC-LSMA and LCMV-weighted AC-LSMA in quantification of full panel pixels, but not for subpixel panels.

EXAMPLE 14.2

(with No Prior Knowledge About Panel Signatures and Background Signature)

Unlike Example 14.1, no prior knowledge about the synthetic image in Figure 14.2(c) was assumed. In particular, there was no knowledge about how many signatures would represent the image scene. In this case, we ought to find a set of these signatures directly from the data in an unsupervised manner. First, we need to determine the number of signatures required to be generated for the scene. The virtual dimensionality (VD) developed in Chapter 5 provided a good estimate of the number of spectrally distinct signatures, p, in hypeprspectral image data where the Harsanyi–Farrand–Chang (HFC) and noise-whitened HFC (NWHFC) methods were used in this example to determine p. Table 14.1 tabulates the VD estimated by the HFC and NWHFC



panels in the first row

panels in the second row

panels in the third row

panels in the fourth row

panels in the fifth row

Figure 14.3 Abundance fraction results of 20 panels estimated by the supervised five AC-LSMA methods and unconstrained LSOSP.

methods in accordance with various false-alarm probabilities indicated by P_F. According to Table 14.2, a good estimate for p was set to 6.

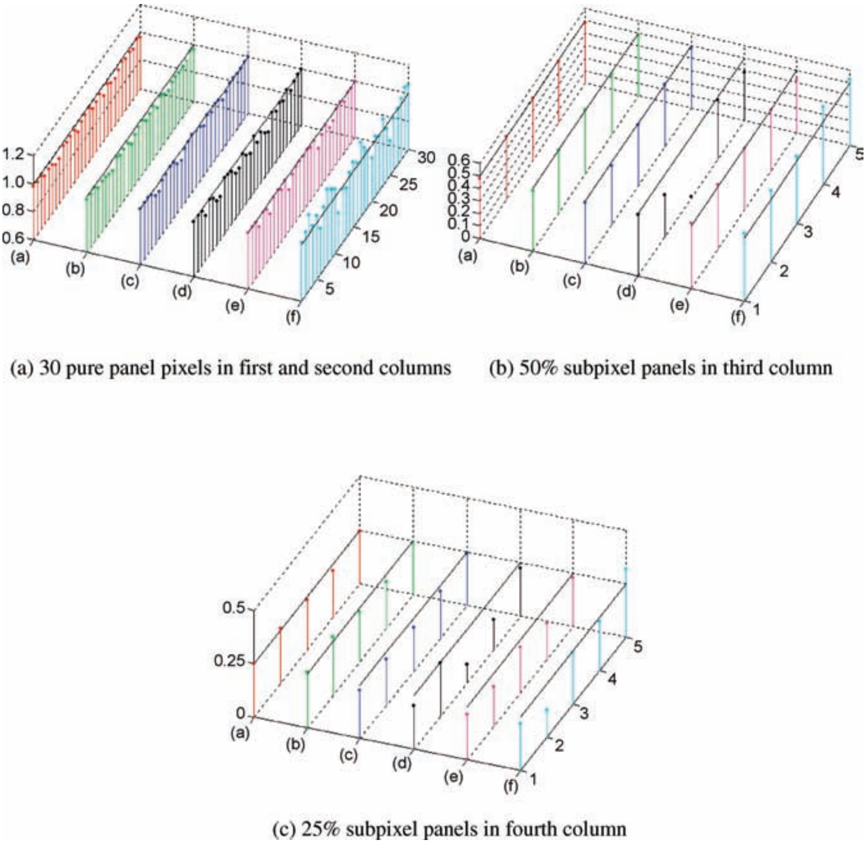
In order to produce a set of six desired signatures for the synthetic scene, the iterative N-finder algorithm (IN-FINDR) in Chapter 7 was used to find six endmembers, $\{\mathbf{e}_j\}_{j=1}^{6}$, as shown in Figure 14.5 that included five panel pixels specified by all the five different panel signatures, $\{\mathbf{p}_i\}_{i=1}^{5}$, and one background pixel.

The spectral signatures of these six pixels were then used to form the desired signature matrix **M**. In addition, according to Chapter 5 in Chang (2003a), the performance can be improved by eliminating interference prior to classification. In this case, ATGP was applied to find all potential interferers and terminated when a

Table 14.1 Quantitative results produced by the supervised MD-weighted AC-LSMA, LCMV-weighted AC-LSMA, $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA, OSP-weighted AC-LSMA, FCLS, and unconstrained LSOSP

First row	$p_{1,11}^1$	$p_{1,12}^1$	$p_{1,21}^1$	$p_{1,22}^1$	$p_{2,11}^1$	$p_{2,12}^1$	$p_{3,11}^1$	$p_{4,11}^1$
MD-weighted AC-LSMA	0.979	0.975	0.98	0.943	0.953	0.947	0.481	0.252
LCMV-weighted AC-LSMA	0.975	0.975	0.985	0.941	0.954	0.95	0.481	0.258
$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	0.989	0.998	0.997	0.981	0.933	0.927	0.482	0.226
OSP-weighted AC-LSMA	0.983	0.992	0.986	0.915	1	0.968	0.486	0.201
FCLS	0.994	1	0.991	0.926	0.991	0.972	0.519	0.211
unconstrained LSOSP	1.017	1.044	1.131	1.015	1.049	0.923	0.537	0.219
Second row	$p_{1,11}^2$	$p_{1,12}^2$	$p_{1,21}^2$	$p_{1,22}^2$	$p_{2,11}^2$	$p_{2,12}^2$	$p_{3,11}^2$	$p_{4,11}^2$
MD-weighted AC-LSMA	0.911	0.918	0.974	0.957	0.964	0.97	0.504	0.259
LCMV-weighted AC-LSMA	0.928	0.919	0.976	0.954	0.962	0.964	0.489	0.268
$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	0.972	0.977	0.997	0.99	0.941	0.926	0.47	0.216
OSP-weighted AC-LSMA	0.909	0.903	1	0.993	0.971	0.924	0.337	0.247
FCLS	0.902	0.983	1	0.968	0.997	0.892	0.52	0.185
unconstrained LSOSP	1.037	1.131	1.096	1.059	0.845	0.982	0.566	0.127
Third row	$p_{1,11}^3$	$p_{1,12}^3$	$p_{1,21}^3$	$p_{1,22}^3$	$p_{2,11}^3$	$p_{2,12}^3$	$p_{3,11}^3$	$p_{4,11}^3$
MD-weighted AC-LSMA	0.923	0.957	0.969	0.984	0.942	0.952	0.498	0.236
LCMV-weighted AC-LSMA	0.925	0.954	0.97	0.981	0.943	0.95	0.502	0.229
$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	0.976	0.992	1	0.99	0.984	0.953	0.487	0.207
OSP-weighted AC-LSMA	0.923	0.992	0.937	0.914	1	0.923	0.014	0.086
FCLS	0.962	0.983	0.984	0.981	0.995	0.971	0.491	0.215
Unconstrained LSOSP	0.955	1.028	1.064	1.02	1.132	1.058	0.531	0.241
Fourth row	$\mathrm{p}_{1,11}^4$	$\mathrm{p}_{1,12}^4$	$\mathrm{p}_{1,21}^4$	$\mathrm{p}_{1,22}^4$	$\mathrm{p}_{2,11}^4$	$\mathrm{p}_{2,12}^4$	$\mathrm{p}_{3,11}^4$	$\mathrm{p}_{4,11}^4$
MD-weighted AC-LSMA	0.968	0.921	0.97	0.941	0.933	0.958	0.481	0.215
LCMV-weighted AC-LSMA	0.968	0.919	0.972	0.943	0.934	0.957	0.48	0.216
$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	0.995	0.974	1	0.99	0.954	0.976	0.496	0.222
OSP-weighted AC-LSMA	0.958	0.914	0.882	0.988	0.96	0.984	0.475	0.139
FCLS	0.984	0.892	0.953	0.98	0.926	0.94	0.493	0.172
Unconstrained LSOSP	0.992	0.906	1.024	0.985	1.072	0.972	0.618	0.23
Fifth row	$p_{1,11}^5$	$p_{1,12}^5$	$p_{1,21}^5$	$p_{1,22}^5$	$p_{2,11}^5$	$p_{2,12}^5$	$p_{3,11}^5$	$p_{4,11}^5$
MD-weighted AC-LSMA	0.973	0.955	0.969	0.951	0.99	0.962	0.478	0.246
LCMV-weighted AC-LSMA	0.96	0.953	0.959	0.954	0.989	0.963	0.477	0.245
$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	0.999	1	0.996	0.996	1	1	0.477	0.243
OSP-weighted AC-LSMA	0.958	1	0.955	0.987	0.999	0.999	0.385	0.221
FCLS	0.989	1	0.964	0.993	0.987	1	0.419	0.229
Unconstrained LSOSP	0.848	0.966	1	1.039	1.079	1.053	0.526	0.317

warning sign of matrix singularity was flagged. Since some of the ATGP-generated target pixels may also be very similar or identical to IN-FINDR-generated endmembers, these ATGP-generated target pixels could not be considered interferers. Thus, when OSP-weighted AC-LSMA was implemented, the unwanted signature matrix **U** would be formed of all the ATGP-generated target pixels by excluding those that were also IN-FINDR generated endmembers. In this case, SAM was set to 0.04 to determine if an ATGP-target pixel was also an endmember. As a result, 22 interferers were found for OSP annihilation. On the other hand, when $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA was implemented, a set of training samples was required. In this case, SAM was set to 0.04 to find pixels that were similar to each of the six endmembers, $\{\mathbf{e}_j\}_{j=1}^{6}$, to form a set of training data for each of the p classes, $\{C_j\}_{j=1}^{6}$. Then, the means of each of the p classes were further calculated, $\{\boldsymbol{\mu}_j\}_{j=1}^{6}$, to form the desired signature matrix **M**. With knowledge provided by **M** and the training samples by $\{C_j\}_{j=1}^{6}$, six



(a) 30 pure panel pixels in first and second columns (b) 50% subpixel panels in third column

(c) 25% subpixel panels in fourth column

Figure 14.4 Graphical representation of abundance fractions of panel pixels in Figure 14.3 for visual assessment.

Table 14.2 VD estimated by the HFC and NWHFC methods with various false-alarm probabilities P_F

	$P_F = 10^{c1}$	$P_F = 10^{-2}$	$P_F = 10^{-3}$	$P_F = 10^{-4}$
HFC	15	7	5	4
NWHFC	6	6	6	6

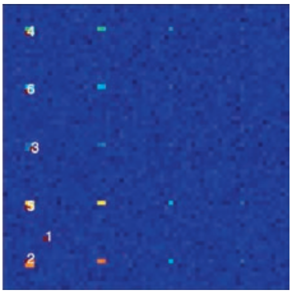
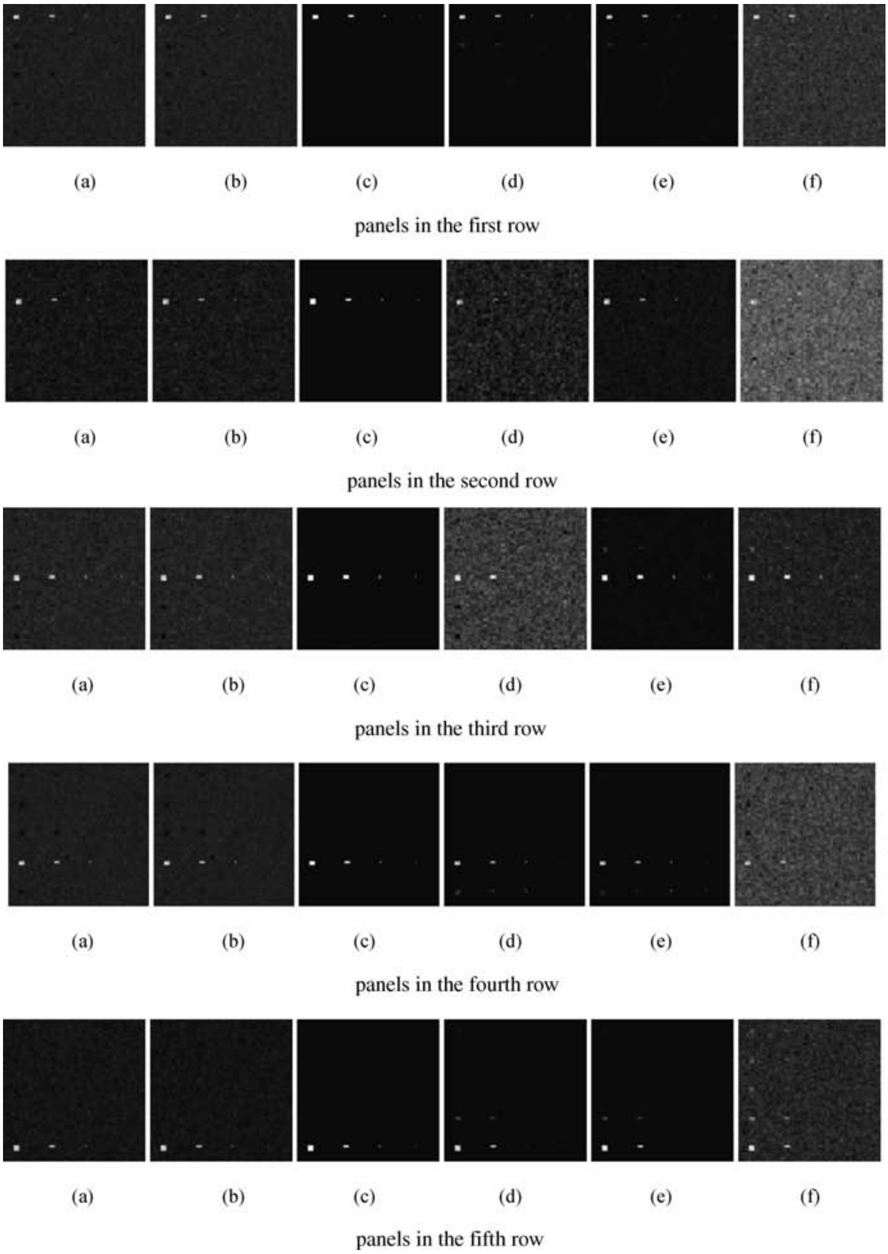


Figure 14.5 Six endmembers found by the N-FINDR.



panels in the first row

panels in the second row

panels in the third row

panels in the fourth row

panels in the fifth row

Figure 14.6 Abundance fraction results of 20 panels estimated by unsupervised five AC-LSMA methods and unconstrained LSOSP.

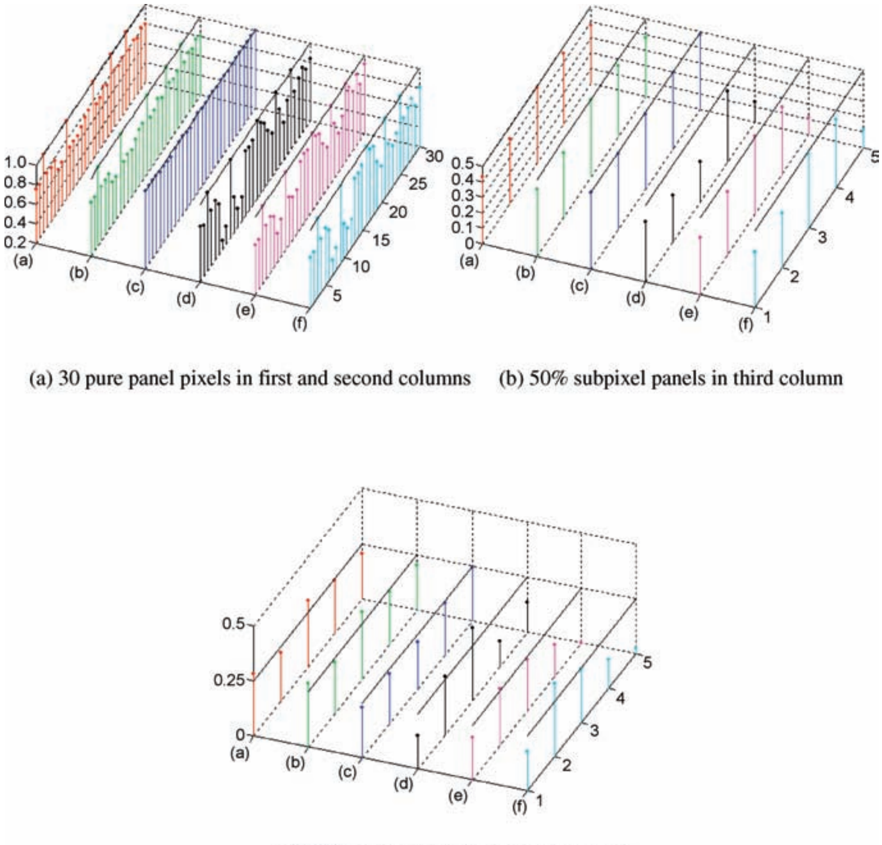
methods, MD-weighted AC-LSMA, LCMV-weighted AC-LSMA, $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA, OSP-weighted AC-LSMA, FCLS, and unconstrained LSOSP labeled by (a–f) were implemented for comparison. Figure 14.6 shows their corresponding abundance fraction results of the 20 panels in Figure 14.2(c) with full abundance constraints (i.e., ASC + ANC), respectively, where the $\mathbf{S}_{W}^{-1}$-weighted AC-LSMA clearly outperformed all the other five methods and the unconstrained LSOSP was the worst.

Table 14.3 Quantitative results produced by the unsupervised MD-weighted AC-LSMA, LCMV-weighted AC-LSMA, $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA, OSP-weighted AC-LSMA, FCLS, and unconstrained LSOSP, with endmembers generated by IN-FINDR

First row	$p_{1,11}^1$	$p_{1,12}^1$	$p_{1,21}^1$	$p_{1,22}^1$	$p_{2,11}^1$	$p_{2,12}^1$	$p_{3,11}^1$	$p_{4,11}^1$
MD-weighted AC-LSMA	0.752	0.746	1	0.745	0.746	0.757	0.431	0.283
LCMV-weighted AC-LSMA	0.75	0.746	1	0.745	0.753	0.76	0.433	0.288
$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	1	0.998	0.999	0.986	0.964	0.973	0.493	0.229
OSP-weighted AC-LSMA	0.768	0.723	1	0.761	0.809	0.713	0.392	0.149
FCLS	0.713	0.707	1	0.738	0.777	0.718	0.37	0.193
Unconstrained LSOSP	0.714	0.713	1	0.739	0.791	0.711	0.362	0.177
Second row	$\mathrm{p}_{1,11}^2$	$\mathrm{p}_{1,12}^2$	$\mathrm{p}_{1,21}^2$	$\mathrm{p}_{1,22}^2$	$\mathrm{p}_{2,11}^2$	$\mathrm{p}_{2,12}^2$	$\mathrm{p}_{3,11}^2$	$\mathrm{p}_{4,11}^2$
MD-weighted AC-LSMA	0.615	0.628	1	0.669	0.678	0.671	0.416	0.222
LCMV-weighted AC-LSMA	0.617	0.618	1	0.656	0.668	0.657	0.408	0.229
$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	0.963	0.952	0.986	0.993	0.992	0.977	0.488	0.226
OSP-weighted AC-LSMA	0.279	0.366	1	0.559	0.383	0.444	0.301	0.262
FCLS	0.488	0.576	1	0.693	0.631	0.65	0.407	0.257
Unconstrained LSOSP	0.345	0.413	1	0.547	0.448	0.482	0.353	0.33
Third row	$p_{1,11}^3$	$p_{1,12}^3$	$p_{1,21}^3$	$p_{1,22}^3$	$p_{2,11}^3$	$p_{2,12}^3$	$p_{3,11}^3$	$p_{4,11}^3$
MD-weighted AC-LSMA	0.711	0.754	0.766	1	0.713	0.726	0.492	0.302
LCMV-weighted AC-LSMA	0.711	0.754	0.766	1	0.713	0.725	0.493	0.302
$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	0.962	0.993	1	0.995	0.993	0.994	0.485	0.216
OSP-weighted AC-LSMA	0.859	0.842	0.858	1	0.923	0.855	0.262	0.33
FCLS	0.886	0.907	0.886	1	0.916	0.882	0.508	0.235
Unconstrained LSOSP	0.842	0.842	0.866	1	0.912	0.885	0.479	0.239
Fourth row	$p_{1,11}^4$	$p_{1,12}^4$	$p_{1,21}^4$	$p_{1,22}^4$	$p_{2,11}^4$	$p_{2,12}^4$	$p_{3,11}^4$	$p_{4,11}^4$
MD-weighted AC-LSMA	0.784	0.741	1	0.779	0.762	0.787	0.452	0.241
LCMV-weighted AC-LSMA	0.786	0.745	1	0.782	0.763	0.787	0.456	0.238
$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	0.999	0.983	1	0.987	0.979	0.995	0.496	0.235
OSP-weighted AC-LSMA	0.722	0.643	1	0.77	0.58	0.754	0.459	0.114
FCLS	0.723	0.63	1	0.757	0.604	0.72	0.44	0.148
unconstrained LSOSP	0.67	0.566	1	0.687	0.604	0.678	0.441	0.129
Fifth row	$p_{1,11}^5$	$p_{1,12}^5$	$p_{1,21}^5$	$p_{1,22}^5$	$p_{2,11}^5$	$p_{2,12}^5$	$p_{3,11}^5$	$p_{4,11}^5$
MD-weighted AC-LSMA	1	0.759	0.802	0.763	0.819	0.792	0.376	0.207
LCMV-weighted AC-LSMA	1	0.765	0.808	0.783	0.827	0.81	0.378	0.202
$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	1	1	0.992	0.996	0.997	1	0.48	0.24
OSP-weighted AC-LSMA	1	0.726	0.807	0.856	0.782	0.837	0.129	0.133
FCLS	1	0.902	0.806	0.917	0.76	0.911	0.109	0
unconstrained LSOSP	1	0.716	0.772	0.805	0.677	0.812	0.112	0.024

Table 14.3 tabulates the abundance fractions of the 20 panels obtained by the six methods (a)–(f) in Figure 14.6, using unsupervised knowledge for quantitative analysis.

According to the quantification results, the $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA in Figure 14.6(c) was the only one that produced most accurate abundance fractions of all the 20 panels and performed in a way that was very much comparable to its supervised counterpart in Example 14.1. Unfortunately, it was not the case for all other five methods in Figures 14.6(a) and (b) and 14.6(d)–(f) that apparently could not be compared with their supervised counterparts in Example 14.1. For a better visual assessment, Figure 14.7 graphically plots the abundance fractions of the 20 panels in Table 14.1.



(a) 30 pure panel pixels in first and second columns (b) 50% subpixel panels in third column

(c) 25% subpixel panels in fourth column

Figure 14.7 Graphical representation of abundance fractions of panel pixels in Figure 14.6 for visual assessment.

14.5 Real Image Experiments

In this section, the 15-panel HYDICE image scene in Figure 1.15(a) was used for experiments. One major difference between the real HYDICE image scene in Figure 1.15(a) and the simulated synthetic image in Figure 14.1(c) was that very little knowledge of the image background in Figure 1.15(a) was known compared to the image background in Figure 14.1(b), which was simulated by complete knowledge. As we may expect, an AC-LSMA classifier may not perform as well as it did for the synthetic image if the image background in Figure 1.15(a) was not well characterized. In order to demonstrate this fact, two scenarios were used to characterize the image background as follows. In addition, we also assumed that knowledge of the 9 R pixels in the 3 m × 3 m and 5 R pixels in the 2 m × 2 m panels in Figure 1.15(b) was available *a priori*. This is because the panels in the 3rd column are subpixel panels which cannot be visualized to obtain as priori knowledge. Thus, the panel signatures in Figure 1.16 and 14 R pixels in both the 3 m × 3 m and 2 m × 2 m panels were considered to be prior knowledge.

By viewing the scene in Figure 14.2, a large portion of the image background is formed of one-fourth of a forest on the left and three-fourth of a large grass field. Using this supervised knowledge, we conducted two experiments to represent the image background. One experiment was to use the area A to characterize the image background. In this case, a single-background signature $\mathbf{b}_{\mathrm{A}}$ was used for experiments and the training samples used for $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA were all the pixels in the area A for one background class, as done in Example 14.1. Another scenario was to use the automatic target generation process (ATGP) to produce necessary background knowledge in an unsupervised manner.

EXAMPLE 14.3

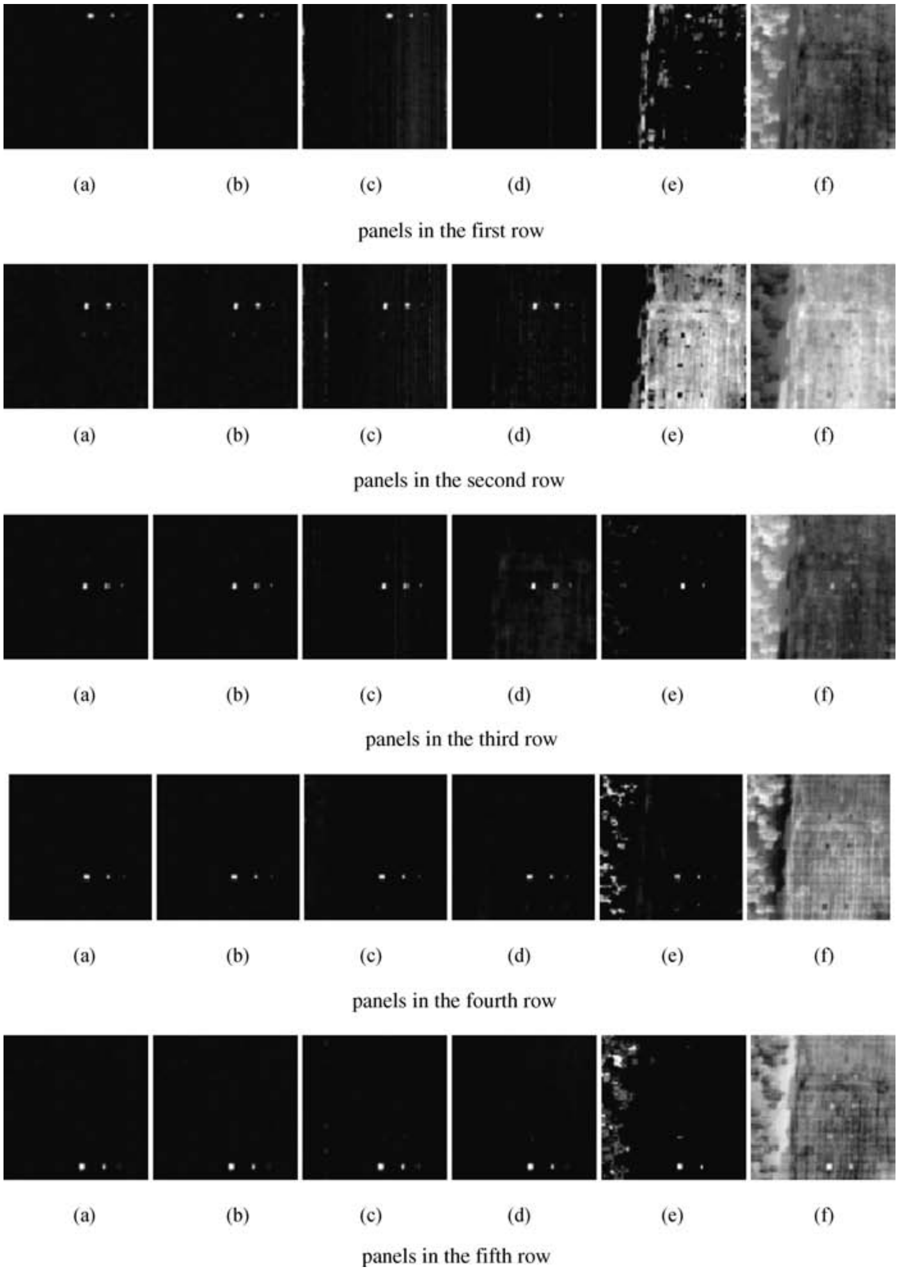
(Scenario 1: Single-Background Signature)

Like Example 1, the signature matrix M used for experiments is formed by $\mathbf{M} = [\mathbf{p}_1\mathbf{p}_2\mathbf{p}_3\mathbf{p}_4\mathbf{p}_5\mathbf{b}_{\mathrm{A}}]$. The 14 R pixels in both the 3 m × 3 m and 2 m × 2 m panels and pixels in the area A provided training samples for $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA. Six methods labeled by five AC-LSMA methods, MD-weighted AC-LSMA, LCMV-weighted AC-LSMA, $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA, OSP-weighted AC-LSMA, FCLS, and unconstrained LSOSP were evaluated for comparative analysis. Figure 14.8 shows their respective abundance fraction results of the 15 panels in Figure 14.2(c) with full abundance constraints (i.e., ASC + ANC), respectively.

Once again, the unconstrained LSOSP was the worst and the three weighted AC LSMA, MD-weighted AC-LSMA, LCMV-weighted AC-LSMA, and $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA seemed among the best. To further justify our conclusions, Table 14.4 tabulates the quantification results obtained for the abundance fractions of the 14 pure R panel pixels (i.e., $p_{11}, p_{12}, p_{211}, p_{221}, p_{22}, p_{311}, p_{312}, p_{32}, p_{411}, p_{412}, p_{42}, p_{511}, p_{521}, p_{52}$) and the 5 R panel subpixels (i.e., $p_{13}, p_{23}, p_{33}, p_{43}, p_{53}$) in Figure 14.8(a)–(f). The quantification results show that the $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA in Figure 14.8(c) was the best in the sense that it produced the most accurate abundance fractions of the 19 panel pixels.

Table 14.4 Quantitative results produced by the MD-weighted AC-LSMA, LCMV-weighted AC-LSMA, $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA, OSP-weighted AC-LSMA, FCLS, and unconstrained LSOSP with a single-background signature

	MD-weighted AC-LSMA	LCMV-weighted AC-LSMA	$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	OSP-weighted AC-LSMA	FCLS	Unconstrained LSOSP
p_{11}	1	1	1	0.971	0.762	1.284
p_{12}	0.731	0.731	0.85	0.654	0.581	0.716
p_{13}	0.186	0.187	0.29	0.169	0.074	0.322
p_{211}	1	1	1	0.961	0.909	1.175
p_{221}	0.994	0.991	1	0.945	0.788	1.243
p_{22}	0.968	0.969	0.962	0.851	0.773	0.582
p_{23}	0.25	0.251	0.324	0.303	0.758	0.887
p_{311}	0.993	0.994	0.987	1	0.941	1.29
p_{312}	1	1	1	1	0.906	1.468
p_{32}	0.733	0.733	0.845	0.709	0.17	0.243
p_{32}	0.393	0.397	0.438	0.448	0	0.002
p_{411}	1	1	1	1	0.509	1.193
p_{412}	1	1	0.999	0.815	0.655	0.85
p_{42}	0.912	0.913	0.929	0.859	0.798	0.957
p_{43}	0.184	0.183	0.224	0.251	0.09	0.744
p_{511}	0.874	0.874	0.894	0.862	0.823	0.92
p_{521}	1	1	1	1	1	1.253
p_{52}	0.901	0.901	0.956	0.879	0.894	0.828
p_{53}	0.166	0.167	0.211	0.122	0	−0.238



(a) (b) (c) (d) (e) (f)

panels in the first row

(a) (b) (c) (d) (e) (f)

panels in the second row

(a) (b) (c) (d) (e) (f)

panels in the third row

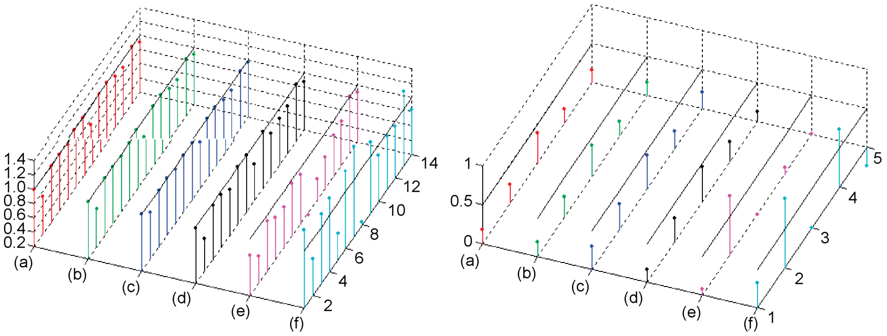
(a) (b) (c) (d) (e) (f)

panels in the fourth row

(a) (b) (c) (d) (e) (f)

panels in the fifth row

Figure 14.8 15-panel abundance fraction results of the supervised five AC-LSMA methods and unconstrained LSOSP.



(a) 14 R pure panel pixels in first and second columns (b) 5 R subpixel panels in third column

Figure 14.9 Graphical representation of abundance fractions of 19 R panel pixels in Figure 14.8 for visual assessment.

Figure 14.9 provides the graphical plots of quantification values in Table 14.4 for an easy visual assessment where the visual inspection of Figure 14.8 may not provide reliable quantification estimates of abundance fractions.

EXAMPLE 14.4

(Scenario 2: Unsupervised Background Knowledge)

As demonstrated in Example 14.3, a single-background signature could not completely characterize the image background. As a result, AC-LSMA performance was not as good as it did for the synthetic image in Example 14.1, where only one signature was used to simulate the image background. In order to improve its performance, we need to find an appropriate set of background pixels that can well represent the image background. According to VD, the number of spectrally distinct signatures in the scene in Figure 14.1 is $n_{\mathrm{VD}} = 9$, which was used as the number of endmembers for **M**. To determine additional background signatures, the number of total signatures in the scene was set to $2n_{\mathrm{VD}} = 18$. A reason for selecting $2n_{\mathrm{VD}}$ will be explained and provided in Chapter 17 as well as Chapter 22. This implies that we need at least 13 spectrally distinct signatures to characterize the image background in addition to the five panel signatures in Figure 14.1. In this case, IN-FINDR was applied to find the 18 endmembers, $\{\mathbf{e}_j\}_{j=1}^{18}$, as shown in Figure 14.10 to form the desired signature matrix **M**, where the

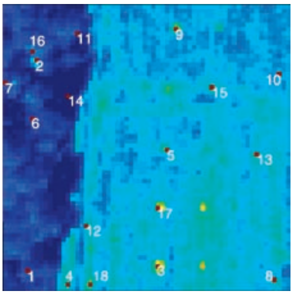
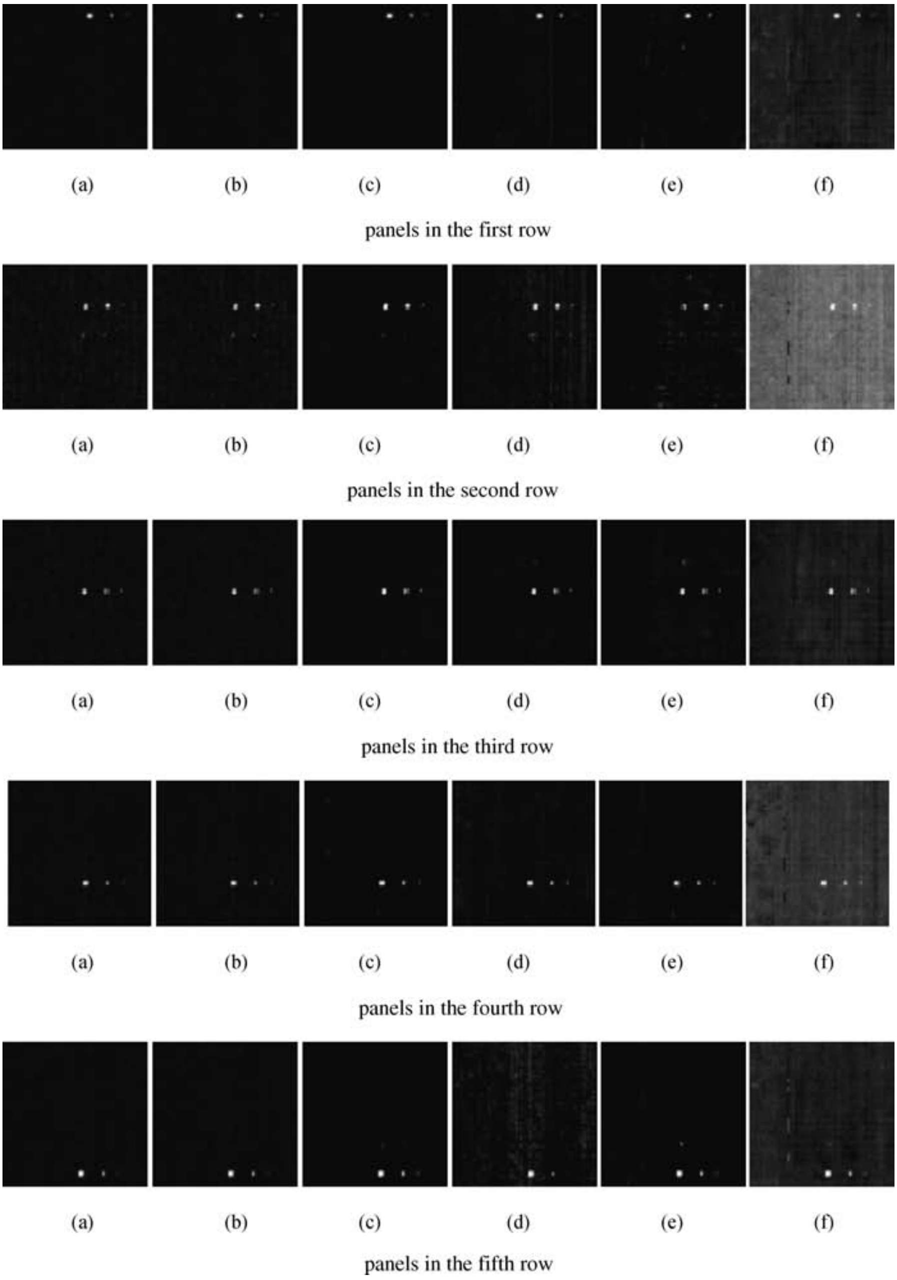


Figure 14.10 Eighteen endmembers produced by IN-FINDR.



panels in the first row

panels in the second row

panels in the third row

panels in the fourth row

panels in the fifth row

Figure 14.11 Unsupervised 15-panel abundance fraction results of five AC-LSMA methods and unconstrained LSOSP.

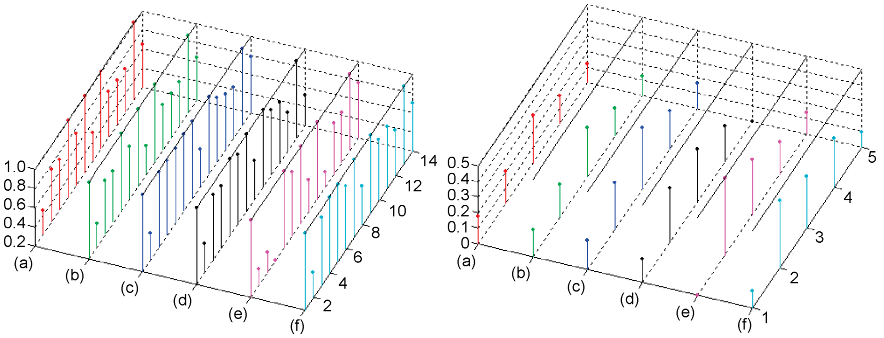
found pixels labeled by numbers 3, 5, 9, 15, and 17 in Figure 14.10 represented five panel signatures in five different rows in Figure 1.15(a) or Figure 14.1.

In analogy with Example 14.2, ATGP was also implemented to find potential interferers until a warning sign of matrix singularity was flagged. In our experiments, there are 169 target pixels. Since some of such

Table 14.5 Quantitative results produced by the MD-weighted AC-LSMA, LCMV-weighted AC-LSMA, $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA, OSP-weighted AC-LSMA, FCLS, and unconstrained LSOSP with endmembers generated by N-FINDR

	MD-weighted AC-LSMA	LCMV-weighted AC-LSMA	$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	OSP-weighted AC-LSMA	FCLS	Unconstrained LSOSP
p_{11}	1	1	1	1	1	1
p_{12}	0.443	0.442	0.477	0.492	0.361	0.461
p_{13}	0.175	0.175	0.185	0.149	0	0.109
p_{211}	0.75	0.756	0.986	0.879	0.406	0.903
p_{221}	0.723	0.729	0.977	0.845	0.2	0.991
p_{22}	1	1	0.978	1	1	1
p_{22}	0.205	0.204	0.3	0.349	0.494	0.436
p_{311}	0.735	0.737	0.998	0.92	0.865	0.855
p_{312}	1	1	1	1	1	1
p_{32}	0.494	0.491	0.583	0.601	0.532	0.6
p_{32}	0.308	0.31	0.395	0.339	0.357	0.331
p_{411}	1	1	1	1	1	1
p_{412}	0.666	0.666	0.871	0.872	0.36	0.83
p_{42}	0.655	0.655	0.772	0.831	0.738	0.837
p_{43}	0.175	0.175	0.241	0.229	0.212	0.315
p_{511}	0.648	0.648	0.725	0.586	0.719	0.67
p_{521}	1	1	1	1	1	1
p_{52}	0.646	0.646	0.789	0.516	0.778	0.697
p_{53}	0.123	0.123	0.161	0	0.14	0.095

ATGP-generated target pixels may also be very similar or identical to the 18 IN-FINDR generated endmembers, these ATGP-generated target pixels could not be considered interferers. So, when OSP-weighted AC-LSMA was implemented, the unwanted signature matrix **U** would be formed of all the ATGP-generated target pixels, except those that were also IN-FINDR-generated endmembers. In this case, SAM was set to 0.06 to determine if an ATGP-target pixel was also an endmember. On the other hand, $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA required a set of training samples. In this case, SAM was set to 0.025 to find pixels that were similar to each of the 18 endmembers, $\{\mathbf{e}_j\}_{j=1}^{18}$, to form a set of training data for each of the 18 classes, $\{C_j\}_{j=1}^{18}$, where the



(a) 14 R pure panel pixels in first and second columns (b) 5 R panel subpixels in third column

Figure 14.12 Graphical representation of abundance fractions of 19 R panel pixels in Figure 1(b) for visual assessment.

total number of found training samples was 3714. Then, the means of training samples in each of the 18 classes were further calculated, $\{\boldsymbol{\mu}_j\}_{j=1}^{18}$, to form the desired signature matrix $\mathbf{M}$. Six methods, AC-LSMA methods, MD-weighted AC-LSMA, LCMV-weighted AC-LSMA, $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA, OSP-weighted AC-LSMA, FCLS, and unconstrained LSOSP labeled by (a-f) were evaluated for comparative analysis.

Figure 14.11 shows their respective abundance fraction results of the 15 panels in Figure 14.1 with full abundance constraints (i.e., ASC + ANC), where the $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA was the best compared to the unconstrained LSOSP that was the worst. Unlike Example 14.3 that only used one background signature, the use of additional 12 background signatures to find training samples made a significant difference for the $\mathbf{S}_{W}^{-1}$-weighted AC-LSMA. As shown in Figure 14.11, the $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA labeled by (c) clearly outperformed all other five AC-LSMA methods. Interestingly, the FCLS seemed to perform well in detection of 19 R panel pixels visually shown in Figure 14.11 at the expense of many falsely alarmed pixels. However, their quantitative results tabulated in Table 14.5 show otherwise. Figure 14.12(a) and (b) plots quantified abundance fractions of 14 R pure panel pixels and 5 R panel subpixels in Figure 14.11, respectively, where the MD-weighted AC-LSMA and LCMV-weighted AC-LSMA labeled by (a) and (b) unmixed abundance fractions more accurately than FCLS for most panel pixels. Nevertheless, the $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA from Figure 14.11(c) was still the best according to Table 14.5 in terms of quantifying abundance fractions of panel pixels.

Figure 14.12 further provides the graphical plots of quantification values in Table 14.5 for a better visual assessment compared to the results in Figure 14.11.

A remark on the threshold used for SAM is noteworthy. This threshold was selected empirically for SAM in our experiments. It is based on our experience gained while working on laboratory and real data. Since laboratory data are generally used for simulations, its tolerance to the threshold is more robust than real data. Thus, the threshold selected for simulations can be higher than that chosen for real data. The interval of [0.02, 0.03] for simulated data and the interval of [0.03, 0.05] for real data seemed reasonable ranges from which a threshold can be selected. As for the threshold used by SAM to find undesired signatures for OSP-weighted LSMA, it was set to 0.06 that was a little bit higher than the thresholds used to find endmembers. This is because undesired signatures are not necessarily as subtle as endmembers are. Nonetheless, the threshold selection is generally sensitive to spectral characteristics of signatures to be analyzed. It is advised that several trial-and-errors of selecting different values in this range may be worthwhile.

As a concluding comment, despite the fact that $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA was shown to be the best among the six evaluated methods, it required a good set of training samples to produce the within-class matrix $\mathbf{S}_{\mathrm{W}}$. If the sample pool is not well representative like Example 14.3, it will not perform effectively. To the contrary, if the training samples are selected judiciously as the way was done in Example 14.4, $\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA could be one of the best AC-LSMA methods. Finally, the threshold values used in our experiments for SAM were not optimal, but rather empirical selections.

14.6 Conclusions

Abundance-constrained linear spectral mixture analysis (AC-LSMA) using LSE as a criterion has been studied extensively in the literature. It is, in general, referred to as least-squares AC-LSMA. However, including a weighting matrix in the least-squares AC-LSMA to account for

Table 14.6 Summary of unweighted AC-LSMA (FCLS) and four weighted AC-LSMA methods

	Weighting matrix $\mathbf{A}$	Training samples	Signature matrix $\mathbf{M}$	Undesired signature matrix $\mathbf{U}$
Unweighted AC-LSMA (FCLS)	$\mathbf{I}$	No	Yes	No
MD-weighted AC-LSMA	$\mathbf{K}^{-1}$	No	Yes	No
LCMV-weighted AC-LSMA	$\mathbf{R}^{-1}$	No	Yes	No
$\mathbf{S}_{\mathrm{W}}^{-1}$-weighted AC-LSMA	$\mathbf{S}_{\mathrm{W}}^{-1}$	Yes	Yes	No
OSP-weighted AC-LSMA	$P_{\mathbf{U}}^{\perp}$	No	Yes	Yes

significance of individual bands has not been explored in the past few years until Chang and Ji, (2006a). This chapter investigates weighted AC-LSMA in terms of LSE and further develops three approaches to weighted AC-LSMA, each of which can be obtained by the commonly used criteria, Mahalanobis distance or Gaussian maximum likelihood estimation, Fisher's ratio, and OSP. In particular, the least-squares AC-LSMA can be considered an unweighted AC-LSMA. The experimental results demonstrate that weighted AC-LSMA generally performs better than unweighted AC-LSMA.

Finally, we summarize the advantages and disadvantages of the unweighted AC-LSMA (i.e., FCLS), along with all the four weighted AC-LSMA methods considered in Table 14.6 in this chapter.

15

Kernel-Based Linear Spectral Mixture Analysis

Linear spectral mixture analysis (LSMA) has been widely used in remote sensing community for spectral unmixing and has enjoyed great success in material detection, classification, and identification. Recently, it has been extended in various approaches to linear spectral random mixture analysis (Chapter 15, Chang 2003a), Fisher's LSMA (FLSMA) in Chapter 13 and weighted abundance-constrained LSMA (WAC-LSMA) in Chapter 14 so as to enhance its performance. All these extensions also inherit drawbacks and limitations of LSMA when it comes to solving linear nonseparable problems. This chapter develops a kernel-based LSMA (KLSMA) to resolve the issue of nonlinear separability. By mapping the original data to a feature space via a nonlinear kernel, LSMA extensions can be further expanded to their kernel-based counterparts, specifically, the three backbone least squares-based LSMA (LS-LSMA) techniques (Chang 2003a), least squares orthogonal subspace projection (LSOSP), non-negativity constrained least squares (NCLS), and fully constrained least squares (FCLS), are readily extended to their corresponding kernel versions, KLSOSP, KNCLS, and KFCLS. Interestingly, according to experiments conducted based on synthetic and real images, KLSMA can be more effective than LSMA only for cases where data sample vectors are heavily mixed, specifically for multispectral imagery which will be discussed in Chapters 31 and 32.

15.1 Introduction

Linear mixture analysis is a theory developed for solving linear problems. It has found many successes in a wide range of applications, such as linear regression analysis in multivariate data analysis, blind source separation in signal processing, and partial volume estimation in magnetic resonance imaging (see Chapter 32). Specifically, LSMA has been widely used in remote sensing community to perform spectral unmixing (Chapters 12–14) where a data sample vector is linearly mixed by a number of so-called endmembers as a linear mixture from which it can be further unmixed as abundance fractions in terms of these endmembers. Using the same notations in Section 12.2 and Eq. (12.2) let $\mathbf{r}$ be an L-dimensional data sample vector and $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ be signatures of interest that are used to

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

unmix the sample vector $\mathbf{r}$. To carry out spectral unmixing, a linear mixing model is required to represent $\mathbf{r}$ in terms of the following form:

$$\mathbf{r} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n} \tag{15.1}$$

where $\mathbf{M} = \left[\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p\right]$ is a signature matrix, $\mathbf{n}$ is a noise vector and can be used to describe a model or measurement error, and $\boldsymbol{\alpha} = \left(\alpha_1, \alpha_2, \ldots, \alpha_p\right)^T$ is an unknown p-dimensional abundance vector needed to be found and is associated with $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ with α_j representing the abundance fraction of the jth endmember $\mathbf{m}_j$ present in the sample vector $\mathbf{r}$. Due to physical constraints, two abundance constraints are generally imposed on (15.1), which are abundance sum-to-one constraint (ASC) specified by $\sum_{j=1}^{p} \alpha_j = 1$ and abundance non-negativity constraint (ANC) specified by $\alpha_j \geq 0$ for all $1 \leq j \leq p$. The LSMA develops techniques to perform the so-called linear spectral unmixing (LSU) that solves a linear inverse problem of (15.1) by unmixing a data sample vector $\mathbf{r}$ via a set of p endmembers, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$, through finding their respective abundance fractions $\alpha_1, \alpha_2, \ldots, \alpha_p$ with/without the abundance constraints, ASC, and ANC.

Over the past years, LSMA has been extended in various directions to enhance its capability in spectral unmixing. For example, in order for LSMA to deal with random signals, a random version of LSMA called linear spectral random mixture analysis (LSRMA) based on projection pursuit was developed in Chang et al. (2002) and Chang (2003a). Since LSE is generally not an optimal criterion used for classification, a different version of LSMA based on Fisher's ratio, called FLSMA, is also derived from Fisher's linear discriminant analysis in Chapter 13 (Chang and Ji, 2006b). Interestingly, it is also shown in Chapter 14 as well as in Chang and Ji (2006a) that introducing a weighting matrix into the LSE criterion results in a new LSMA technique, called WAC-LSMA that includes LS-LSMA and FLSMA as its special cases. However, all such extensions intend to increase and enhance their ability for linear separability. Unfortunately, due its nature in inherent constraints resulting from the use of a linear mixing model this is probably the best we can do with LSMA without going beyond linear approaches. To resolve this dilemma two approaches seem feasible. One is to directly use a nonlinear mixture model, called intimate spectral mixture (Hapke, 1981) to perform spectral unmixing. Such an approach was investigated in Guilfoyle et al. (2001) and Guilfoyle (2003) where radial basis function (RBF) neural networks were used to approximate the unknown parameters used in a nonlinear mixing model. The other approach can be considered as a compromise between linear and nonlinear models. It maps the non-linear decision boundaries made by a classifier via a nonlinear function into a generally high-dimensional feature space in which non-linear separability problems can be solved by linear decisions. The use of such a nonlinear function is similar to nonlinear activation functions used in neural networks for network's internal learning. The approach of this type is known as a kernel-based technique where nonlinear functions are modeled as nonlinear kernels. Interestingly, using kernels to extend the LSMA has not received much attention until a kernel approach developed by Kwon and Nasrabadi (2005) who extended the orthogonal subspace projection (OSP) developed by Harsanyi and Chang (1994) to its kernel counterpart, called kernel-based OSP (KOSP). Later at nearly the same time another LSMA technique, NCLS, and FCLS were further extended to their counterparts, called kernel-based NCLS (KNCLS) and kernel-based FCLS (KFCLS) in Broadwater et al. (2007) and Liu et al. (2009). It should be noted that when the versions of KNCLS and KFCLS were derived in Liu et al. (2009) the details of KNCLS and KFCLS in Broadwater et al. (2007) were not available at that time but only published later in a book chapter (Camps-Valls and L. Bruzzone, 2009). The detailed derivations in Liu et al. (2009) provide a direct extension of

LSOSP, NCLS, and FCLS to their respective kernel counterparts. In essence, the works reported in Broadwater et al. (2007), Camps-Valls and L. Bruzzone (2009), and Liu et al. (2009) are actually derived independently. Nevertheless, this chapter along with Liu et al. (2012) are believed to be the first that derives kernel counterparts of all the three backbone LS-LSMA techniques, LSOSP, NCLS, and FCLS, in a unified framework.

Despite that kernel-based approaches have shown promising in many remote sensing applications it does not imply that KLSMA always has advantages over LSMA in hyperspectral image analysis. As a matter of fact, it will be shown that kernel-based techniques do not necessarily improve classification performance for hyperspectral digital imagery collection experiment (HYDICE) data. This gives rise to an interesting question: Under what circumstances will KLSMA techniques be effective when they are used for spectral unmixing? This chapter tries to answer this question by providing experiments conducted based on two data sets, the visible/infrared imaging spectrometer (AVIRIS) Purdue Indiana Indian Pine data and the HYDICE data to demonstrate that kernel-based approaches can significantly improve performance if hyperspectral data have low spatial resolution such as AVIRIS data, but they cannot do the same for high spatial resolution HYDICE even both data sets do have the same spectral resolution.

15.2 Kernel-Based LSMA (KLSMA)

This section revisits and extends the three least squares-based techniques, OSP/LSOSP, NCLS, and FCLS, that have shown success in hyperspectral unmixing (Chang, 2003a) to their corresponding kernel-based counterparts, each of which represents three categories of techniques implementing LSMA. One is the category of abundance-unconstrained methods that include the well-known Gaussian maximum likelihood (GML) estimation and OSP, both of which have been shown essentially the same technique in Chapter 12. Another is the category of partially abundance-constrained methods, of which the NCLS method developed for constrained signal detection by Chang and Heinz (2000) is a representative. A third category of fully abundance-constrained methods among which the fully constrained least squares (FCLS) method developed by Heinz and Chang (2001) is the most widely used method for this purpose. So, when it comes to extend LSMA to kernel-based LSMA (KLSMA), it is natural to consider these three methods for kernel extension. The works in Kwon and Nasrabadi (2005b), Broadwater et al. (2007), and Liu and Chang (2009) are early attempts to accomplish this goal.

In what follows, we follow the work in Liu et al. (2012) to develop a unified kernel theory for extending LSMA to KLSMA by first developing the kernel-based LSOSP (KLSOSP) that is derived directly from the structure of the OSP. This KLSOSP is then used to derive a KNCLS that is in turn to be used to further derive KFCLS. It is our belief that this section provides most detailed derivations for kernel versions of the four LSMA, OSP, LSOSP, NCLS, and FCLS, including step-by-step algorithmic implementations.

15.2.1 Kernel Least Squares Orthogonal Subspace Projection (KLSOSP)

A kernel version of OSP was first derived in Kwon and Nasrabadi (2005b) for hyperspectral LSU. The idea is to use single value decomposition (SVD) to partition the undesired signature matrix $\mathbf{U}$ as $\mathbf{U} = \mathbf{BDC}^T$ so that $P_{\mathbf{U}}^{\perp}$ in (2.78) or (12.3) can be decomposed and simplified to

$$\begin{aligned} P_{\mathbf{U}}^{\perp} &= \mathbf{I} - \mathbf{UU}^{\#} = \mathbf{I} - \mathbf{U}\left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T \\ &= \mathbf{I} - \mathbf{BDC}^T\left(\mathbf{CD}^T\mathbf{DC}^T\right)^{-1}\mathbf{CD}^T\mathbf{B}^T = \mathbf{I} - \mathbf{BB}^T \end{aligned} \tag{15.2}$$

where **I** is an $L \times L$ identity matrix, **B**, **C** are orthogonal matrices, and **D** is a diagonal matrix. The purpose of (15.2) is to form dot products so that the kernel trick (Hofmann et al., 2008) can be applied without a need for performing matrix inversion. Since the used kernel mapping function is a positive-definite kernel, so is the output mapping matrix. As a result, the matrix resulting from the kernel trick should be in fact nonsingular and invertible. Using this fact, the kernel trick can be directly applied to $P_{\mathbf{U}}^{\perp}$ specified by (15.2) without using SVD where $\left(K(\mathbf{U}^T\mathbf{U})\right)^{-1}$ is obtained from mapping $(\mathbf{U}^T\mathbf{U})^{-1}$ in (15.2) into a feature space via a positive definite kernel and is always invertible. Consequently, a much simpler approach can be derived as follows.

First, let a nonlinear kernel be specified by ϕ. Then $P_{\mathbf{U}}^{\perp}$ in (15.2) mapped into the feature space can be expressed as

$$\begin{aligned} P_{\phi(\mathbf{U})}^{\perp} &= \mathbf{I}_{\phi_{L\times L}} - \phi(\mathbf{U})\phi(\mathbf{U})^{\#} \\ &= \mathbf{I}_{\phi_{L\times L}} - \phi(\mathbf{U})\left(\phi(\mathbf{U})^T\phi(\mathbf{U})\right)^{-1}\phi(\mathbf{U})^T \end{aligned} \tag{15.3}$$

Now, $\hat{\alpha}_p^{\mathrm{OSP}} = \mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r}$ specified by (12.9) is mapped into a kernel version of OSP induced by ϕ in a feature space as

$$\begin{aligned} \alpha_p^{\mathrm{KOSP}}(\mathbf{r}) &= \phi(\mathbf{d})^T P_{\phi(\mathbf{U})}^{\perp}\phi(\mathbf{r}) \\ &= \phi(\mathbf{d})^T\mathbf{I}_{\Phi_{L\times L}}\phi(\mathbf{r}) - \phi(\mathbf{d})^T\phi(\mathbf{U})\left(\phi^T(\mathbf{U})\phi(\mathbf{U})\right)^{-1}\phi(\mathbf{U})^T\phi(\mathbf{r}) \\ &= \phi(\mathbf{d})\phi(\mathbf{r}) - \phi(\mathbf{d})^T\phi(\mathbf{U})\left(\phi(\mathbf{U})^T\phi(\mathbf{U})\right)^{-1}\phi(\mathbf{U})^T\phi(\mathbf{r}) \end{aligned} \tag{15.4}$$

where $P_{\phi(\mathbf{U})}^{\perp}$ in (15.3) is used to derive the second equality. It should be noted that the identity mapping $\phi(\mathbf{d})^T\mathbf{I}_{\phi_{L\times L}}\phi(\mathbf{r})$ in (15.4) can be simplified into $\phi(\mathbf{d})^T\phi(\mathbf{r})$ based on the bilinearity properties from which each kernel function inherits according to Hofmann et al.'s work (Hofmann et al. 2008). That is,

$$\langle K(\mathbf{d}, \mathbf{I}_{\Phi_{L\times L}}), K(\mathbf{I}_{\Phi_{L\times L}}, \mathbf{r})\rangle = K(\mathbf{d}, \mathbf{r}) \tag{15.5}$$

Using (15.5) and the kernel trick described by

$$K(\mathbf{x}, \mathbf{y}) = \langle\phi(\mathbf{x}), \phi(\mathbf{y})\rangle = \phi(\mathbf{x})^T\phi(\mathbf{y}) = \phi(\mathbf{x})\cdot\phi(\mathbf{y}) \tag{15.6}$$

where K is a kernel function defined in the original data, (15.4) can be rewritten as

$$\alpha_p^{\mathrm{KOSP}}(\mathbf{r}) = K(\mathbf{d}, \mathbf{r}) - K(\mathbf{d}, \mathbf{U})K(\mathbf{U}, \mathbf{U})^{-1}K(\mathbf{U}, \mathbf{r}) \tag{15.7}$$

Since $\hat{\alpha}_p^{\mathrm{OSP}}$ is obtained by a matched filter designed for signal detection not estimation, OSP is further extended to LSOSP (Tu et al., 1997) as abundance estimator as $\hat{\alpha}_p^{\mathrm{LSOSP}}$ by including an abundance correction term, $\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$ in $\hat{\alpha}_p^{\mathrm{OSP}}$ as

$$\hat{\alpha}_p^{\mathrm{LSOSP}} = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} \tag{15.8}$$

So, by taking advantage of (15.7) a kernel-based LSOSP (KLSOSP) can be further derived as

$$\alpha_p^{\text{KLSOSP}}(\mathbf{r}) = \frac{\alpha_p^{\text{KOSP}}(\mathbf{r})}{K(\mathbf{d},\mathbf{r}) - K(\mathbf{d},\mathbf{U})K(\mathbf{U},\mathbf{U})^{-1}K(\mathbf{U},\mathbf{d})} \tag{15.9}$$

which is not derived in Kwon and Nasrabadi (2005b).

15.2.2 Kernel-Based Non-Negative Constraint Least Square (KNCLS)

One of the major issues in implementing OSP is that it produces negative values for abundance fractions that are supposed to be non-negative. To mitigate this problem, ANC must be imposed as part of solving the optimization problem. An approach, called abundance NCLS, was recently developed by Chang and Heinz (2000). It minimizes an objective function, $(\mathbf{M}\boldsymbol{\alpha} - \mathbf{r})^T(\mathbf{M}\boldsymbol{\alpha} - \mathbf{r})$ by constraining $\alpha_j \geq 0$ for $j \in \{1, 2, \ldots, p\}$. In doing so it introduces a Lagrange multiplier vector $\boldsymbol{\lambda} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_p \end{bmatrix}^T$ in the following constrained optimization problem:

$$J = \frac{1}{2}(\mathbf{M}\boldsymbol{\alpha} - \mathbf{r})^T(\mathbf{M}\boldsymbol{\alpha} - \mathbf{r}) + \boldsymbol{\lambda}(\boldsymbol{\alpha} - \mathbf{c}) \tag{15.10}$$

where $\mathbf{c} = \begin{bmatrix} c_1 & c_2 & \ldots & c_p \end{bmatrix}^T$ and c_j for $1 \leq i \leq p$ are constraints. With $\boldsymbol{\alpha} = \mathbf{c}$ it follows that

$$\hat{\boldsymbol{\alpha}} = (\mathbf{M}^T\mathbf{M})^{-1}\mathbf{M}^T\mathbf{r} - (\mathbf{M}^T\mathbf{M})^{-1}\boldsymbol{\lambda} \tag{15.11}$$

and

$$\boldsymbol{\lambda} = \mathbf{M}^T\mathbf{r} - \mathbf{M}^T\mathbf{M}\hat{\boldsymbol{\alpha}} \tag{15.12}$$

Since there is no closed form that can be derived from (15.10), we ought to rely on a numerical algorithm for finding an optimal abundance vector $\hat{\boldsymbol{\alpha}}^{\text{NCLS}}(\mathbf{r})$ for the abundance vector $\boldsymbol{\alpha}$ that should iterate two equations specified by (15.11) and (15.12) while satisfying the following Kuhn–Tucker conditions:

$$\begin{aligned} \lambda_i &= 0,\ i \in P \\ \lambda_i &< 0,\ i \in R \end{aligned} \tag{15.13}$$

where P and R represent passive and active sets that contain indices representing negative or positive abundances, respectively. By replacing dot products in (15.11) and (15.12) using kernel tricks (15.6), $\hat{\boldsymbol{\alpha}}^{\text{KNCLS}}(\mathbf{r})$ can be derived by

$$\hat{\boldsymbol{\alpha}}^{\text{KNCLS}}(\mathbf{r}) = (K(\mathbf{M},\mathbf{M}))^{-1}K(\mathbf{M},\mathbf{r}) - (K(\mathbf{M},\mathbf{M}))^{-1}\boldsymbol{\lambda} \tag{15.14}$$

and

$$\boldsymbol{\lambda} = K(\mathbf{M},\mathbf{r}) - K(\mathbf{M},\mathbf{M})\hat{\boldsymbol{\alpha}}^{\text{KNCLS}}(\mathbf{r}) \tag{15.15}$$

Since $(K(\mathbf{M},\mathbf{M}))^{-1}K(\mathbf{M},\mathbf{r})$ can be considered as kernel LSOSP estimator and $\boldsymbol{\alpha}^{\text{KLSOSP}}(\mathbf{r})$ specified by

$$\hat{\boldsymbol{\alpha}}^{\text{KLSOSP}}(\mathbf{r}) = \left(\hat{\alpha}_1^{\text{KLSOSP}}(\mathbf{r}), \hat{\alpha}_2^{\text{KLSOSP}}(\mathbf{r}), \ldots, \hat{\alpha}_p^{\text{KLSOSP}}(\mathbf{r})\right)^T \tag{15.16}$$

KNCLS actually makes use of KLSOSP as its initial guess to iterate two equations (15.14) and (15.15) to find its final KNCLS solution as follows.

KNCLS algorithm:

1. Initialization: set the passive set $P^{(0)} = \{1, 2, \ldots, p\}$ and active set $R^{(0)} = \phi$, that is, empty set. Let $k = 0$.
2. Compute $\hat{\boldsymbol{\alpha}}^{\text{KLSOSP}}$ and let $\hat{\boldsymbol{\alpha}}^{\text{KNCLS}(k)} = \hat{\boldsymbol{\alpha}}^{\text{KLSOSP}}$.
3. At the *j*th iteration. If all components in $\hat{\boldsymbol{\alpha}}^{\text{KNCLS}(k)}$ are positive, the algorithm is terminated. Otherwise, continue.
4. Let $k = k + 1$.
5. Move all indices in $P^{(k-1)}$ that correspond to negative components of $\hat{\boldsymbol{\alpha}}^{\text{KNCLS}(k-1)}$ to $R^{(k-1)}$ and the resulting index sets are denoted by $P^{(k)}$ and $R^{(k)}$, respectively. Create a new index set $S^{(k)}$ and set it equal to $R^{(k)}$.
6. Let $\boldsymbol{\alpha}_{R^{(k)}}$ denote the vector consisting of all components $\hat{\boldsymbol{\alpha}}^{\text{KLSOSP}}$ in $R^{(k)}$.
7. Form a steering matrix $\boldsymbol{\Phi}_{\alpha}^{(k)}$ by deleting all rows and columns in the matrix $(K(\mathbf{M}, \mathbf{M}))^{-1}$ that are specified by $P^{(k)}$.
8. Calculate $\lambda^{(k)} = \left(\Phi_{\alpha}^{(k)}\right)^{-1} \hat{\alpha}_{R^{(k)}}$. If all components in $\lambda^{(k)}$ are negative go to step 15. Otherwise, continue.
9. Find $\lambda_{\max}^{(k)} = \max_j \lambda_j^{(k)}$ and move the index in $R^{(k)}$ that corresponds to $\lambda_{\max}^{(k)}$ to $P^{(k)}$.
10. Calculate a new $\lambda^{(k)}$, as shown in step 8, with the new $R^{(k)}$ and $P^{(k)}$ index sets.
11. Form another matrix $\Psi_{\lambda}^{(k)}$ by deleting every column of $(K(\mathbf{M}, \mathbf{M}))^{-1}$ specified by $P^{(k)}$.
12. Set $\hat{\boldsymbol{\alpha}}_{S^{(k)}} = \boldsymbol{\alpha}^{\text{KLSOSP}} - \Psi_{\lambda}^{(k)} \lambda^{(k)}$.
13. If any components of $\hat{\boldsymbol{\alpha}}_{S^{(k)}}$ in $S^{(k)}$ are negative, then move these components from $P^{(k)}$ to $R^{(k)}$. Go to step 6.
14. Form another matrix $\Psi_{\lambda}^{(k)}$ by deleting every column of $(K(\mathbf{M}, \mathbf{M}))^{-1}$ specified by $P^{(k)}$.
15. Set $\hat{\boldsymbol{\alpha}}^{\text{KNCLS}(k)} = \hat{\boldsymbol{\alpha}}^{\text{KLSOSP}} - \Psi_{\lambda}^{(k)} \lambda^{(k)}$. Go to step 3.

15.2.3 Kernel-Based Fully Constraint Least Square (KFCLS)

According to Heinz and Chang (2001) one simple approach to implementing ASC in conjunction with ANC is to introduce a new signature matrix **N** and an auxiliary vector **s** into the NCLS with

$$\mathbf{N} = \begin{bmatrix} \delta\mathbf{M} \\ \mathbf{1}^T \end{bmatrix} \text{ and } \mathbf{s} = \begin{bmatrix} \delta\mathbf{r} \\ 1 \end{bmatrix} \tag{15.17}$$

where $\mathbf{1} = (1, 1, \ldots, 1)^T$ is a *p*-dimensional vector and δ controls the rate of convergence for FCLS algorithm. By replacing the **M** and **r** in KNCLS with modified signature matrix **N** and the new vector **s** in (15.17), respectively, a KFCLS algorithm can be derived as follows.

KFCLS algorithm:

1. Initialization: Generate the modified signature matrix, **N** and modified pixel vector **s** based on (15.16). Set the passive set $P^{(0)} = \{1, 2, \ldots, p\}$ and active set $R^{(0)} = \phi$, that is, empty set. Let $k = 0$.
2. Compute $\hat{\boldsymbol{\alpha}}^{\text{KNCLS}}$ and let $\hat{\boldsymbol{\alpha}}^{\text{KFCLS}(k)} = \hat{\boldsymbol{\alpha}}^{\text{KNCLS}}$.
3. At the *j*th iteration. If all components in $\hat{\boldsymbol{\alpha}}^{\text{KFCLS}(k)}$ are positive, the algorithm is terminated. Otherwise, continue.

4. Let $k = k + 1$.
5. Move all indices in $P^{(k-1)}$ that correspond to negative components of $\hat{\boldsymbol{\alpha}}^{\mathrm{KNCLS}(k-1)}$ to $R^{(k-1)}$ and the resulting index sets are denoted by $P^{(k)}$ and $R^{(k)}$, respectively. Create a new index set $S^{(k)}$ and set it equal to $R^{(k)}$.
6. Let $\boldsymbol{\alpha}_{R^{(k)}}$ denote the vector consisting of all components $\hat{\boldsymbol{\alpha}}^{\mathrm{KNCLS}}$ in $R^{(k)}$.
7. Form a steering matrix $\boldsymbol{\Phi}_{\alpha}^{(k)}$ by deleting all rows and columns in the matrix $(K(\mathbf{N},\mathbf{N}))^{-1}$ that are specified by $P^{(k)}$.
8. Calculate $\lambda^{(k)} = \left(\Phi_{\alpha}^{(k)}\right)^{-1}\hat{\alpha}_{R^{(k)}}$. If all components in $\lambda^{(k)}$are negative go to step 15. Otherwise, continue.
9. Find $\lambda_{\max}^{(k)} = \max_j \lambda_j^{(k)}$ and move the index in $R^{(k)}$ that corresponds to $\lambda_{\max}^{(k)}$ to $P^{(k)}$.
10. Calculate a new $\lambda^{(k)}$, as shown in step 8, with the new $R^{(k)}$ and $P^{(k)}$ index sets.
11. Form another matrix $\Psi_{\lambda}^{(k)}$ by deleting every column of $(K(\mathbf{N},\mathbf{N}))^{-1}$ specified by $P^{(k)}$.
12. Set $\hat{\boldsymbol{\alpha}}_{S^{(k)}} = \hat{\boldsymbol{\alpha}}^{\mathrm{KNCLS}} - \Psi_{\lambda}^{(k)}\lambda^{(k)}$.
13. If any components of $\hat{\boldsymbol{\alpha}}_{S^{(k)}}$ in $S^{(k)}$ are negative, then move these components from $P^{(k)}$ to $R^{(k)}$. Go to step 6.
14. Form another matrix $\Psi_{\lambda}^{(k)}$ by deleting every column of $(K(\mathbf{N},\mathbf{N}))^{-1}$ specified by $P^{(k)}$.
15. Set $\hat{\boldsymbol{\alpha}}^{\mathrm{KFCLS}(k)} = \hat{\boldsymbol{\alpha}}^{\mathrm{KNCLS}} - \Psi_{\lambda}^{(k)}\lambda^{(k)}$. Go to step 3.

As a concluding note, the KNCLS and KFCLS presented above are also independently derived in Broadwater et al. (2007) and later published in Camps-Valls and Bruzzone (2009) as a book chapter where the KNCLS was only used as a means of deriving the KFCLS and was not used in their experiments at all. In addition, many details presented here are not available in these two references.

15.2.4 A Note on Kernelization

The key idea of kernelization is the use of the kernel trick that has not been really clarified in many reported efforts when it is used. As a consequence, there is always confusion. As an example, support vector machine (SVM) is originally developed as a binary classifier to find a hyperplane maximizing the distance between support vectors in two separate classes and has nothing to do with kernelization. The introduction of kernelization into SVM allows SVM to make linear decisions in a feature space to resolve linear nonseparability problems. As a matter of fact, SVM used in the literature is indeed kernel-based SVM (KSVM) not SVM without using kernel. In general, there two ways to use the kernel trick to perform kernelization. One is to kernelize a transformation to map all data samples in a high-dimensional feature space. Examples of this type include kernel-based principal components analysis (PCA) and kernel-based RX detector, both of which require all data samples for kernelization of the sample covariance matrix. The advantage of this approach is no prior knowledge is needed since all data sample are involved for kernelization. But its major disadvantage is computational complexity which is exceedingly expensive. Unfortunately, most of the reported research efforts along this line have avoided addressing this issue. To mitigate this dilemma, an alternative approach is to use a specifically selected of data samples to perform kernelization such as kernel-based Fisher's linear discriminant analysis and KSVM. Consequently, the computational complexity is reduced to that required to perform kernelization only on these selected samples. However, this also comes with a price of how to judiciously select data samples for kernelization. This is a very challenging issue. A third approach is the approach proposed in this chapter which kernelizes a classifier instead of training samples. More specifically, KLSMA only kernelizes the signatures used by OSP, LSOSP, NCLS, and FCLS that are used to perform spectral unmixing where these signatures are provided *a priori* not necessarily training samples. This is quite different from existing kernel-based techniques as described above which operate entire data samples or training samples to be mapped into a high dimensional feature space. More importantly, the computational cost saving is tremendous because the number of the signatures kernelized by KLSMA is generally very small compared to other kernel-based methods which requires a large number of training samples such as SVM or Fisher's linear discriminant analysis or the entire data samples such as PCA.

15.3 Synthetic Image Experiments

The synthetic image has the five mineral spectral signatures, A, B, C, K, M, marked by circles in Figure 1.12(b) are used to simulate 25 panels shown in Figure 15.1 with five panels in each row simulated by the same mineral signature and five panels in each column having the same size. Among 25 panels are five 4 × 4 pure-pixel panels for each row in the first column and five 2 × 2 pure-pixel panels for each row in the second column, the five 2 × 2-mixed pixel panels for each row in the third column and both the five 1 × 1 subpixel panels for each row in the fourth column and the fifth column where the mixed and subpanel pixels are simulated according to legends in Figure 15.1. So, a total of 100 pure pixels (80 in the first column and 20 in second column), referred to as endmember pixels are simulated in the data by the five endmembers, A, B, C, K, M.

These 25 panels are then inserted in a synthetic image with size of 200 × 200 pixels in two ways. One is the background pixels removed to accommodate the inserted target pixels that result in target implantation (TI). The other is the inserted target panels directly superimposed over the background pixels that result in target embededness (TE). In both cases the background is simulated by the sample mean of the real image scene in Figure 1.12(a). Depending upon how a Gaussian noise is added to the TI and TE, three scenarios for each case are also simulated with details descried in Section 4.3. The goal of synthetic image experiments presented in this section is to study the role that kernelization plays in LSMA for hyperspectral imagery as well as its impact on unmixing performance. The six scenarios described in Chapter 4, TI1, TE2, TI3 and TE1, TE2, TE3 are simulated by the information provided in Figure 15.1 for performance evaluation.

Since all six scenarios produced similar results only the results for TE3 is selected as a representative example for illustration. Figure 15.2 shows the unmixed results of 130 panel pixels in TE

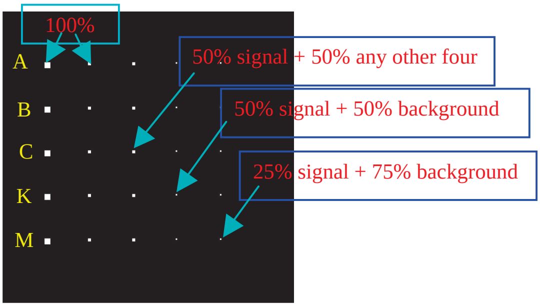
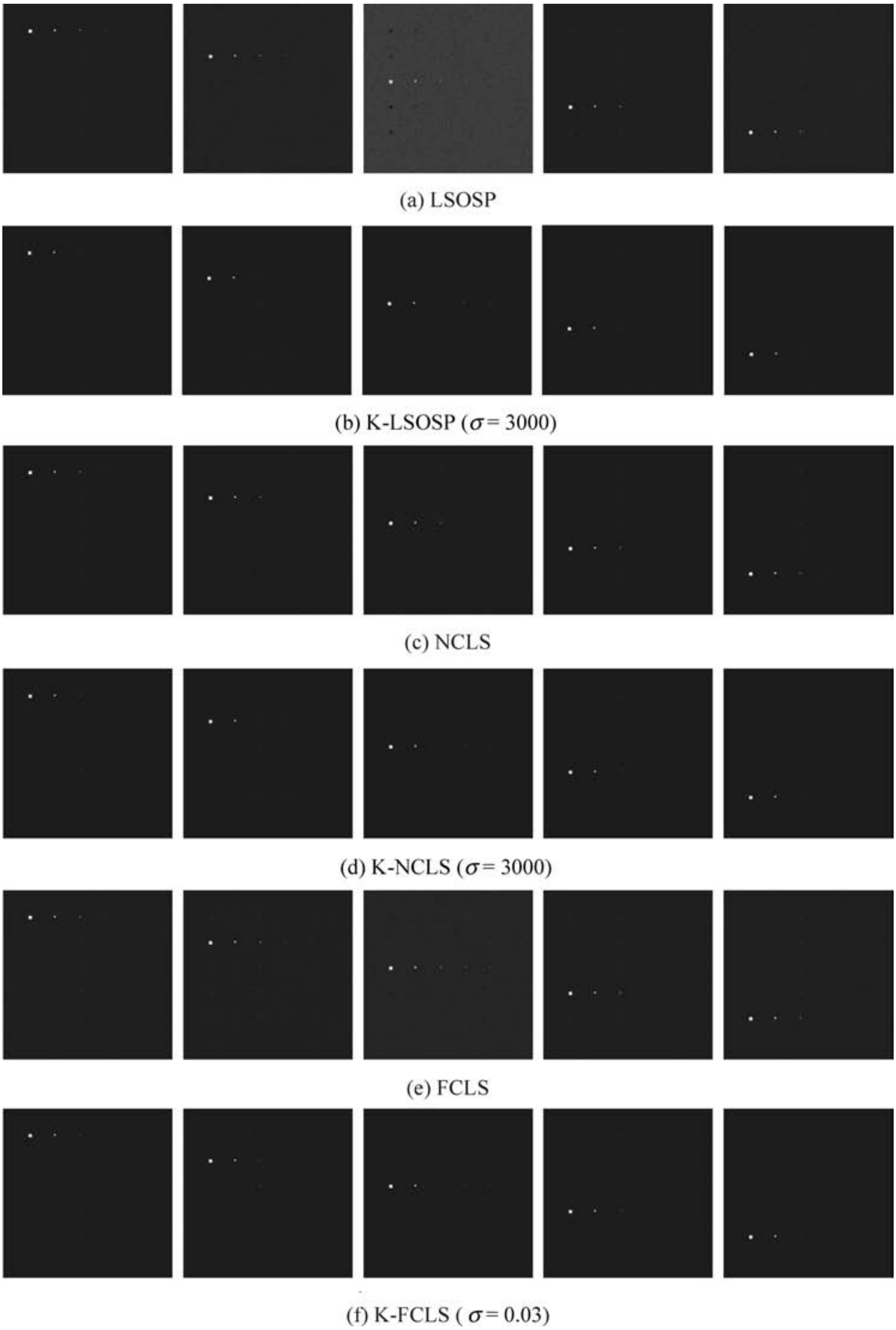

100%
A
B
C
K
M
50% signal + 50% any other four
50% signal + 50% background
25% signal + 75% background

Figure 15.1 A set of 25 panels simulated by A, B, C, K, M.



(a) LSOSP

(b) K-LSOSP (σ = 3000)

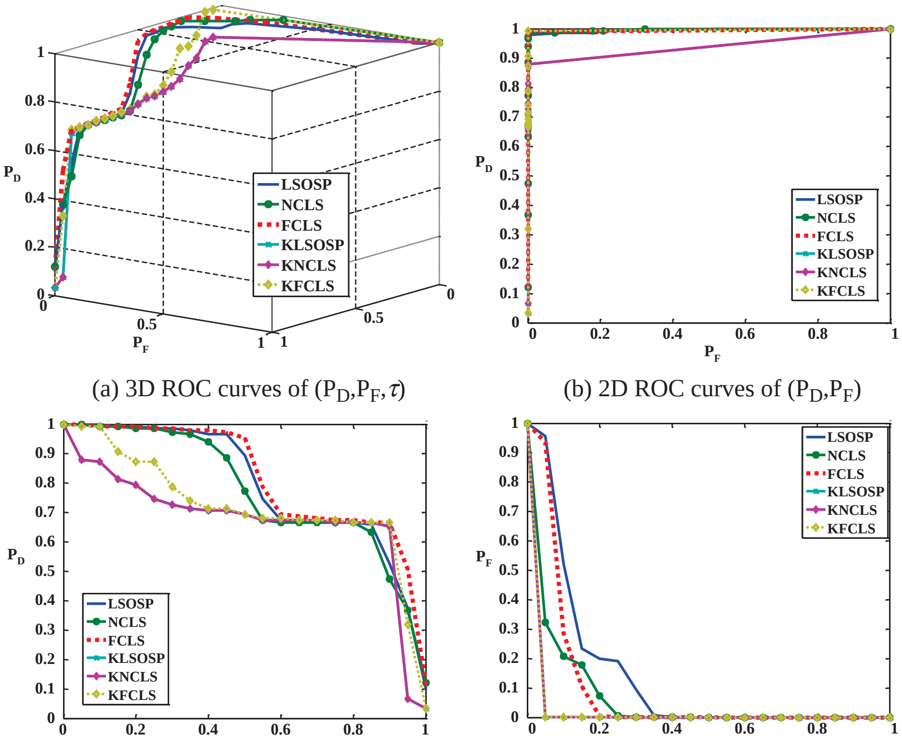
(c) NCLS

(d) K-NCLS (σ = 3000)

(e) FCLS

(f) K-FCLS (σ = 0.03)

Figure 15.2 LSMA and K-LSMA resulting image of synthetic linear mixture experiments.



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

(d) 2D ROC curves of (P_F,τ)

Figure 15.3 3D ROC analysis of TE3 scenario, (a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of (P_F,τ).

3 by LSOSP, NCLS, FCLS, and their kernel counterparts with RBF kernels used to perform kernelization where a Gaussian noise with SNR 20:1 is added to the target panel pixels which are directly superimposed over the background pixels.

Through visual inspection of the results in Figure 15.1 only KLSOSP could slightly improve the detection of the panel pixels in the third row compared to its counterpart without using kernelization. Other than that it seemed that LSMA was not really benefited from kernelization in the sense of unmixing. To further support this conclusion, a 3D ROC analysis developed in Chapter 3 was performed on the unmixed results in Figures 15.2 and 15.3 show a 3D ROC curves of (P_D,P_F,τ), 2D ROC curves of (P_D,P_F), 2D ROC curves of (P_D,τ), and 2D ROC curves of (P_F,τ) where it was surprising to observe that the best results were not those produced by using kernelization.

To see the whole picture of performance analysis conducted for six scenarios, Tables 15.1–15.3 tabulate the averaged detection rates of 130 panel pixels over the six scenarios resulting from 3D ROC analysis where the results indeed confirmed that the best results were those produced by LSMA shown in Tables 15.1 and 15.2 in terms of areas under the 2D ROC curves of (P_D,P_F) and 2D ROC curves of (P_D,τ). According to the areas of 2D ROC curves of (P_F,τ) in Table 15.3 kernelization did help LSMA reduce false alarm rates.

The above synthetic image experiments demonstrate an important fact that kernelization did not necessarily improve LSMA performance as we expected when the target pixels of interest were

Table 15.1 Areas under 2D ROC curves of P_D versus P_F for TI and TE

	TI 1	TI 2	TI 3	TE 1	TE 2	TE3
LSOSP	**0.9967**	0.9978	0.9978	**0.9967**	**0.9977**	0.9958
NCLS	**0.9967**	0.9972	0.9972	**0.9967**	0.9968	**0.9974**
FCLS	**0.9967**	**0.9980**	**0.9980**	0.9960	0.9963	0.9959
K-LSOSP	0.9307	0.9261	0.9261	0.9400	0.9400	0.9400
K-NCLS	0.9307	0.9261	0.9261	0.9400	0.9400	0.9400
K-FCLS	0.9900	0.9856	0.9856	0.9966	0.9966	0.9966

Table 15.2 Areas under 2D ROC curves of (P_D vs. τ) for TI and TE

	TI 1	TI 2	TI 3	TE 1	TE 2	TE3
LSOSP	**0.8300**	0.8400	0.8400	0.8300	0.8423	0.7982
NCLS	**0.8300**	0.8323	0.8323	0.8300	0.8350	0.7770
FCLS	**0.8300**	**0.8427**	**0.8427**	**0.8407**	**0.8517**	**0.8215**
K-LSOSP	0.7363	0.7350	0.7350	0.7343	0.7343	0.6792
K-NCLS	0.7363	0.7350	0.7350	0.7343	0.7343	0.6792
K-FCLS	0.7477	0.7483	0.7483	0.7620	0.7620	0.7252

Table 15.3 Areas under 2D ROC curves of (P_F vs. τ) for TI and TE

Linear synthetic mixture						
	TI 1	TI 2	TI 3	TE 1	TE 2	TE3
LSOSP	**0.0250**	0.1124	0.1124	**0.0250**	0.1133	0.1353
NCLS	**0.0250**	0.0709	0.0709	**0.0250**	0.0760	0.0644
FCLS	**0.0250**	0.1090	0.1090	0.0350	0.0805	0.0915
K-LSOSP	0.0550	0.0586	0.0586	**0.0250**	**0.0250**	**0.0250**
K-NCLS	0.0550	0.0586	0.0586	**0.0250**	**0.0250**	**0.0250**
K-FCLS	**0.0250**	**0.0305**	**0.0305**	0.0251	0.0251	0.0251

only lightly mixed in which case FCLS was always preferred to any other LSMA techniques regardless whether or not they were kernelized. This conclusion is further supported by the following experiments using two real data sets, AVIRIS Purdue data where data sample vectors are generally heavily mixed and HYDICE data where data sample vectors are less mixed.

15.4 AVIRIS Data Experiments

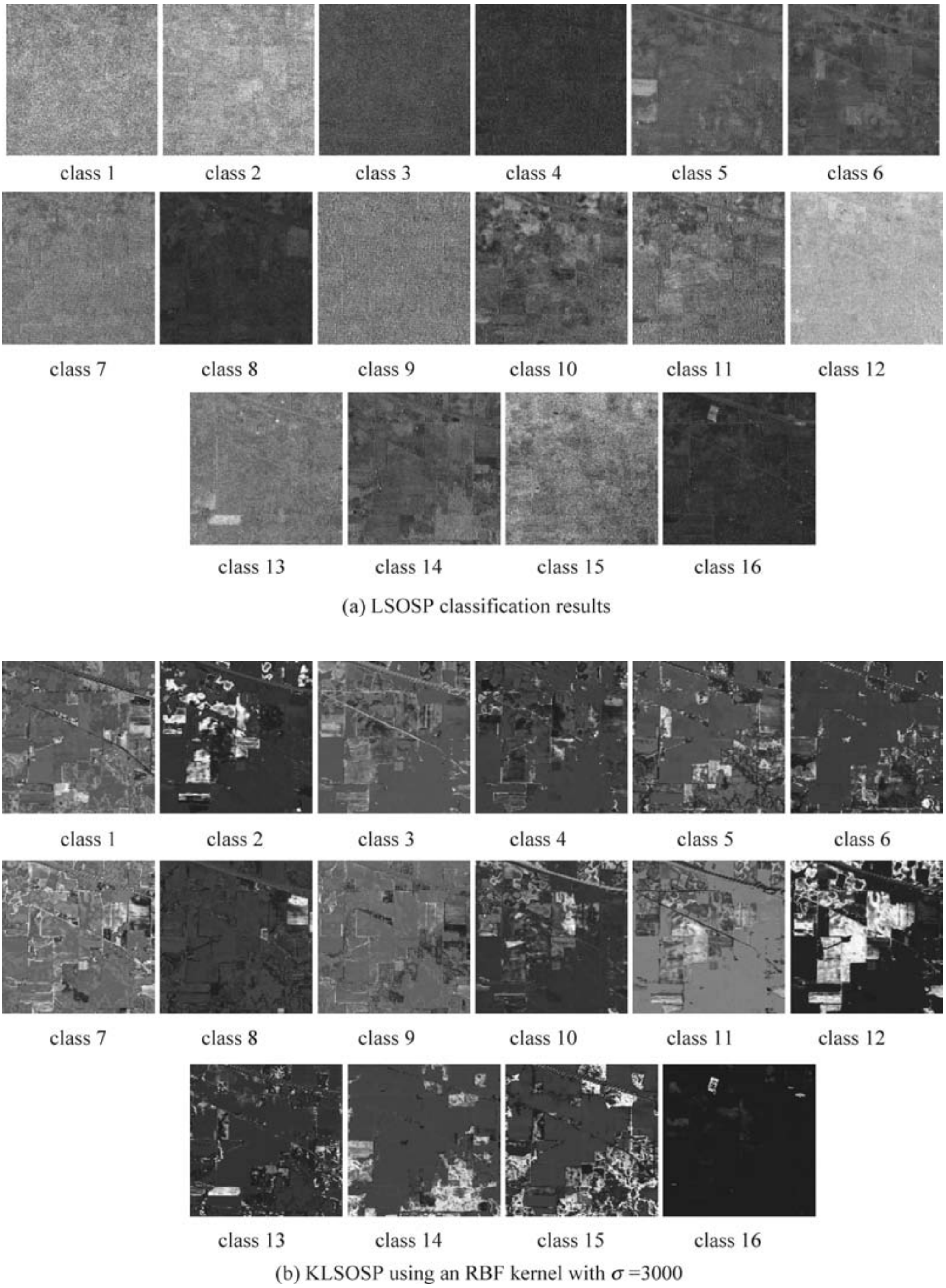
To demonstrate how much benefit can be gained from using kernel-based LSMA in hyperspectral classification, two real hyperspectral data sets were described in Chapter 1, Purdue Indiana Indian Pine test site in Figure 1.13 and HYDICE data scene in Figure 1.15 will be used for experiments for this purpose. In this section, we first study the AVIRIS data of Purdue Indiana

Indian Pine test site in Figure 1.15. According to the provided ground truth there are 17 classes in this image scene shown in Figure 1.13(c) including the background labeled by class 17 that includes a wide variety of targets such as highways, railroad, houses/buildings, and vegetation that may not be of interest in agriculture applications. The spatial locations of all the 17 classes are shown in Figure 1.13(d) where the number in a parenthesis after a class label in Figure 1.13(d) is the total number of data samples in that particular class. The total number of data samples in the scene is $145 \times 145 = 21025$. Two sets of experiments were conducted based on this scene to evaluate the KLSMA performance. One was mixed pixel classification to show the superior performance of KLSMA to that of LSMA without using kernels where data sample vectors are heavily mixed. The other was to use a three-dimensional receiver operating characteristics (3D ROC) analysis developed in Chapter 3 to conduct a quantitative analysis of classification performance of 16 classes to evaluate the selection of three kernels and their used parameters.

The following experiments were conducted by randomly selecting 10% of data samples of each of 16 classes as training samples to produce the class sample means that are further used as endmembers, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{16}$ in model (15.1). Figure 15.4(a) and (b) shows comparative 16-class classification results produced by LSOSP and KLSOSP using the RBF kernel with $\sigma = 3000$ where the KLSOSP significantly improved the LSOSP in classification by visual inspection. Those classes missed in Figure 15.4(a) were clearly classified in Figure 15.4(b). It should be noted that the values of the parameter σ were empirically selected to only illustrate the benefit of using kernels by LSMA. Their values were not optimal. The issue in selection of kernels with various parameters will be discussed later. Nevertheless, finding optimal value for σ is an optimization issue which is very challenging and beyond the scope of this chapter.

Similar experiments were also conducted to compare the relative performances of NCLS and FCLS to their kernel-based counterparts. The results are shown in Figures 15.5 and 15.6, respectively, where the results using RBF kernels in Figures 15.5(b) and 15.6(b) showed improvement over their counterparts without using kernels in Figures 15.5(a) and 15.6(a). Specifically, a great improvement was witnessed in Figure 15.6(b) compared to that in Figure 15.6(a). However, it was interesting to note that there was appreciable visual difference between the NCLS and KNCLS by comparing results in Figure 15.5(a) and (b). This implied that NCLS had better classification ability without using kernels.

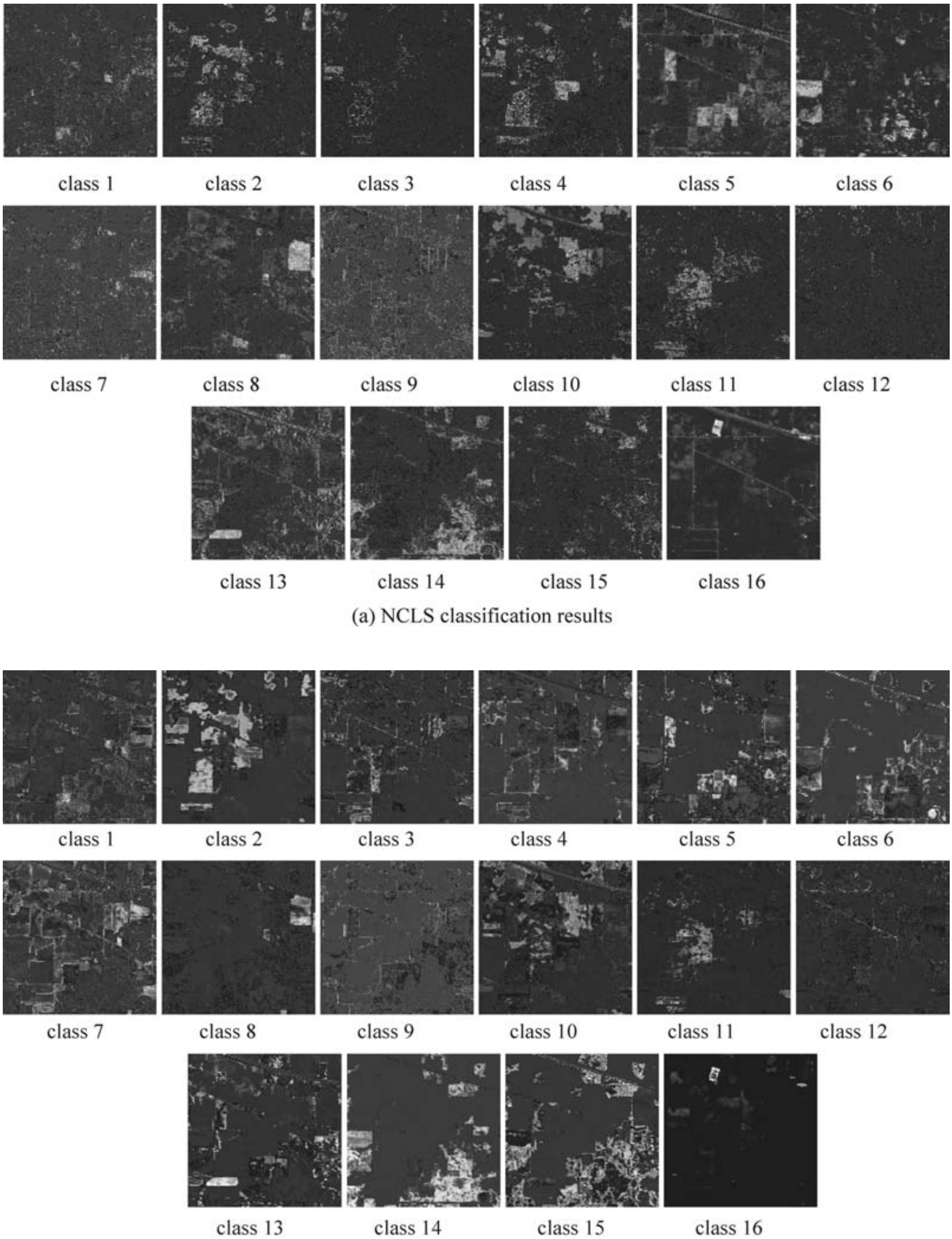
The results of Figures 15.4–15.6 showed visual classification to provide qualitative analysis. To further conduct quantification analysis the ground truth of 16 classes provided in Figure 1.13(d) was used for this purpose. Since LSMA is basically designed to unmix the abundance fraction of each of endmembers that formed a linear mixing model (15.1) it is a soft decision-made classifier which must convert an unmixed real-valued abundance fraction to a binary decision to perform hard-decision classification. To resolve this dilemma, the 3D ROC analysis once again was used for performance evaluation. More specifically, a 3D ROC curve is a curve of three parameters, detection power, $\mathrm{P_D}$, false alarm probability, $\mathrm{P_F}$ and a threshold parameter τ where extends the commonly used 2D ROC curve of $\mathrm{P_D}$ and $\mathrm{P_F}$ by including the parameter τ as a third dimension. By virtue of the parameter τ the unmixed abundance fractions were thresholded by the τ to binary values so as to achieve detection. As a result, a soft decision-made LSMA classifier can produce a 3D ROC curve of (P_D, P_F, τ) from which three 2D ROC curves of (P_D, P_F), (P_D, τ) and (P_F, τ) can be also plotted for detection. By virtue of such a 3D ROC analysis we can actually evaluate the classification performance of KLSMA versus LSMA according to three types of commonly used kernels in the following sections.



(a) LSOSP classification results

(b) KLSOSP using an RBF kernel with σ =3000

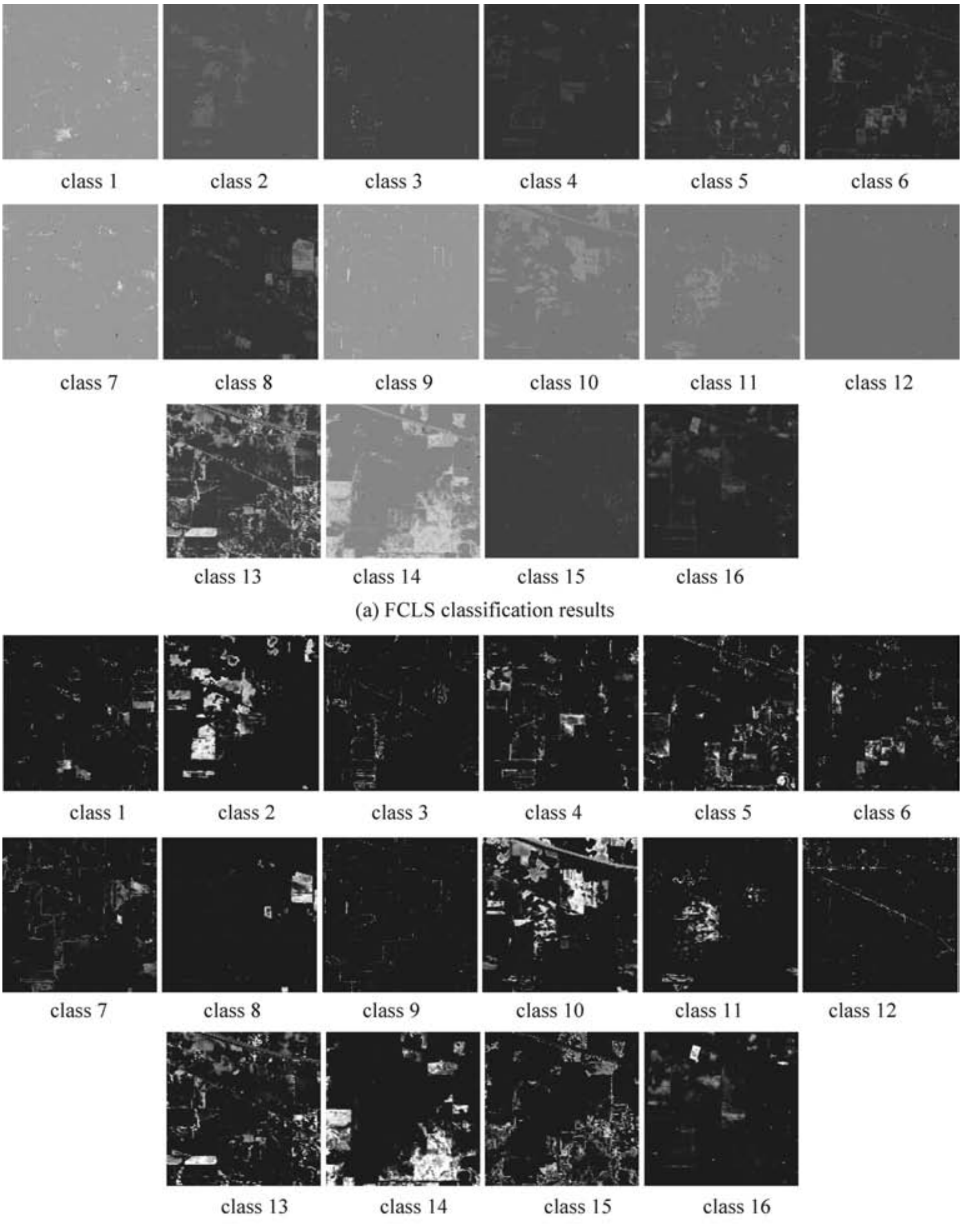
Figure 15.4 Comparative results produced by LSOSP and KLSOSP classification results using RBF kernel.



(a) NCLS classification results

(b) KNCLS using an RBF kernel with $\sigma = 3000$

Figure 15.5 Comparative results produced by NCLS and KNCLS classification results using RBF kernel.



(a) FCLS classification results

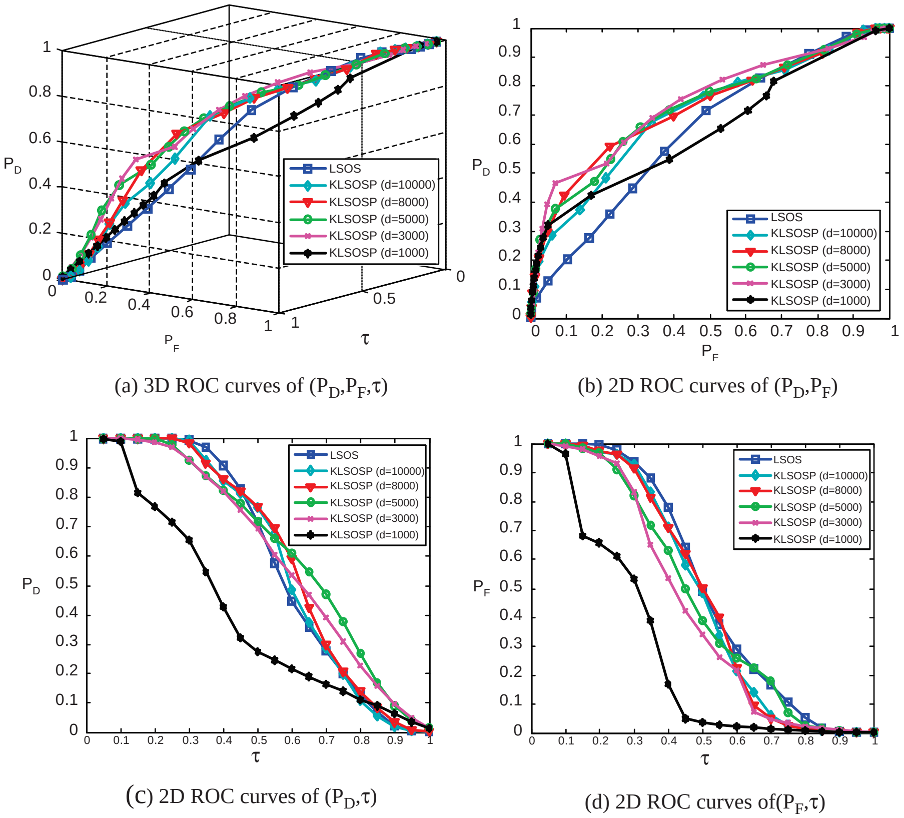
(b) KFCLS using an RBF kernel with $\sigma=0.1$

Figure 15.6 Comparative results produced by FCLS and KFCLS classification results using RBF kernel.

15.4.1 Radial Basis Function Kernels

The RBF kernels are mostly widely used in kernel-based approaches since this type of kernels have been shown to be more effective than other types of kernels in the literature. This is also the case in our experiments. The kernels are parameterized by the width σ of Gaussian kernels. Figures 15.7–15.9 show 3D ROC curves in (a) along with their corresponding 2D ROC curves in (b–d) of 3 LSMA classifiers, LSOSP, NCLS, and FCLS along with their kernel counterparts, KLSOSP, KNCLS, and KFCLS, respectively. The five values of the parameter σ were empirically chosen to demonstrate their relative performance. Tables 15.4–15.6 also calculate the area under each of 2D ROC curves, A_z for quantitative analysis where the shade rows indicate the best values for the σ according to the 2D ROC curves of (P_D,P_F) for each classifier versus its kernel counterpart.

As shown in Figures 15.7–15.9 and Tables 15.4–15.6, it was clear that in order for a kernel-based LSMA classifier to perform better than its counterpart without kernels the selection of the value for the parameter σ was crucial. However, the experimental results also showed that the effectiveness of using kernels also depended upon the classifier to be used for mixed pixel



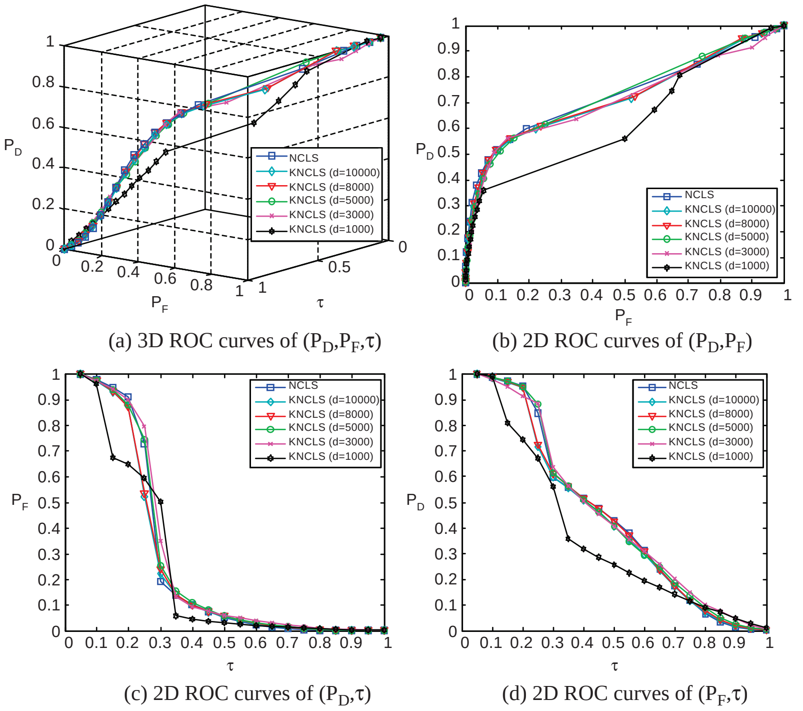
(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

(d) 2D ROC curves of(P_F,τ)

Figure 15.7 LSOSP/KLSOSP curves obtained by 16 classes average classification rate using RBF kernel with different σ(a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of $(\mathrm{P_D},\tau)$; (d) 2D ROC curves of (P_F,τ).



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

(d) 2D ROC curves of (P_F,τ)

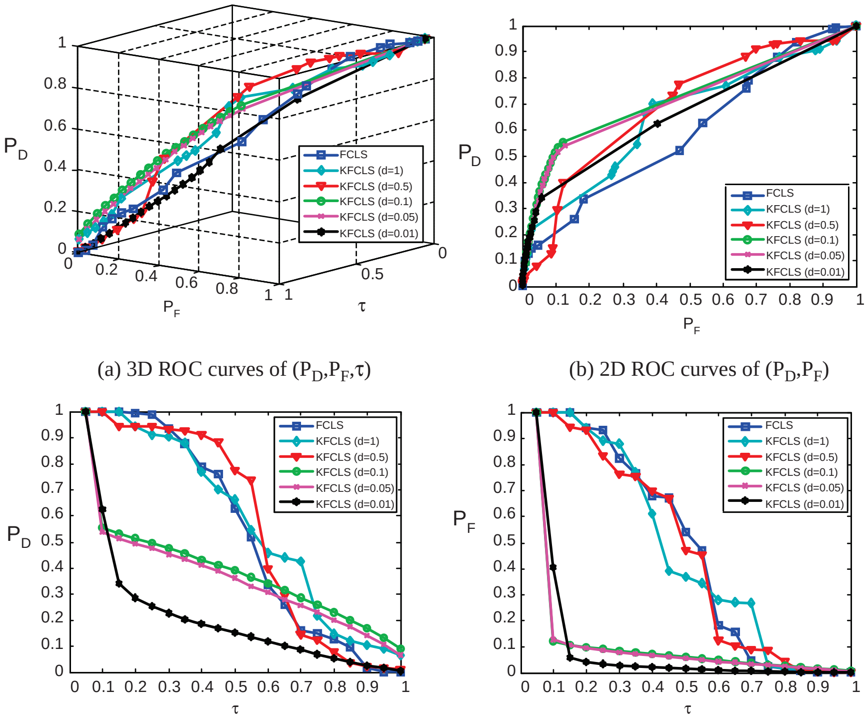
Figure 15.8 NCLS/KNCLS curves obtained by 16 classes average classification rate using RBF kernel with different σ(a) 3D ROC curves of (P_D,P_F,τ);(b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of (P_F,τ).

Table 15.4 Values of A_z under three 2D ROC curves in Figure 15.7

	P_D versus P_F	P_D versus τ	P_F versus τ
LSOSP	0.6406	0.5498	0.4714
KLSOSP ($\sigma = 10000$)	0.7030	0.5524	0.4376
KLSOSP ($\sigma = 8000$)	0.7209	0.5658	0.4402
KLSOSP ($\sigma = 5000$)	0.7228	0.5916	0.4254
KLSOSP ($\sigma = 3000$)	0.7412	0.5675	0.3909
KLSOSP ($\sigma = 1000$)	0.6465	0.3626	0.2346

Table 15.5 Values of A_z under three 2D ROC curves in Figure 15.8

	P_D versus P_F	P_D versus τ	P_F versus τ
NCLS	0.7285	0.4083	0.2354
KNCLS ($\sigma = 10000$)	0.7161	0.3973	0.2245
KNCLS ($\sigma = 8000$)	0.7198	0.4036	0.2267
KNCLS ($\sigma = 5000$)	0.7296	0.4115	0.2409
KNCLS ($\sigma = 3000$)	0.7091	0.4152	0.2495
KNCLS ($\sigma = 1000$)	0.6286	0.3286	0.2073



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

(d) 2D ROC curves of (P_F,τ)

Figure 15.9 FCLS/KFCLS curves obtained by 16 classes average classification rate using RBF kernel with different σ(a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of (P_F,τ).

classification. For example, worst cases for KLSOSP and KFCLS were those with $\sigma = 1000$ and 1, respectively, where LSOSP was nearly as worst as KLSOSP with $\sigma = 1000$ and KFCLS was the worst classifier. However, the above phenomena were completely reversed for NCLS where NCLS without using kernels could perform as well as the best KNCLS did with $\sigma = 5000$. These experiments demonstrated that the only LSMA classifier could compete against kernel-based LSMA classifier was NCLS. The results also suggested that using appropriate RBF kernels could indeed improve mixed pixel classification, when pixels are heavily mixed.

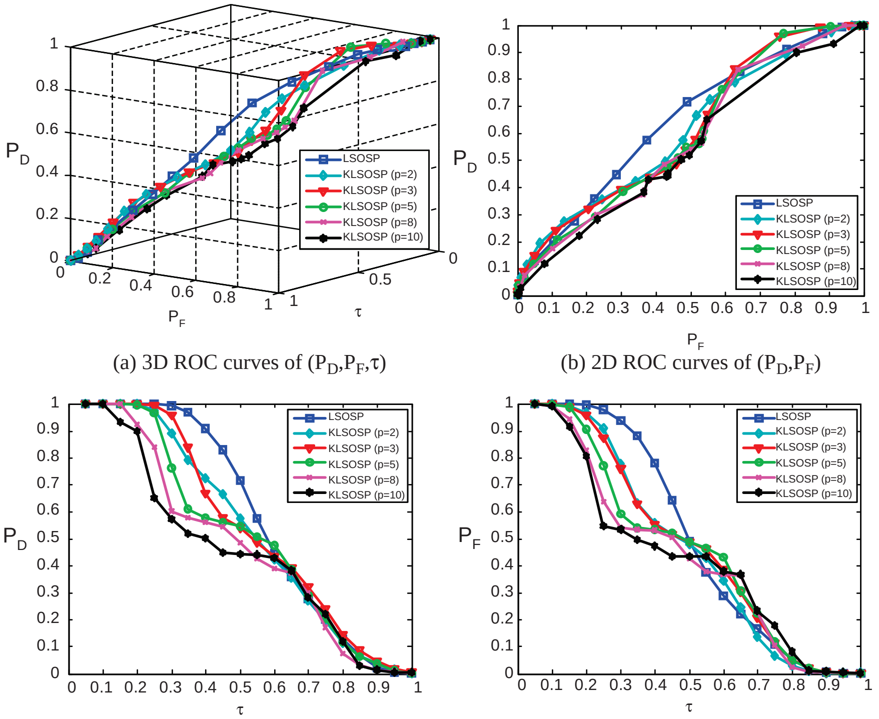
Table 15.6 Values of A_z under three 2D ROC curves in Figure 15.9

	P_D versus P_F	P_D versus τ	P_F versus τ
FCLS	0.5933	0.5064	0.4376
KFCLS ($\sigma = 1$)	0.6526	0.5425	0.4295
KFCLS ($\sigma = 0.5$)	0.6919	0.5303	0.4231
KFCLS ($\sigma = 0.1$)	0.7258	0.3545	0.0783
KFCLS ($\sigma = 0.05$)	0.7163	0.3310	0.0758
KFCLS ($\sigma = 0.01$)	0.6636	0.1792	0.0597

15.4.2 Polynomial Kernels

The parameter used by polynomial kernels was the degree, p. Figures 15.10–15.12 show 3D ROC curves in (a) along with their corresponding 2D ROC curves in (b–d) of three LSMA classifiers, LSOSP, NCLS, and FCLS along with their kernel counterparts, KLSOSP, KNCLS, and KFCLS, respectively. The five values of the parameter p were empirically chosen, $p=2$, 3, 5, 8, 10 for KLSOSP, KNCLS and 10, 20, 30, 40 50 for KFCLS to demonstrate their relative performance. Tables 15.7–15.9 also calculate the area under each of 2D ROC curves, A_z for quantitative analysis where the shade rows indicate the best values for the p according to ROC curves of (P_D,P_F) for each classifier versus its kernel counterpart.

As shown in Figures 15.10–15.12 and Tables 15.7–15.9, it was clear that the KLSMA using polynomial kernels did not perform as well as KLSMA using RBF kernels except the case of NCLS that actually showed the benefit of using kernels by KNCLS compared to their corresponding KNCLS using RBF kernels. However, this was the only case that NCLS using polynomial kernels did better than those using RBF kernels. We believe that such improvement resulted from NCLS itself not from the kernels it used. The same phenomenon could be also seen and demonstrated by the case of using sigmoid kernels in the following section. Also, from Figure 15.10 and Table 15.7 LSOSP showed better performance than KLSOSP regardless of what degrees were used.



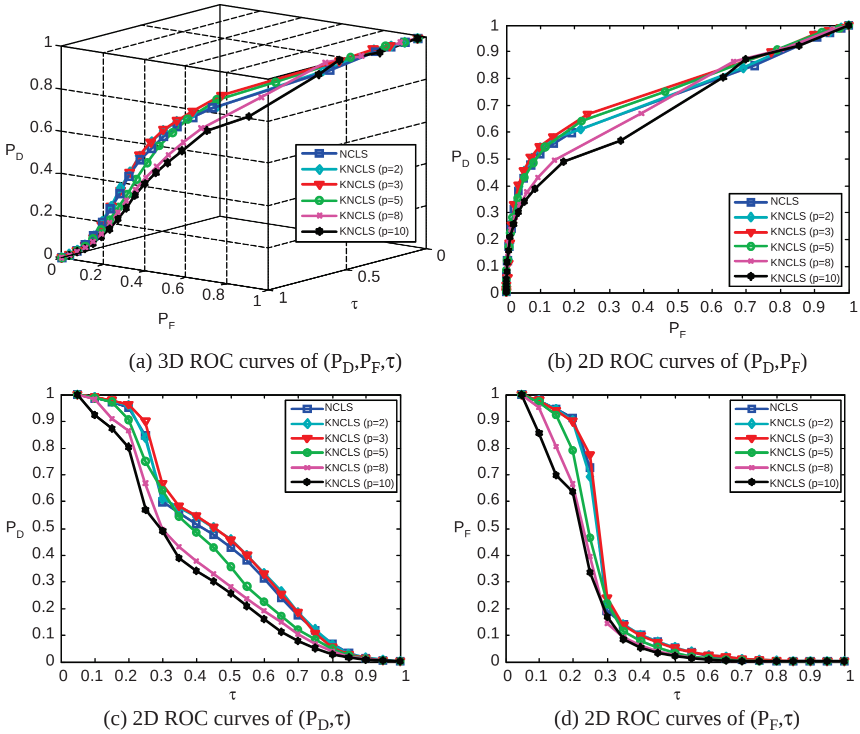
(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c)2D ROC curves of (P_D,τ)

(d) 2D ROC curves of (P_F,τ)

Figure 15.10 LSOSP/KLSOSP curves obtained by 16 classes average classification rate using polynomial kernel with different p (a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of (P_F,τ).



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

(d) 2D ROC curves of (P_F,τ)

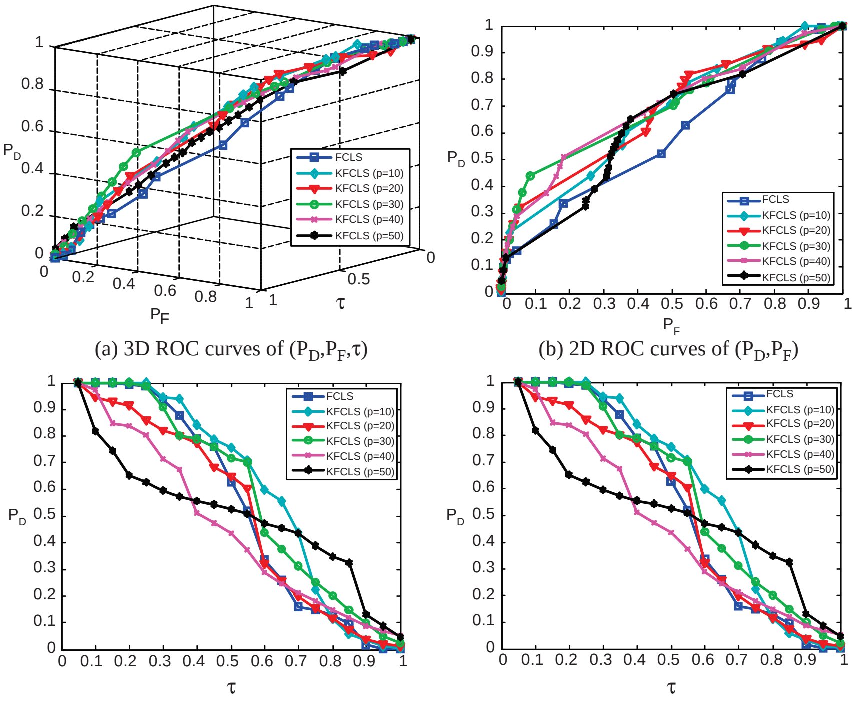
Figure 15.11 NCLS/KNCLS curves obtained by 16 classes average classification rate using Polynomial kernel with different p (a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of (P_F,τ).

Table 15.7 Values of A_z under three 2D ROC curves in Figure 15.10

	P_D versus P_F	P_D versus τ	P_F versus τ
LSOSP	0.6406	0.5498	0.4714
KLSOSP ($p = 2$)	0.6069	0.5042	0.4286
KLSOSP ($p = 3$)	0.6110	0.5116	0.4396
KLSOSP ($p = 5$)	0.5981	0.4806	0.4221
KLSOSP ($p = 8$)	0.5790	0.4401	0.3955
KLSOSP ($p = 10$)	0.5511	0.4193	0.3910

Table 15.8 Values of A_z under three 2D ROC curves in Figure 15.11

	P_D versus P_F	P_D versus τ	P_F versus τ
NCLS	0.7285	0.4083	0.2354
KNCLS ($p = 2$)	0.7364	0.4186	0.2352
KNCLS ($p = 3$)	0.7584	0.4229	0.2396
KNCLS ($p = 5$)	0.7489	0.3770	0.2099
KNCLS ($p = 8$)	0.7176	0.3326	0.1855
KNCLS ($p = 10$)	0.6884	0.3053	0.1709



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

(d) 2D ROC curves of (P_F,τ)

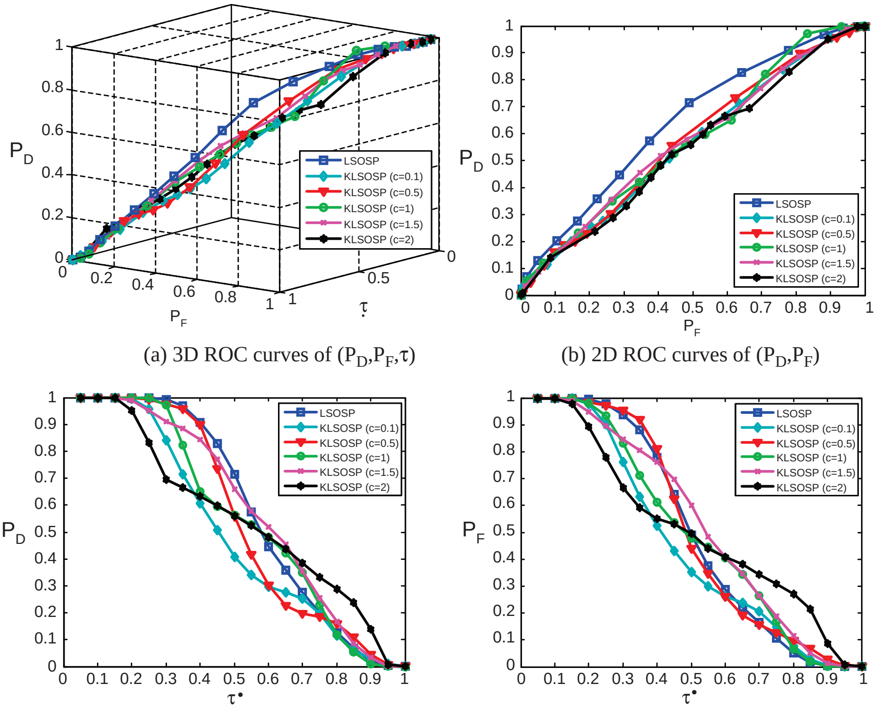
Figure 15.12 FCLS/KFCLS curves obtained by 16 classes average classification rate using polynomial kernel with different p (a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of (P_F,τ).

15.4.3 Sigmoid Kernels

The β_0 and β_1 of the sigmoid kernels used for experiments were set to $\beta_0 = 1$ and $\beta_1 = c/\max(\mathbf{x}^T\mathbf{y})$ with the parameter c where **x** and **y** are taken over all the pixel vectors. Figures 15.13–15.15 show 3D ROC curves in (a) along with their corresponding 2D ROC curves in (b–d) of 3 LSMA classifiers, LSOSP, NCLS, and FCLS along with their kernel counterparts, KLSOSP, KNCLS, and KFCLS, respectively. The five values of the parameter c were empirically chosen as

Table 15.9 Values of A_z under three 2D ROC curves in Figure 15.12

	P_D versus P_F	P_D versus τ	P_F versus τ
FCLS	0.5933	0.5064	0.4376
KFCLS ($p = 10$)	0.6702	0.5749	0.4625
KFCLS ($p = 20$)	0.6773	0.4825	0.3429
KFCLS ($p = 30$)	0.6963	0.5519	0.4130
KFCLS ($p = 40$)	0.6948	0.4255	0.2556
KFCLS ($p = 50$)	0.6288	0.4643	0.3103



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

(d) 2D ROC curves of (P_F,τ)

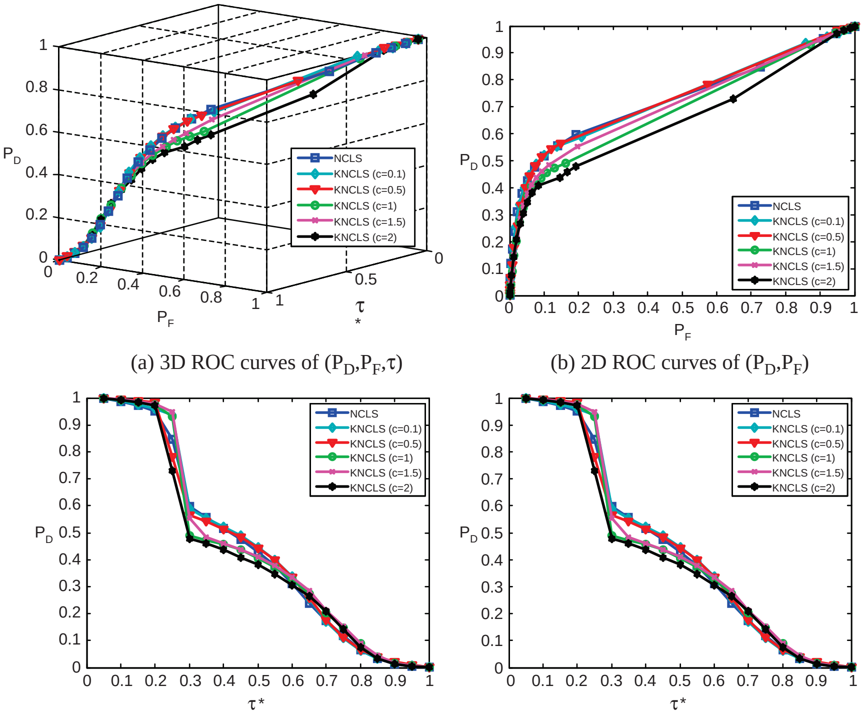
Figure 15.13 LSOSP/KLSOSP curves obtained by 16 classes average classification rate using sigmoid kernel with different σ (a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of (P_F,τ).

indicated in Tables 15.10–15.12 to demonstrate their relative performance where Tables 15.10–15.12 calculate the area under each of 2D ROC curves, A_z, for quantitative analysis and the shade rows indicate the best values for the c according to the 2D ROC curves of (P_D,P_F) for each classifier versus its kernel counterpart.

According to Figures 15.13–15.15 and Tables 15.10–15.12, the kernel-based LSMA classifier using sigmoid kernels performed much worse than KLSMA using RBF kernels and also worse than KLSMA using polynomial kernels. In analogy with KLSMA using polynomial

Table 15.10 Values of A_z under three 2D ROC curves in Figure 15.13

	P_D versus P_F	P_D versus τ	P_F versus τ
LSOSP	0.6406	0.5498	0.4714
KLSOSP ($c = 0.1$)	0.5581	0.4552	0.4174
KLSOSP ($c = 0.5$)	0.5719	0.5135	0.4740
KLSOSP ($c = 1$)	0.5754	0.5146	0.4649
KLSOSP ($c = 1.5$)	0.5707	0.5481	0.4958
KLSOSP ($c = 2$)	0.5457	0.5131	0.4720



(a) 3D ROC curves of (P_D,P_F,τ) (b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ) (d) 2D ROC curves of (P_F,τ)

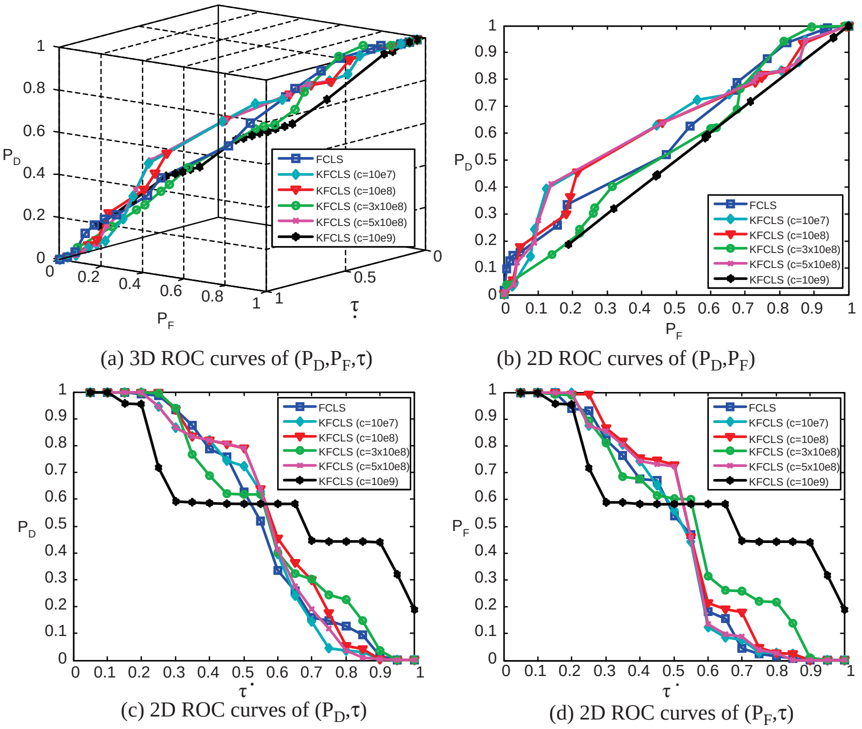
Figure 15.14 NCLS/KNCLS curves obtained by 16 classes average classification rate using sigmoid kernel with different σ (a) 3D ROC curves of (P_D,P_F,τ) (b) 2D ROC curves of (P_D,P_F) (c) 2D ROC curves of (P_D,τ) (d) 2D ROC curves of (P_F,τ).

kernels, NCLS was the only case that KNLCS could perform well but only had a slight improvement over KNCLS. As also noted earlier, this was mainly due to NCLS but not from the use of the kernels because the effectiveness of NCLS in mixed pixel classification has been demonstrated in previous experiments.

Finally, in order to further compare the relative performance of KLSMA using three different types of kennels, Figures 15.16–15.18 show 3D ROC curves in (a) along with their corresponding 2D ROC curves in (b–d) of the best performance resulting from each of KLSOSP, KNCLS, and KFCLS along with their counterparts, LSOSP, NCLS, and FCLS without using kernels for

Table 15.11 Values of A_z under three 2D ROC curves in Figure 15.14

	P_D versus P_F	P_D versus τ	P_F versus τ
NCLS	0.7285	0.4083	0.2354
KNCLS ($c = 0.1$)	0.7321	0.4178	0.2449
KNCLS ($c = 0.5$)	0.7313	0.4112	0.2306
KNCLS ($c = 1$)	0.6899	0.4061	0.2495
KNCLS ($c = 1.5$)	0.7083	0.4139	0.2512
KNCLS ($c = 2$)	0.6528	0.3872	0.2436



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

(d) 2D ROC curves of (P_F,τ)

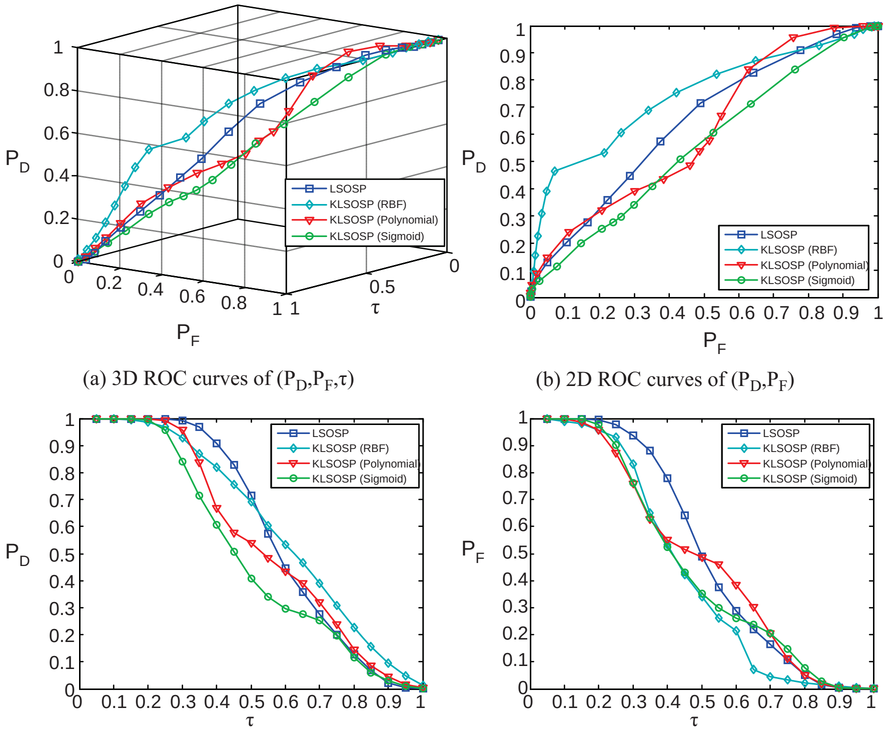
Figure 15.15 FCLS/KFCLS curves obtained by 16 classes average classification rate using sigmoid kernel with different σ (a) 3D ROC curves of (P_D,P_F,τ) (b) 2D ROC curves of (P_D,P_F) (c) 2D ROC curves of (P_D,τ) (d) 2D ROC curves of (P_F,τ).

comparison. Tables 15.13–15.15 calculate the area under each of 2D ROC curves, A_z for quantitative analysis where the shade rows indicate the best KLSMA classifier according to the 2D ROC curves of (P_D,P_F) plotted in Figures 15.16–15.18.

As shown in Figures 15.16–15.18 and Tables 15.13–15.15, the best kernel used for LSMA classifiers was RBF kernels except NCLS that used the polynomial kernel.

Table 15.12 Values of A_z under three 2D ROC curves in Figure 15.15

	P_D versus P_F	P_D versus τ	P_F versus τ
FCLS	0.5933	0.5064	0.4376
KFCLS ($c = 10e7$)	0.6327	0.4980	0.4407
KFCLS ($c = 20e8$)	0.6193	0.5360	0.4774
KFCLS ($c = 3 \times 10e8$)	0.5531	0.5220	0.4898
KFCLS (c = 5 × 10e8)	0.6327	0.5128	0.4545
KFCLS ($c = 10e9$)	0.4826	0.5722	0.5720



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

(d) 2D ROC curves of (P_F,τ)

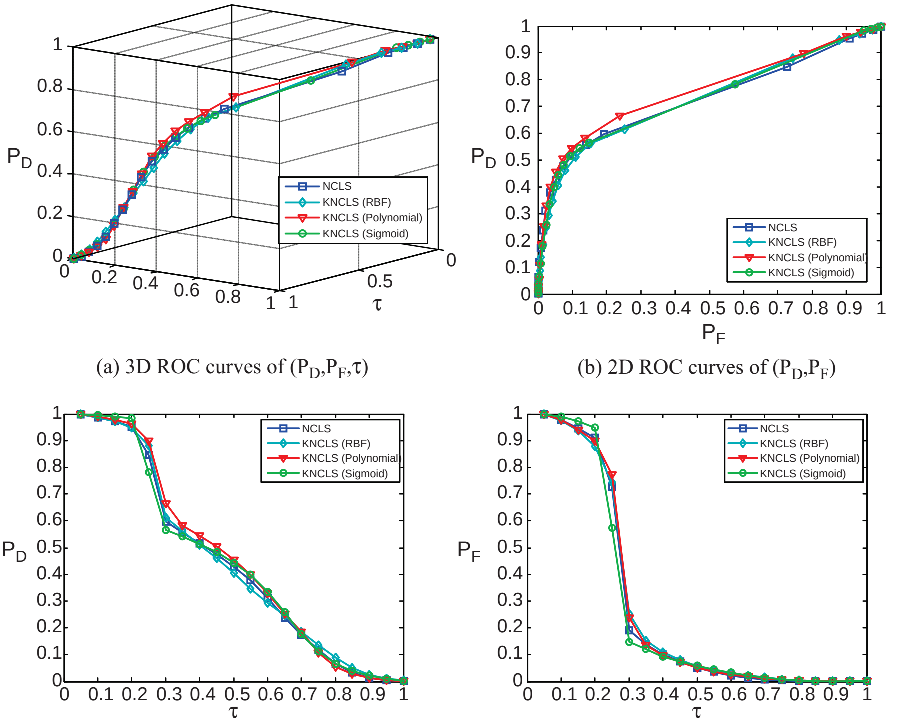
Figure 15.16 LSOSP/KLSOSP curves obtained by 16 classes average classification rate using three kernel function with appropriate parameters (a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of (P_F,τ).

Table 15.13 Values of A_z under three 2D ROC curves in Figure 15.16

	P_D versus P_F	P_D versus τ	P_{F} versus τ
LSOSP	0.6406	0.5498	0.4714
KLSOSP (RBF, $\sigma = 3000$)	0.7412	0.5675	0.3909
KLSOSP (polynomial, $p = 3$)	0.6110	0.5116	0.4396
KLSOSP (sigmoid, $c = 1$)	0.5581	0.4552	0.4174

Table 15.14 Values of A_z under three 2D ROC curves in Figure 15.17

	P_D versus P_F	P_D versus τ	P_F versus τ
NCLS	0.7285	0.4083	0.2354
KNCLS (RBF, $\sigma = 5000$)	0.7296	0.4115	0.2409
KNCLS (Polynomial, $p = 3$)	0.7584	0.4229	0.2396
KNCLS (sigmoid, $c = 0.5$)	0.7313	0.4112	0.2306



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

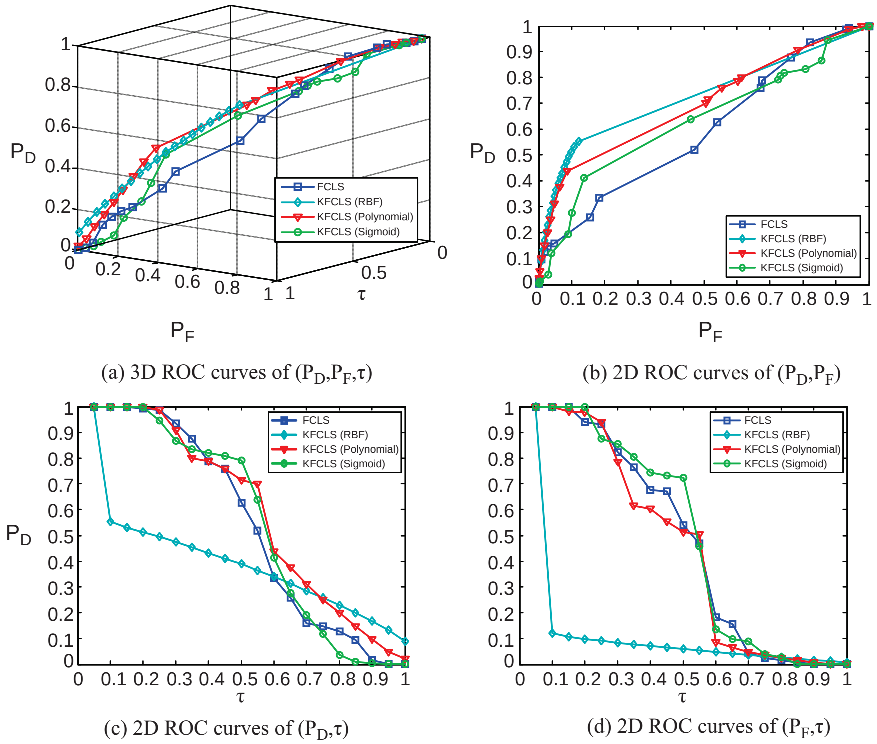
(d) 2D ROC curves of (P_F,τ)

Figure 15.17 NCLS/KNCLS curves obtained by 16 classes average classification rate using three kernel function with appropriate parameters (a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of (P_F,τ).

By concluding this section, KLSMA with an appropriate use of RBF kernels has shown potential in mixed pixel classification via the Purdue Indiana Indian Pine test site in which the image pixel vectors are heavily mixed. Such a heavy mixing may be also nonlinear in which case LSMA may not work effectively. However, as will be shown in the following section, when the sample vectors in a data set are not mixed as much as those in the Purdue's data, the advantages gained from the use of kernels by LSMA vanish.

Table 15.15 Values of A_z under three 2D ROC curves in Figure 15.18

	P_D versus P_F	P_D versus τ	P_F versus τ
FCLS	0.5933	0.5064	0.4376
KFCLS (RBF, $\sigma = 0.1$)	0.7258	0.3545	0.0783
KFCLS (polynomial, $p = 30$)	0.6963	0.5519	0.4130
KFCLS (sigmoid, $c = 5 \times 10^8$)	0.6327	0.5128	0.4545



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

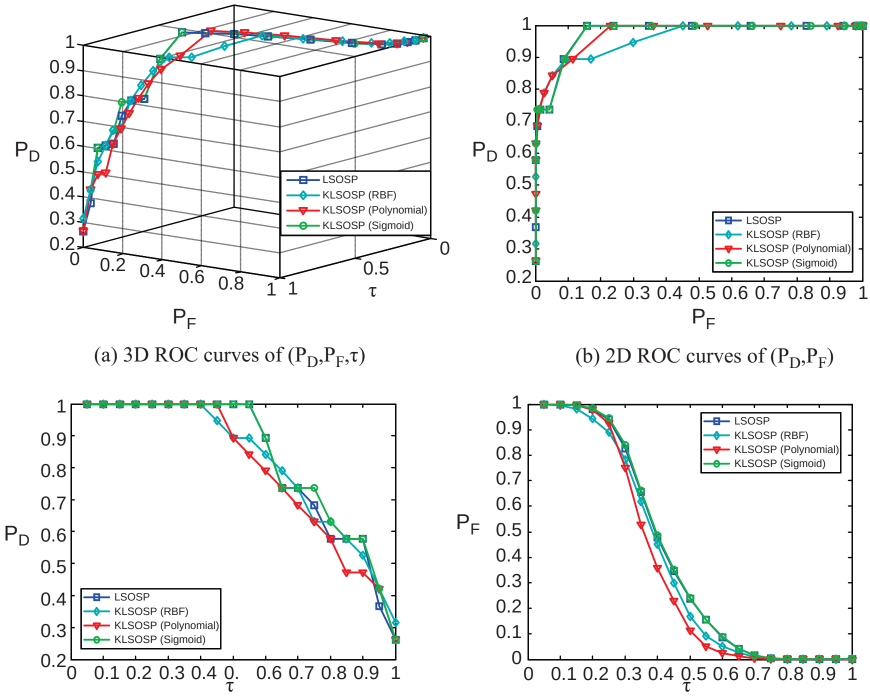
(d) 2D ROC curves of (P_F,τ)

Figure 15.18 FCLS/KFCLS curves obtained by 16 classes average classification rate using three kernel function with appropriate parameters (a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of (P_F,τ).

15.5 HYDICE Data Experiments

In this section, we used another real hyperspectral data set, HYDICE scene in Figure 1.15 for experiments that will show completely opposite results from those obtained from Purdue Indiana Indian Pine data set. First of all, VD estimated for this scene was used as $p = 9$ with the false alarm probability $P_F \leq 10^{-3}$. In this case, nine signatures were used for classification. These include five panel signatures in Figure 1.15(b) and other four undesired signatures, referred to as grass, road, tree, and interferer as identified in Figure 1.17. The LSMA performance was evaluated by detection of the 19 R panel pixels shown in 1.15(b) based on their abundance fractions unmixed by LSMA. The same 3D ROC analysis was also used for performance evaluation. Since the experiments on mixed pixel classification conducted for the HYDICE scene are previously reported in Chang (2003a), their results are not included here. Furthermore, the HYDICE experiments were performed in an exactly same manner that was conducted for Purdue data in Section 15.3. In this case, only those results similar to Figures 15.16–15.18 and Tables 15.13–15.15 are shown in Figures 15.19–15.21 and Tables 15.16–15.18 to avoid unnecessary redundancy where the unmixed results of KLSOSP, KNCLS, and KFCLS were obtained from the best empirical selection of their corresponding parameters as they were done for Purdue's data with shaded rows representing the best cases.

According to the results in Figures 15.19–15.21 and Tables 15.16–15.18 it was very obvious that using kernels did not provide any benefit for LSMA to perform better than LSMA without



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

(d) 2D ROC curves of (P_F,τ)

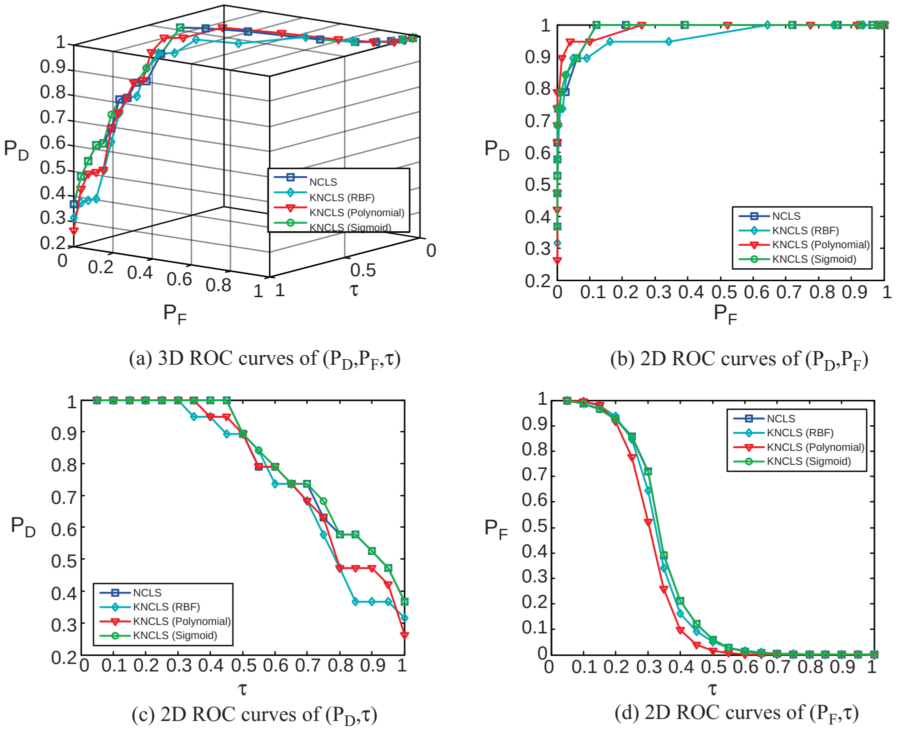
Figure 15.19 LSOSP/KLSOSP curves obtained by 19 R panel pixels with averaged classification rate using three kernel functions with appropriate parameters (a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of (P_F,τ).

Table 15.16 Values of areas under 2D ROC curves for Figure 15.19

	P_D versus P_F	P_D versus τ	P_F versus τ
LSOSP	0.9760	0.7895	0.3637
KLSOSP (RBF, $\sigma = 100000$)	0.9602	0.7776	0.3408
KLSOSP (polynomial, $p = 2$)	0.9739	0.7579	0.3237
KLSOSP (sigmoid, $c = 0.1$)	0.9763	0.7974	0.3661

Table 15.17 Values of A_z under three 2D ROC curves in Figure 15.20

	P_D versus P_F	P_D versus τ	P_F versus τ
NCLS	0.9851	0.7711	0.2896
KNCLS (RBF, $\sigma = 100000$)	0.9616	0.7250	0.2807
KNCLS (polynomial, $p = 2$)	0.9878	0.7447	0.2559
KNCLS (sigmoid, $c = 0.1$)	0.9863	0.7763	0.2896



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

(d) 2D ROC curves of (P_F,τ)

Figure 15.20 NCLS/KNCLS curves obtained by 19 R panel pixels with averaged classification rate using three kernel functions with appropriate parameters (a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of $(\mathrm{P_F},\tau)$.

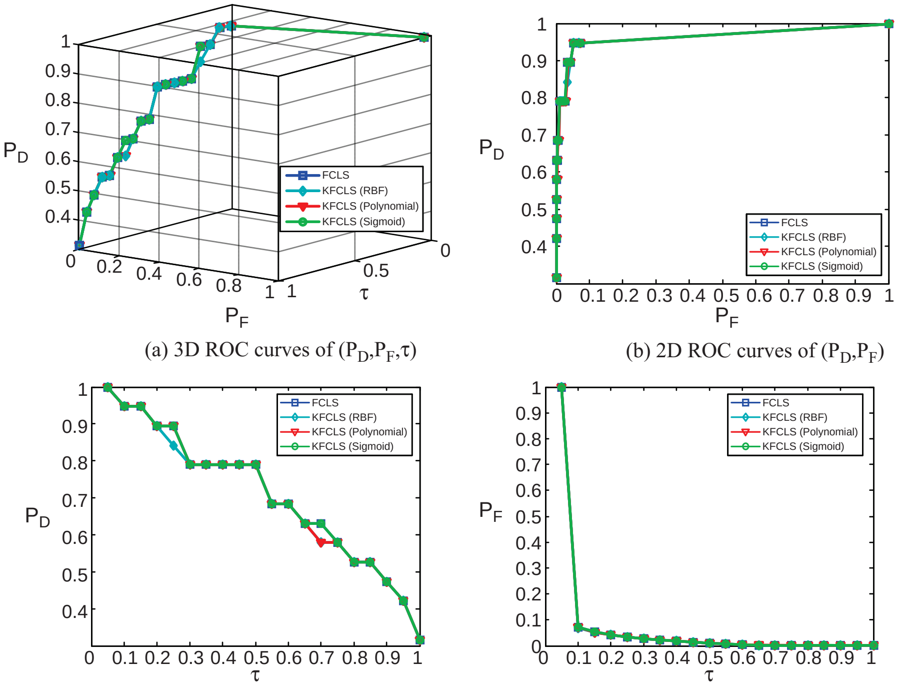
using kernels for the HYDICE scene as it did for the Purdue data. The main reason for this is because the panel pixels in the HYDICE image scene are mostly pure and least mixed even if they are not pure.

15.6 Conclusions

This chapter introduces a kernel version of LSMA, called kernel-based LSMA (KLSMA) to perform spectral unmixing in a feature space transformed by a nonlinear kernel function. Despite that a kernel-based OSP was also proposed by Kwon and Nasrabadi (2005) the derivation for the KOSP or KLSOSP presented in this chapter is much simpler than the one in Kwon and Nasrabadi (2005). Most importantly, it can be used as a base to extend NCLS and FCLS to KNCLS and KFCLS which were not developed in Kwon and Nasrabadi (2005). The kernel versions of NCLS and FCLS derived in this chapter are independent of that developed in Broadwater et al. (2007). In particular, the details of derivations for the three kernel-based algorithms, KLSOSP, KNCLS, and KFCLS including their step-by-step algorithmic implementations provided in this chapter are by far most comprehensive and can serve as guidelines for those who are interested in their implementations. It is also worth being mentioned that since the fundamental framework of kernelizing LSMA is laid out in this chapter, extensions of FLSMA in Chapter 13 and WACLSMA in Chapter 14 to their kernel counterparts can be carried out by a treatment similar to the one in extending LSMA to KLSMA presented in this chapter, but more complicated matrix manipulations are involved in their

Table 15.18 Values of areas under 2D ROC curves for Figure 15.21

	P_D versus P_F	P_D versus τ	P_F versus τ
FCLS	0.9646	0.6724	0.0406
KFCLS (RBF, $\sigma = 1$)	0.9643	0.6671	0.0404
KFCLS (polynomial, $p = 2$)	0.9642	0.6697	0.0411
KFCLS (sigmoid, $c = 10^6$)	0.9646	0.6724	0.0406



(a) 3D ROC curves of (P_D,P_F,τ)

(b) 2D ROC curves of (P_D,P_F)

(c) 2D ROC curves of (P_D,τ)

(d) 2D ROC curves of (P_F,τ)

Figure 15.21 FCLS/KFCLS curves obtained by 19 R panel pixels with averaged classification rate using three kernel functions with appropriate parameters (a) 3D ROC curves of (P_D,P_F,τ); (b) 2D ROC curves of (P_D,P_F); (c) 2D ROC curves of (P_D,τ); (d) 2D ROC curves of (P_F,τ).

derivations (Liu, 2011). Nevertheless, such extensions may not be as trivial as expected. In addition, to conduct quantification analysis for performance evaluation, a 3D ROC analysis is also used for this purpose. As expected, KLSMA should perform better than LSMA. While this is generally true for multispectral imagery, it may not be true for hyperspectral images as demonstrated by experiments conducted based on two rather different real hyperspectral image data sets, Purdue Indiana Indian Pine test site and HYDICE scene where the kernel-based classifiers can only significantly improve performance than those without using kernels only when the data sample vectors are heavily mixed. This is an interesting finding which may help those who design and develop hyperspectral imaging algorithms. In Chapters 31 and 32, multispectral image experiments will be also conducted for land cover classification where results provide evidence that the kernel-based approaches can be found to be more effective in multispectral image analysis than hyperspectral image analysis.

IV

Unsupervised Hyperspectral Image Analysis

Unsupervised target analysis is one of principal strengths that hyperspectral imaging has edge over multispectral image processing. This is due to its high spectral resolution that allows users to uncover and reveal many unknown signal sources such as subtle material substances, subpixel targets, mixed constituent compositions, anomalies, etc. However, this advantage also comes at a price that target analysis must be performed by unsupervised means because such targets of interest generally cannot be identified by visual inspection or prior knowledge. Three specific applications are of particular interest in hyperspectral data exploitation: target discrimination, unsupervised target detection, and unsupervised target classification. In unsupervised target detection target, knowledge is not provided *a priori* in which case potential targets must be obtained directly from the data to be processed without prior knowledge. In this case, an issue is how to discriminate one detected target from another. As for unsupervised target classification it is more challenging because two main key issues, which are not encountered in unsupervised target detection, must be addressed. One is how to determine the number of targets of interest assumed to be in the data for classification. The other is how to find unknown target training samples for classification without prior knowledge. Part IV is included to address these issues. Three chapters are included in this part.

Chapter 16 is devoted to hyperspectral measures that can be used to perform target discrimination. Two types of measures are considered: signature-based and correlation-weighted measures. While signature-based measures, such as spectral angle mapper (SAM), Euclidean distance (ED), spectral information divergence (SID), and orthogonal projection divergence (OPD), only deal with spectral information provided by all spectral bands, correlation-weighted measures, such as Mahalanobis distance and matched filter-based distance, take into account additional spectral correlation among signatures. In doing so the first task is to develop criteria that can be used to measure spectral similarity between two data sample vectors to discriminate one from another or

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

identify a specific target of interest from a set of known signatures such as data-bases or spectral libraries. Several commonly used signature-based spectral similarity measures are revisited and further extended to correlation-weighted hyperspectral measures when sample correlation among data samples is available for data processing in which case a weighting correlation matrix can be introduced into a signature-based spectral measure. Next, a follow-up task is to design and develop unsupervised target generation algorithms that allow us to extract potential targets in an unsupervised fashion with no prior knowledge required. These found targets can be used as so-called unsupervised target knowledge for target detection as well as a base to produce an appropriate set of training samples for target classification. Since unsupervised target analysis completely relies on the knowledge obtained directly from the data, it is important to analyze the information provided by data sample vectors. The issue of pixel information extracted from hyperspectral imagery is further explored for data analysis where four types of pixels are categorized, endmembers, mixed pixels, anomalies, and homogeneous pixels, in accordance with their spectral/spatial properties.

Chapter 17 takes up issues of determining the number of target signatures in the data and, in the mean time, finds these unknown target signatures. In order to address the first issue, a new concept of *spectral* targets is introduced to differentiate "*spatial*" targets that are generally identified by their spatial properties such as size, shape, and texture. More specifically, a target analyzed based on its spectral properties within a single signature vector is called *spectral target.* With such a defined spectral target what we are particularly interested in data analysis from an aspect of statistical signal processing is the concept of sample spectral statistics generated by interband spectral information (IBSI) among a set of data sample vectors, *S*, denoted by IBSI(*S*). There are two types of spectral targets based on sample spectral statistics, one characterized by the second-order IBSI and the other by sample intra-pixel IBSI of order higher than 2, referred to as high-order IBSI. It should be noted that the term of IBSI is defined as sample spectral correlation resulting from a set of data sample vectors specified by *S*. It is actually the size of *S* closely related to how to define second-order IBSI and high-order IBSI. In the context of IBSI we assume that background (BKG) pixels are those spectral targets characterized by second-order IBSI, while the target pixels of interest are those characterized by high-order IBSI. In hyperspectral image analysis this seems a reasonable assumption since the spectral targets of interest in hyperspectral data exploitation are those that either (1) occur with low probability or (2) have small populations when they are present. Such spectral targets generally appear in small population and also occur with low probabilities, for example, special spices in agriculture and ecology, toxic wastes in environmental monitoring, rare minerals in geology, drug/smuggler trafficking in law enforcement, combat vehicles in the battlefield, landmines in war zones, chemical/biological agents in bioterrorism, weapon concealment, and mass graves. As a result, the sample size of data sample vectors specified by such a spectral target, *S*, is relatively small and can be generally considered as insignificant objects because of their very limited spatial information, but they are actually critical and crucial for defense and intelligence analysis due to the fact that they are generally hard to be identified by visual inspection. From a statistical point of view since they are insignificant compared to targets with large sample pools, the spectral information statistics of such special targets cannot be captured by second-order IBSI but rather by high-order IBSI.

Once image pixel vectors are categorized into BKG and target classes according to IBSI, a follow-up task is how to find them, in which case two issues need to be addressed. One is how many of them. The other is how to extract them. The first issue can be resolved by the virtual dimensionality (VD) discussed in Chapter 5. The beauty of VD lies in the fact that its value is completely determined by the false alarm probability, P_F. By varying the value of P_F, the number of spectrally distinct signatures estimated by the VD varies. For example, if P_F is set too low, fewer tests will fail and thus too fewer targets are assumed to be in the data and vice versa. To address the second issue two approaches are developed to design an unsupervised target sample finding algorithm (UTSFA) to extract a set of target samples of interest directly from the data. One is based on three least squares (LS)-based algorithms, automatic target generation process (ATGP), unsupervised non-negativity constrained least squares (UNCLS) method, and unsupervised fully constrained least squares method (UFCLS). In order for these unsupervised methods to extract and distinguish spectral targets of second-order IBSI from high-order IBSI, two data sets, original data and its sphered data, are used. It assumes that the BKG in a hyperspectral image is most likely characterized by second-order IBSI while hyperspectral targets will be more likely to be captured by high-order IBSI as outliners due to their small spatial presence. In this case, high-order spectral targets are referred to as desired targets to be used for image analysis, while second-order spectral targets are considered as undesired targets for which we would like them to be annihilated or suppressed prior to data processing so as to improve image analysis. The other is component analysis-based algorithms where the principal components analysis (PCA) and independent component analysis (ICA) are used to accomplish what the LS-based algorithms described above do to extract second-order spectral targets and high-order spectral targets, respectively.

One of great challenges in hyperspectral data exploitation is analysis of pixel information extracted from various unsupervised algorithms. In remote sensing image processing, many algorithms have been developed for various applications in data exploitation. An important issue is whether these algorithms really do what they claim to do. For example, endmember extraction algorithms are designed to find endmembers that are assumed to be pure signatures. Is it really the case that their extracted endmembers are true pure signatures? Chapter 18 investigates the issue of pixel information extracted by three classes of exploitation-based algorithms: endmember extraction, unsupervised target detection, and anomaly detection. In order to facilitate pixel information analysis, four types of pixels are considered: pure pixel, mixed pixel, anomalous pixel, and homogeneous pixel. A pure pixel is a pixel whose spectral signature is completely specified by a single material substance as opposed to a mixed pixel whose spectral signature is made up of more than one material substance. According to classical image processing, a homogeneous region is defined by an area in which the pixels have very close gray scale values. Using the same idea we can also define a homogenous neighborhood as a set of neighboring pixels whose spectral signatures are very close and similar. A pixel falling in a homogeneous neighborhood is called a homogeneous pixel. The concept of a homogenous pixel is completely opposite to an anomalous pixel, whose signature is considered to be spectrally distinct from those of its surrounding pixels. While pure pixels and mixed pixels can be analyzed by their spectral properties on a single pixel basis, homogeneous pixels and anomalous pixels must take into account the surrounding pixels within their neighborhoods, that is, both spatial and spectral properties. It is interesting to note that a pure or mixed pixel can be homogenous or anomalous. Similarly, a homogeneous or an anomalous pixel can also be pure or mixed. These four types of pixels can help to shed light on the pixel information that a particular algorithm is designed to extract. Of particular interest is that if the extracted endmembers happen to be pixels, are they really endmember pixels? If not, what type of pixels are they?

16

Hyperspectral Measures

One simplest and easiest means to conduct unsupervised target analysis is to use hyperspectral measures for target discrimination detection, classification, recognition, and identification. Many hyperspectral measures have been studied in the literature, particularly in Chang (2003a, Chapter 2). They are primarily designed to measure spectral similarity among signatures for the purpose of detection, discrimination, classification, and identification. This chapter revisits several commonly used signature vector-based spectral similarity measures and further generalizes signature vector-based hyperspectral measures to sample correlation-weighted hyperspectral measures by including a weighting correlation matrix into a signature vector-based spectral measure so as to improve its performance. The idea of such generalization is similar to that using a weighting matrix to extend linear spectral mixture analysis (LSMA) to weighted abundance-constrained LSMA (WAC-LSMA) developed in Chapter 14.

16.1 Introduction

A traditional approach to designing hyperspectral measures assumes that data samples are not necessarily collected from imaging sensors as image pixels and could be also from other types of non-imaging optical sensors. In this case, the data samples should be analyzed as one-dimensional signals on the basis of their spectral characteristics (see Chapter 2, Chang 2003a; Category 2: Hyperspectral Signal Processing, Chapters 24–29). Since there is no prior knowledge available regarding the data sample vectors to be analyzed, the target analysis must be performed by some sort of an unsupervised fashion. In certain applications such as chemical/biological (CB) warfare defense there is a spectral library or database that can be used to identify unknown CB agents for target discrimination and target identification where in the former case, unknown data samples can be only discriminated one from another, while in the latter case an unknown data sample can be identified by comparing its spectral profile against the spectral signatures in a database or spectral library. However, technically speaking, such a target identification is actually target verification because it does not perform identification but rather verifies data sample vectors of interest via an existing data base or spectral library. To do so, signature vector-based spectral measures such as spectral angle mapper (SAM) and spectral information divergence (SID) developed in Chapter 2 of Chang (2003a) are generally used for this purpose. On many occasions a set of collected unknown data samples may be correlated their spectral profiles with reference data in a database or spectral library that are used to compare against data samples. Under this circumstance, the data sample correlation should help to increase spectral discriminability. Since signature vector-based spectral measures do not take

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

advantage of sample correlation, a weighting matrix that accounts for correlation resulting from sample pools is included in signature vector-based spectral measures whenever sample correlation is available to derive weighted spectral measures. This chapter investigates these two types of hyperspectral measures, signature vector-based and correlation-based spectral measures in various applications.

16.2 Signature Vector-Based Hyperspectral Measures for Target Discrimanition and Identification

Spectral characteristics provide important and crucial features in material identification, discrimination, detection, and classification. Many spectral similarity measures have been developed and can be used for this purpose such as SAM (Schwengerdt, 1997), SID (Chang, 2000, 2003a, Chapter 2), Euclidean distance (ED), and many others (Chang, 2003a). When there is no prior target class information available, these measures are performed on a single signature vector basis to measure spectral variability between two signature vectors, in which case they are generally used for signature discrimination and identification, but not used for classification. Furthermore, they are effective only if the spectral signature vectors to be compared are true signatures of the materials that they really represent the signature vectors. However, this idealistic case is generally not true in many real applications where many factors may contaminate and corrupt spectral signature vectors to be identified. One scenario is mixed signature discrimination and identification where a spectral signature vector is mixed with a number of signature vectors resident in the signature vector. Another is subsample target discrimination and identification where the target to be identified is embedded in a single signature vector and its spectral signature vector is clearly mixed with other signature vectors that are also present in the signature vector. In either case, single signature vector-based spectral measures such as SAM may not work effectively and sometimes may even identify wrong targets. This section investigates the issue of discrimination and identification for mixed signature vectors and subsample targets and provides evidence that such examples indeed occur in real hyperspectral imagery where some commonly used spectral measures fail in discrimination and identification of mixed signature vectors and subsample targets. To remedy this problem it further develops new spectral measures for mixed signature vectors and subsample targets in identification and discrimination. Unlike single signature vector-based spectral measures described above, these measures take advantage of the sample spectral correlation to account for spectral variability of mixed signature vectors and subsample targets within the signature vectors. In particular, when a signature vector is an image pixel vector in an image data where mixed pixel vectors or subpixel targets may spatially and spectrally correlated with their neighboring pixel vectors, the inclusion of such sample spectral correlation in a spectral measure offers additional spectral information that any single signature vector-based spectral measure cannot provide. Two types of sample spectral correlation-based spectral measures are of interest. They are previously developed as target discrimination measures for anomaly classification in hyperspectral imagery and can be considered as candidates to be studied. One is Mahalanobis distance (MD)-based and the other is matched filter-based hyperspectral measures, both of which include the sample spectral covariance/correlation matrix to capture spectral mixed signature vectors and subsample target signature vectors more effectively. Due to the use of the sample spectral covariance/correlation matrix, these measures can be considered as second-order spectral measures as opposed to single signature vector-based spectral measures that can be thought of as first-order spectral measures which do not include the sample covariance/correlation matrix to account for sample correlation.

In what follows, we describe four signature vector-based spectral measures, ED SAM, OPD, and SID, all of which are discussed in Chang (2003a) and closely related to each other in one way or another. For example, when the angle between two signatures is small, ED and SAM are essentially the same measures as shown in Chang (2003a). On the other hand, the pair of SAM and OPD can be related by orthogonal projection. Additionally, while the OPD measures the divergence of one signature vector projection onto the other, SID can be considered as a stochastic version of OPD with orthogonal projection replaced with information divergence (Cover and Thomas, 1991). The definitions of these four measures are summarized as follows (Chang, 2003a).

Assume that $\mathbf{s}_i = (s_{i1}, s_{i2}, \ldots, s_{iL})^T$ and $\mathbf{s}_j = \left(s_{j1}, s_{j2}, \ldots, s_{jL}\right)^T$ are two spectral signature vectors where L is the total number of spectral bands.

16.2.1 Euclidean Distance

The ED is one of most widely used metrics in mathematics to measure the distance between two spectral signatures, $\mathbf{s}_i$ and $\mathbf{s}_j$, given by

$$\mathrm{ED}(\mathbf{s}_i, \mathbf{s}_j) = ||\mathbf{s}_i - \mathbf{s}_j|| = \sqrt{\sum_{l=1}^{L} \left(s_{il} - s_{jl}\right)^2} \tag{16.1}$$

16.2.2 Spectral Angle Mapper

The SAM measures spectral similarity by finding the angle between the spectral signatures $\mathbf{s}_i$ and $\mathbf{s}_j$

$$\mathrm{SAM}(\mathbf{s}_i, \mathbf{s}_j) = \cos^{-1}\left(\frac{\langle \mathbf{s}_i, \mathbf{s}_j \rangle}{||\mathbf{s}_i||||\mathbf{s}_j||}\right) \tag{16.2}$$

where $\langle \mathbf{s}_i, \mathbf{s}_j \rangle = \sum_{l=1}^{L} s_{il} s_{jl\,jl}$, $||\mathbf{s}_i|| = \left(\sum_{l=1}^{L} (s_{il})^2\right)^{1/2}$ and $||\mathbf{s}_j|| = \left(\sum_{l=1}^{L} \left(s_{jl}\right)^2\right)^{1/2}$.

16.2.3 Orthogonal Projection Divergence

The concept of the OPD is first defined in Chang (2003a) which is originated from the orthogonal subspace projection (OSP) developed in Harsanyi and Chang (1994). It finds the residuals of orthogonal projections resulting from two pixel vectors, $\mathbf{s}_i$ and $\mathbf{s}_j$ given by

$$\mathrm{OPD}(\mathbf{s}_i, \mathbf{s}_j) = \left(\mathbf{s}_i^T P_{\mathbf{s}_j}^{\perp} \mathbf{s}_i + \mathbf{s}_j^T P_{\mathbf{s}_i}^{\perp} \mathbf{s}_j\right)^{1/2} \tag{16.3}$$

where $P_{\mathbf{s}_k}^{\perp} = \mathbf{I} - \mathbf{s}_k \left(\mathbf{s}_k^T \mathbf{s}_k\right)^{-1} \mathbf{s}_k^T$ for $k = i,j$ and $\mathbf{I}$ is the $L \times L$ identity matrix.

16.2.4 Spectral Information Divergence

Let $\mathbf{p} = (p_1, p_2, \ldots, p_L)^T$ and $\mathbf{q} = (q_1, q_2, \ldots, q_L)^T$ be the two probability mass functions generated by $\mathbf{s}_i$ and $\mathbf{s}_j$, respectively, with $p_l = s_{il} / \sum_{l=1}^{L} s_{il}$ and $q_l = s_{jl} / \sum_{l=1}^{L} s_{jl}$. So, the self-information provided by $\mathbf{s}_i$ and $\mathbf{s}_j$ for band l is defined by

$$I_l(\mathbf{s}_i) = -\log_{pl} \tag{16.4}$$

$$I_l(\mathbf{s}_j) = -\log_{ql} \tag{16.5}$$

respectively. By virtue of (16.4) and (16.5) we can define the discrepancy of the self-information of band image B_l provided by $\mathbf{s}_j$ relative to the self-information of band image B_l provided by $\mathbf{s}_i$, denoted by $D_l(\mathbf{s}_i||\mathbf{s}_j)$ as

$$D_l(\mathbf{s}_i||\mathbf{s}_j) = I_l(\mathbf{s}_i) - I_l(\mathbf{s}_j) = \log(p_l/q_l) \tag{16.6}$$

Averaging $D(\mathbf{s}_i||\mathbf{s}_j)$ in (16.6) over all the band images $\{B_l\}_{l=1}^{L}$ results in

$$D(\mathbf{s}_i||\mathbf{s}_j) = \sum\nolimits_{l=1}^{L} D_l(\mathbf{s}_i||\mathbf{s}_j)p_l = \sum\nolimits_{l=1}^{L} p_l \log(p_l/q_l) \tag{16.7}$$

where $D(\mathbf{s}_i||\mathbf{s}_j)$ is the average discrepancy in the self-information of $\mathbf{s}_j$ relative to the self-information of $\mathbf{s}_i$. In context of information theory, $D(\mathbf{s}_i||\mathbf{s}_j)$ in (16.7) is called the relative entropy of $\mathbf{s}_j$ with respect to $\mathbf{s}_i$, which is also known as Kullback–Leibler information measure, directed divergence or cross entropy (Cover and Thomas, 1991). Similarly, we can also define the average discrepancy in the self-information of $\mathbf{s}_i$ relative to the self-information of $\mathbf{s}_j$ by

$$D(\mathbf{s}_j||\mathbf{s}_i) = \sum\nolimits_{l=1}^{L} D_l(\mathbf{s}_j||\mathbf{s}_i)q_l = \sum\nolimits_{l=1}^{L} q_l \log(q_l/p_l) \tag{16.8}$$

Summing (16.7) and (16.8) yields spectral information divergence (SID) defined by

$$\mathrm{SID}(\mathbf{s}_i, \mathbf{s}_j) = D(\mathbf{s}_i||\mathbf{s}_j) + D(\mathbf{s}_j||\mathbf{s}_i) \tag{16.9}$$

which can be used to measure the discrepancy between two pixel vectors $\mathbf{s}_i$ and $\mathbf{s}_j$ in terms of their corresponding probability mass functions, $\mathbf{p}$ and $\mathbf{q}$. It should be noted that while $\mathrm{SID}(\mathbf{s}_i, \mathbf{s}_j)$ is symmetric, $D(\mathbf{s}_i||\mathbf{s}_j)$ is not. This is because $\mathrm{SID}(\mathbf{s}_i, \mathbf{s}_j) = \mathrm{SID}(\mathbf{s}_j, \mathbf{s}_i)$ and $D(\mathbf{s}_i||\mathbf{s}_j) \neq D(\mathbf{s}_j||\mathbf{s}_i)$. As a final remark on SID, it is worth noting that a recent work (Du et al., 2004) suggested a means of mixing SID and SAM as SID-SAM mixed measures specified by $\mathrm{SID}(\mathbf{s}_i, \mathbf{s}_j)$xtan $(\mathrm{SAM}(\mathbf{s}_i, \mathbf{s}_j))$ and $\mathrm{SID}(\mathbf{s}_i, \mathbf{s}_j)$xsin$(\mathrm{SAM}(\mathbf{s}_i, \mathbf{s}_j))$. Their results have shown better discriminability in spectral similarity. Those who are interested in these measures can find more details in Du et al. (2004).

16.3 Correlation-Weighted Hyperspectral Measures for Target Discrimanition and Identification

The signature vector-based spectral measures described in Section 16.2 calculate the spectral similarity value between a pair of two signature vectors using only the spectral information provided by L bands within these two signature vectors. So, if a material signature vector is mixed by other substances, the spectral characteristics of the signature vector to be processed do not necessarily characterize the spectral properties of the material signature vector it represents. This often occurs in real applications when a material signature vector is either mixed with other signature vectors such as background signatures or embedded in a single signature vector as a subsample target. In both cases, using a signature vector-based spectral measure to measure material similarity is generally not effective. In order to resolve this dilemma, signature vector-based hyeprspectral measures are extended to correlation-weighted hyperspectral measures that can be categorized into two classes. One comprises of hyperspectral measures that introduce *a priori* sample spectral correlation into signature vector-based spectral measure so as to improve discrimination performance in spectral similarity. The other is made up of hyperspectral measures weighted by *a posteriori* sample spectral correlation to do what *a priori* sample spectral correlation does.

16.3.1 Hyperspectral Measures Weighted by A Priori Correlation

A hyperspectral measure weighted by *a priori* correlation assumes that there is known correlation available prior to material discrimination and identification. A good example of best utilization of such prior correlation is the OSP approach recently developed by Harsanyi and Chang (1994) that separates desired target signature vectors from undesired target signature vectors to achieve better hyperspectral image classification. This OSP concept can be used to design new hyperspectral measures as follows.

Assume that there are p target signature vectors which are known *a priori* and $\mathbf{s}_i$ is a signature vector of interest. We can define a matrix, $\mathbf{U}$ to be a matrix formed by all *a priori* known target signature vectors except $\mathbf{s}_i$. By taking advantage of the following orthogonal projector defined by Harsanyi and Chang (1994) or (2.86) in Chapter 2:

$$P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T \tag{2.86}$$

we can de-correlate the signature vector $\mathbf{s}_i$ via orthogonal projection from all other known target signatures in $\mathbf{U}$ by projecting $\mathbf{s}_i$ to the space $\langle\mathbf{U}\rangle^{\perp}$ orthogonal to the space linearly spanned by undesired target signature vectors in $\mathbf{U}$, and derive a family of several OSP-based hyperspectral measures. Depending on how to use $P_{\mathbf{U}}^{\perp}$, these OSP-based hyperspectral measures can be either used for discrimination or identification.

16.3.1.1 OSP-Based Hyperspectral Measures for Discrimination

Two separate problems, discrimination, and identification are considered. When discrimination is performed, it only needs to discriminate one signature vector from another. When identification is performed, we generally assume that there is a database, $\Delta = \{\mathbf{d}_k\}_{k=1}^{K}$ available for signature identification.

Two OSP-based hyperspectral measures can be designed for discrimination between two signature vectors $\mathbf{s}_i$ and $\mathbf{s}_j$. One includes $P_{\mathbf{U}}^{\perp}$ as a weighting matrix into ED in (16.2), called $\mathrm{ED}_{\mathrm{OSP}}$, defined by

$$\mathrm{ED}_{\mathrm{OSP}}(\mathbf{s}_i, \mathbf{s}_j) = \left(\mathbf{s}_i - \mathbf{s}_j\right)^T P_{\mathbf{U}}^{\perp}\left(\mathbf{s}_i - \mathbf{s}_j\right) \tag{16.10}$$

where $\mathbf{U}$ is an undesired signature matrix formed by all known signatures excluding $\mathbf{s}_i$ and $\mathbf{s}_j$.

The other is derived from the concept of OPD in (16.3), called OSP-based divergence, $\mathrm{D}_{\mathrm{OSP}}$, defined by

$$\mathrm{D}_{\mathrm{OSP}}(\mathbf{s}_i, \mathbf{s}_j) = |\mathbf{s}_i^T P_{\mathbf{U}}^{\perp}\mathbf{s}_j| \tag{16.11}$$

with the same $\mathbf{U}$ defined in (16.10).

If $P_{\mathbf{U}}^{\perp}$ in (16.10) is assumed to be the identity matrix, then (16.10) is reduced to ED in (16.1). On the other hand, (16.11) is an extension of the OPD by including $P_{\mathbf{U}}^{\perp}$ to eliminate the interference caused by undesired target signature vectors. However, it should be noted that unlike the $\mathrm{ED}_{\mathrm{OSP}}$, which is a nonnegative measure, the $\mathbf{s}_i^T P_{\mathbf{U}}^{\perp}\mathbf{s}_j$ in (16.11) can take positive or negative values depending upon whether or not $\mathbf{s}_i$ and $\mathbf{s}_j$ point to the same direction. To avoid this problem, the absolute value is used in (16.11).

16.3.1.2 OSP-Based Hyperspectral Measures for Identification

In the previous subsection, the signature vectors $\mathbf{s}_i$ and $\mathbf{s}_j$ are assumed to be signature vectors in real data and (16.10) and (16.11) are used for signature discrimination. In this subsection, we assume that there is a database or spectral library $\Delta = \{\mathbf{d}_k\}_{k=1}^{K}$ available to be used to identify a signature vector in real data $\mathbf{s}_i$. In this case, $\mathrm{D}_{\mathrm{OSP}}$ can be modified to perform signature identification in two different ways depending on how to use the matched signature either from the database Δ, denoted by $\mathrm{ID}_{\mathrm{OSP},\Delta}$ or itself, denoted by $\mathrm{ID}_{\mathrm{OSP}}$ as follows:

$$\mathrm{ID}_{\mathrm{OSP},\Delta}(\mathbf{s}_i) = \max_{\mathbf{d}_k \in \Delta} |\mathbf{d}_k^T P_{\mathbf{U}_k}^{\perp} \mathbf{s}_i| \tag{16.12}$$

$$\mathrm{ID}_{\mathrm{OSP}}(\mathbf{s}_i) = \max_{\mathbf{U}_k} \mathbf{s}_i^T P_{\mathbf{U}_k}^{\perp} \mathbf{s}_i \tag{16.13}$$

where $\mathbf{U}_k$ is a matrix formed by all signature vectors in Δ excluding signature vector $\mathbf{d}_k$.

Using (16.12) and (16.13) a signature vector $\mathbf{s}_i$ can be identified via a database Δ by the one in Δ that yields the $\mathrm{ID}_{\mathrm{OSP},\Delta}$ or $\mathrm{ID}_{\mathrm{OSP}}$.

16.3.2 Hyperspectral Measures Weighted by A Posteriori Correlation

In many applications, obtaining *a priori* correlation information is very difficult, if not impossible. Therefore, it is highly desirable if we can generate necessary information directly from the image data without relying on prior knowledge. It has been demonstrated in Chapter 12 that the *a priori* correlation provided by $P_{\mathbf{U}}^{\perp}$ can be approximated by the inverse of the sample spectral correlation matrix, $\mathbf{R}^{-1}$, which can be used to account for *a posteriori* correlation. In light of this interpretation, the sample spectral correlation/covariance is used to derive new *a posteriori* correlation-weighted hyperspectral measures.

16.3.2.1 Covariance Matrix-Weighted Hyperspectral Measures

As noted in Section 12.5, the RX detector (RXD) defined by (12.76) was developed by Reed and Yu (1990) for anomaly detection. If the $\mathbf{r}$ and $\boldsymbol{\mu}$ in (12.76) are replaced with $\mathbf{s}_i$ and $\mathbf{s}_j$, respectively, then the RX anomaly detector becomes

$$\left(\mathbf{s}_i - \mathbf{s}_j\right)^T \mathbf{K}^{-1} \left(\mathbf{s}_i - \mathbf{s}_j\right) \tag{16.14}$$

Since (16.14) is a Mahalanobis distance (MD)-like measure, we can define a new hyperspectral MD-like measure via (16.14), called $\mathrm{MD}_{\mathrm{RX}}$ as follows:

$$\mathrm{MD}_{\mathrm{RX}}\left(\mathbf{s}_i, \mathbf{s}_j\right) = \left(\mathbf{s}_i - \mathbf{s}_j\right)^T \mathbf{K}^{-1} \left(\mathbf{s}_i - \mathbf{s}_j\right) \tag{16.15}$$

Two comments on (16.15) are noteworthy.

1. The covariance matrix-weighted hyperspectral measure, $\mathrm{MD}_{\mathrm{RX}}$ defined by (16.15) is essentially derived from the widely used maximum likelihood classifier (MLC) by replacing $\mathbf{s}_i$ and $\mathbf{s}_j$ in (16.15) with a data sample vector $\mathbf{r}$ to be classifier and the mean of the jth class, $\boldsymbol{\mu}_j$, (16.15), respectively, as follows:

$$\mathrm{MLC}(\mathbf{r}) = \arg\left\{\min_{1 \le j \le p} \left(\mathbf{r} - \boldsymbol{\mu}_j\right)^T \mathbf{K}^{-1} \left(\mathbf{r} - \boldsymbol{\mu}_j\right)\right\} \tag{16.16}$$

   where MLC assigns to the data sample vector a class that yields the minimum of $\left(\mathbf{r} - \boldsymbol{\mu}_j\right)^T \mathbf{K}^{-1} \left(\mathbf{r} - \boldsymbol{\mu}_j\right)$ over all the p classes, $C_1, C_2, \ldots, C_p$.
2. Since the MLC described in (16.16) is supervised in the sense that the knowledge of $\left\{\boldsymbol{\mu}_j\right\}_{j=1}^p$, means of p classes, $\left\{C_j\right\}_{j=1}^p$ must be available *a priori*, the MLC in (16.16) can

be further improved by including an undesired signature annihilator defined by (2.86) operating on the data sample vector in (16.16) in a similar manner that the OSP is defined by (2.85) as follows:

$$\text{MLC}_{\text{OSP}}(\mathbf{r}) = \arg\left\{\min_{1\le j\le p}\left(P_{\mathbf{U}_j}^{\perp}\mathbf{r} - \boldsymbol{\mu}_j\right)^T \mathbf{K}^{-1}\left(P_{\mathbf{U}_j}^{\perp}\mathbf{r} - \boldsymbol{\mu}_j\right)\right\} \tag{16.17}$$

where $\mathbf{U}_j$ is a matrix made up of all signatures excluding $\mathbf{s}_j$, that is, $\mathbf{U}_j = \left[\mathbf{s}_1 \ldots \mathbf{s}_{j-1}\mathbf{s}_{j+1} \ldots \mathbf{s}_p\right]$.

3. In (16.16) and (16.17) the knowledge of the number of classes, p and $\left\{\boldsymbol{\mu}_j\right\}_{j=1}^{p}$ is assumed to be known *a priori*. If such knowledge is not available, there is impossible to perform classification. The class means $\boldsymbol{\mu}_j$ in (16.16) must be replaced by the global mean $\boldsymbol{\mu}$. As a result, the MLC specified by (16.16) becomes the well-known anomaly detector, RXD defined by (12.76):

$$(\mathbf{r} - \boldsymbol{\mu})^T\mathbf{K}^{-1}(\mathbf{r} - \boldsymbol{\mu}) \tag{12.76}$$

16.3.2.2 Correlation Matrix-Weighted Hyperspectral Measures

As also noted in Section 12.4.1.2, replacing the $P_{\mathbf{U}}^{\perp}$ used in the OSP in (12.22) with the inverse of spectral correlation matrix, $\mathbf{R}^{-1}$ yields the constrained energy minimization (CEM) in (12.52). Using the same token we can also define a new hyperspectral measure weighted by *a posteriori* correlation from the ED_{OSP}, called MD_{CEM} defined by

$$\text{MD}_{\text{CEM}}\left(\mathbf{s}_i, \mathbf{s}_j\right) = \left(\mathbf{s}_i - \mathbf{s}_j\right)^T\mathbf{R}^{-1}\left(\mathbf{s}_i - \mathbf{s}_j\right) \tag{16.18}$$

Interestingly, comparing (16.18) to (16.15) MD_{RX} and MD_{CEM} have the same identical structure except the *a posteriori* correlation in (16.15) provided by $\mathbf{K}^{-1}$ as opposed to $\mathbf{R}^{-1}$ used to account for *a posteriori* correlation in (16.18).

As special cases where the sample spectral covariance matrix $\mathbf{K}$ and the sample spectral correlation matrix $\mathbf{R}$ are whitened, that is, $\mathbf{K} = \mathbf{I}$ and $\mathbf{R} = \mathbf{I}$, both (16.15) and (16.18) are reduced to ED

$$\begin{aligned}\text{MD}_{\text{RX},\mathbf{K}=\mathbf{I}}(\mathbf{s}_i, \mathbf{s}_j) &= \text{MD}_{\text{CEM},\mathbf{R}=\mathbf{I}}(\mathbf{s}_i, \mathbf{s}_j) = \left(\mathbf{s}_i - \mathbf{s}_j\right)^T\left(\mathbf{s}_i - \mathbf{s}_j\right) \\ &= ||\mathbf{s}_i - \mathbf{s}_j||^2 = \text{ED}(\mathbf{s}_i, \mathbf{s}_j)\end{aligned} \tag{16.19}$$

16.3.2.3 Covariance Matrix-Weighted Matched Filter Distance

According to RXD specified by (12.76), we can also derive a new hyperspectral measure, denoted by MFD_{RX} by replacing the both $\mathbf{r}$'s in (12.76) with $\mathbf{s}_j$ and $\mathbf{s}_i$, respectively, as follows:

$$\text{MFD}_{\text{RX}}\left(\mathbf{s}_i, \mathbf{s}_j\right) = |(\mathbf{s}_i - \boldsymbol{\mu})^T\mathbf{K}^{-1}\left(\mathbf{s}_j - \boldsymbol{\mu}\right)| \tag{16.20}$$

where $\boldsymbol{\mu}$ is the sample mean of the data to be processed. Since $\mathbf{s}_j$ and $\mathbf{s}_i$ are not the same signature vectors, the values produced by $(\mathbf{s}_i - \boldsymbol{\mu})^T\mathbf{K}^{-1}\left(\mathbf{s}_j - \boldsymbol{\mu}\right)$ in (16.20) are not necessarily non-negative. In this case, the absolute value is used in (16.20).

16.3.2.4 Correlation Matrix-Weighted Matched Filter Distance

As an alternative, we can also take advantage of CEM in another way to account for *a posteriori* correlation by designing a new hyperspectral measure, called CEM-matched filter distance, $\mathrm{MFD}_{\mathrm{CEM}}$ defined by

$$\mathrm{MFD}_{\mathrm{CEM}}\left(\mathbf{s}_i, \mathbf{s}_j\right) = |\mathbf{s}_i^T \mathbf{R}^{-1} \mathbf{s}_j| \tag{16.21}$$

Comparing (16.21) with (16.20), (16.20) can be also obtained by replacing $\mathbf{s}_i$, $\mathbf{R}^{-1}$ and $\mathbf{s}_j$ with $\mathbf{s}_i - \boldsymbol{\mu}$, $\mathbf{K}^{-1}$ and $\mathbf{s}_j - \boldsymbol{\mu}$, respectively.

In analogy with (16.17), if $\mathbf{K}$ and $\mathbf{R}$ are whitened, that is, $\mathbf{K} = \mathbf{I}$ and $\mathbf{R} = \mathbf{I}$, and $\boldsymbol{\mu} = \mathbf{0}$, (16.20) and (16.21) are reduced to SAM

$$\begin{aligned} \mathrm{MFD}_{\mathrm{CEM},\mathbf{R}=\mathbf{I}}(\mathbf{s}_i, \mathbf{s}_j) &= \mathrm{MFD}_{\mathrm{RX},\mathbf{R}=\mathbf{I}}(s_i, s_j) = |s_i^T s_j| = |\langle s_i, s_j \rangle| \\ &= \left(||s_i|| ||s_j||\right)\left(\left|\cos\left(\mathrm{SAM}(s_i, s_j)\right)\right|\right) \end{aligned} \tag{16.22}$$

where SAM($\mathbf{s}_i$,$\mathbf{s}_j$) is defined by (16.2). Interestingly, the four proposed *a posteriori* correlation-based hyperspectral measures, $\mathrm{MD}_{\mathrm{RX}}$, $\mathrm{MD}_{\mathrm{CEM}}$, $\mathrm{MFD}_{\mathrm{RX}}$, and $\mathrm{MFD}_{\mathrm{CEM}}$ turn out to be the same four target discrimination measures developed in Chang and Chiang (2002).

Recently, an adaptive coherence estimator (ACE) developed by Kraut et al. (1999) and

$$\delta^{\mathrm{ACE}}(\mathbf{d}, \mathbf{r}) = \frac{\left(\mathbf{d}^T \mathbf{K}^{-1} \mathbf{r}\right)^2}{\left(\mathbf{d}^T \mathbf{K}^{-1} \mathbf{d}\right)\left(\mathbf{r}^T \mathbf{K}^{-1} \mathbf{r}\right)} \tag{16.23}$$

where $\mathbf{d}$ is the target signal to be detected and $\mathbf{r}$ is a data sample vector to be processed. By virtue of (16.23) we can further define a new discrimination measure, denoted by $\mathrm{MFD}_{\mathrm{ACE}}$ by replacing $\mathbf{d}$ with $\mathbf{s}_i$ and $\mathbf{r}$ with $\mathbf{s}_j$ as

$$\mathrm{MFD}_{\mathrm{ACE}}(\mathbf{s}_i, \mathbf{s}_j) = \frac{\left|\mathbf{s}_i^T \mathbf{K}^{-1} \mathbf{s}_j\right|}{\left(\mathbf{s}_i^T \mathbf{K}^{-1} \mathbf{s}_i\right)^{1/2} \left(\mathbf{s}_j^T \mathbf{K}^{-1} \mathbf{s}_j\right)^{1/2}}. \tag{16.24}$$

If we further use $\mathbf{K}^{-1/2}$ as a whitening matrix as discussed in Section 6.3.1 of Chapter 6 (i.e., making $\mathbf{K}$ an identity matrix $\mathbf{I}$ by $\mathbf{K}$=$\mathbf{I}$) and define $\tilde{\mathbf{d}} = \mathbf{K}^{-1/2}\mathbf{d}$, $\tilde{\mathbf{r}} = \mathbf{K}^{-1/2}\mathbf{r}$, $\tilde{\mathbf{s}}_i = \mathbf{K}^{-1/2}\mathbf{s}_i$ and $\tilde{\mathbf{s}}_j = \mathbf{K}^{-1/2}\mathbf{s}_j$, then (16.23) and (16.24) can be re-expressed respectively as follows

$$\delta^{\mathrm{ACE}}(\mathbf{d}, \mathbf{r}) = \frac{\left(\tilde{\mathbf{d}}^T \tilde{\mathbf{r}}\right)^2}{\left(\tilde{\mathbf{d}}^T \tilde{\mathbf{d}}\right)\left(\tilde{\mathbf{r}}^T \tilde{\mathbf{r}}\right)} = \left(\mathbf{u}_{\mathbf{d}}^T \mathbf{u}_{\mathbf{r}}\right)^2 \tag{16.25}$$

which is a square of an inner product of two unit vectors, $\mathbf{u}_{\tilde{\mathbf{d}}} = \frac{\tilde{\mathbf{d}}}{||\tilde{\mathbf{d}}||}$ and $\mathbf{u}_{\tilde{\mathbf{r}}} = \frac{\tilde{\mathbf{r}}}{||\tilde{\mathbf{r}}||}$, and

$$\mathrm{MFD}_{\mathrm{ACE}}(\tilde{\mathbf{s}}_i, \tilde{\mathbf{s}}_j) = \frac{\left|\tilde{\mathbf{s}}_i^T \tilde{\mathbf{s}}_j\right|}{\left(\tilde{\mathbf{s}}_i^T \tilde{\mathbf{s}}_i\right)^{1/2} \left(\tilde{\mathbf{s}}_j^T \tilde{\mathbf{s}}_j\right)^{1/2}} = \frac{\left|\tilde{\mathbf{s}}_i^T \tilde{\mathbf{s}}_j\right|}{||\tilde{\mathbf{s}}_i|| \, ||\tilde{\mathbf{s}}_j||} = \left|\cos\left(\mathrm{SAM}(\mathbf{u}_{\tilde{\mathbf{s}}_i}, \mathbf{u}_{\tilde{\mathbf{s}}_j})\right)\right| \tag{16.26}$$

which also becomes the SAM measure of two unit signature vectors $\mathbf{u}_{\tilde{\mathbf{s}}_i} = \frac{\tilde{\mathbf{s}}_i}{||\tilde{\mathbf{s}}_i||}$ and $\mathbf{u}_{\tilde{\mathbf{s}}_j} = \frac{\tilde{\mathbf{s}}_j}{||\tilde{\mathbf{s}}_j||}$ similar to (16.22) with signature vector lengths being unit length.

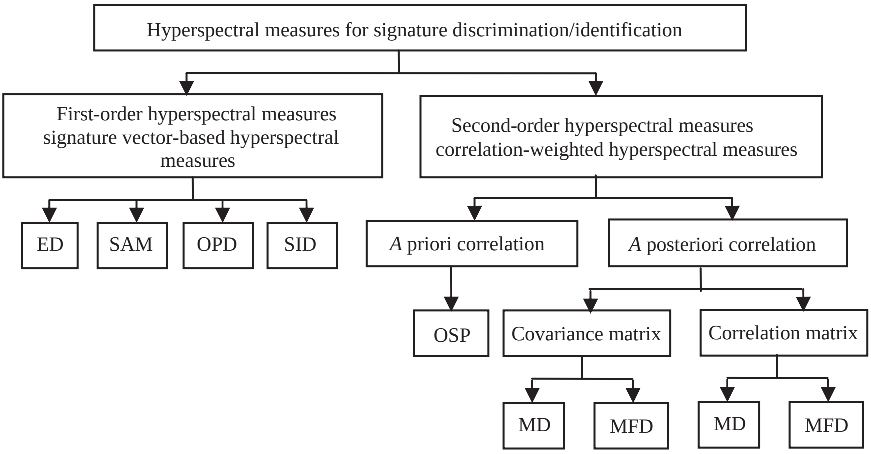

Hyperspectral measures for signature discrimination/identification
First-order hyperspectral measures signature vector-based hyperspectral measures
Second-order hyperspectral measures correlation-weighted hyperspectral measures
ED
SAM
OPD
SID
A priori correlation
A posteriori correlation
OSP
Covariance matrix
Correlation matrix
MD
MFD
MD
MFD

Figure 16.1 Block diagram of various hyperpsectral measures.

In addition to discrimination specified by (16.15) and (16.18) and (16.21), (16.22), (16.24) and (16.26) these equations can be also used to perform identification in the same way that (16.12) and (16.13) do by comparing a real signature vector against a database Δ. However, such identification is rather straightforward because the used *a posteriori* correlation is provided by either sample covariance matrix $\mathbf{K}$ or sample correlation matrix $\mathbf{R}$ compared to the *a priori* correlation which must be determined by the undesired signature matrix $\mathbf{U}$ used for annihilation.

Since all the spectral measures defined by (16.10) and (16.11), (16.15) and (16.18), and (16.20), (16.21), (16.24) involve the calculation of correlation either provided by $P_{\mathbf{U}}^{\perp}$ or the inverse of the sample spectral correlation matrix, $\mathbf{R}^{-1}$ or the inverse of the sample covariance matrix $\mathbf{K}^{-1}$, they can be referred to as second-order hyperspectral measures. On the other hand, the sample spectral correlation is not included in any pure signature vector-based spectral similarity measure. So, SAM and SID can be considered as the first-order spectral measures. Figure 16.1 summarizes relationships among various first-order and second-order hyperspectral measures.

16.4 Experiments

Two data sets are used for experiments, the HYDICE image data in Figure 1.15 and Purdue's Indian Pine test site AVIRIS data in Figure 1.13.

16.4.1 HYDICE Image Experiments

Since the precise knowledge of the 19 R panel pixels is known according to the ground truth provided in Figure 1.15(b), the mean of each of five panel signatures is calculated by averaging the R pixels for each of five rows and shown in Figure 1.16. These five panel signatures were used for discrimination and also as a database for identification. Table 16.1 tabulates identification errors of 19 R panel pixels resulting from pixel-based hyperspectral measures, ED, SAM, OPD, SID, and correlation-weighted hyperspectral measures, MD_{RX}, MD_{CEM}, MFD_{RX}, and MFD_{CEM} where all the four correlation-weighted hyperspectral measures made no errors compared to pixel-based hyperspectral measures that made errors ranging from 4 to 6 with the SID and ED being the best and worst measures.

Table 16.1 Identification errors of 19 R pixels resulting from signature vector-based hyperspectral measures and second-order statistics weighted hyperspectral measures

ED	SAM	OPD	SID
6	5	5	4
$\mathrm{MD}_{\mathrm{RX}}$	$\mathrm{MD}_{\mathrm{CEM}}$	$\mathrm{MFD}_{\mathrm{RX}}$	$\mathrm{MFD}_{\mathrm{CEM}}$
0	0	0	0

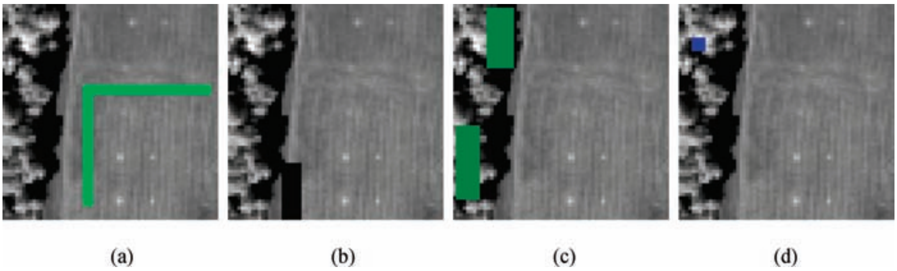


Figure 16.2 (a) Sample grass area; (b) sample road area; (c) sample tree area; (d) sample interference area.

Since the performance of *a posteriori* correlation-weighted hyperspectral measures varies with the knowledge of the **U** used in their measures, their results are not included in Table 16.1. Instead, this issue is investigated separately. To see the impact of various knowledge of **U** on $P_{\mathbf{U}}^{\perp}$-weighted hyperspectral measures, the particular sampling areas specified by the marked areas in Figure 16.2 (a)–(d) were used to obtain undesired signatures for **U**.

Let $\mathbf{u}_5$, $\mathbf{u}_6$, $\mathbf{u}_7$, and $\mathbf{u}_8$ denote grass, road, tree, and interference signatures averaged over these four sample areas, respectively. Table 16.2 tabulates identification errors resulting from the two $P_{\mathbf{U}}^{\perp}$-weighted hyperspectral measures for identification, $\mathrm{ID}_{\mathrm{OSP},\Delta}$ and $\mathrm{ID}_{\mathrm{OSP}}$ where $\mathbf{U}_4$ consists of four undesired panel signatures and $\mathbf{U}_5 = [\mathbf{U}_4\mathbf{u}_5]$, $\mathbf{U}_6 = [\mathbf{U}_4\mathbf{u}_5\mathbf{u}_6]$, $\mathbf{U}_7 = [\mathbf{U}_4\mathbf{u}_5\mathbf{u}_6\mathbf{u}_7]$, $\mathbf{U}_8 = [\mathbf{U}_4\mathbf{u}_5\mathbf{u}_6\mathbf{u}_7\mathbf{u}_8]$.

As shown in Table 16.2, $\mathrm{ID}_{\mathrm{OSP},\Delta}$ performed better than $\mathrm{ID}_{\mathrm{OSP}}$ and both improved their performance if more undesired signatures were eliminated. In particular, $\mathrm{ID}_{\mathrm{OSP},\Delta}$ made no errors once a background signature was eliminated, while $\mathrm{ID}_{\mathrm{OSP}}$ must wait until all four background signatures were eliminated. Nevertheless, both $\mathrm{ID}_{\mathrm{OSP},\Delta}$ and $\mathrm{ID}_{\mathrm{OSP}}$ generally performed better than signature vector-based hyperspectral measures.

16.4.2 AVIRIS Image Experiments

Another image data set to be used in this section is Purdue's Indian Pine test site shown in Figure 1.13 that is an AVIRIS image collected from an area of mixed agriculture and forestry in

Table 16.2 Identification errors resulting from the a posteriori-weighted hyperspectral measures, $\mathrm{ID}_{\mathrm{OSP\text{-}D},\Delta}$ and $\mathrm{ID}_{\mathrm{OSP\text{-}D}}$ with various knowledge provided by **U**

	U_4	U_5	U_6	U_7	U_8
$\mathrm{ID}_{\mathrm{OSP},\Delta}$	5	0	0	0	0
$\mathrm{ID}_{\mathrm{OSP}}$	6	2	2	1	0

Table 16.3 Classification resulting from various signature vector-based hyperspectral measures

Class	Sample number	ED (%)	SAM (%)	SID (%)	OPD (%)
9	20	75	95	100	90
7	26	92.308	92.308	92.308	92.308
1	54	87.037	87.037	87.037	87.037
16	95	94.737	92.632	93.684	92.632
13	212	95.283	93.868	95.283	94.811
4	234	21.795	58.974	61.111	59.402
15	380	27.895	20.789	22.105	22.105
8	489	56.033	77.096	76.892	76.892
5	497	2.6157	3.2193	3.0181	3.2193
12	614	12.378	27.85	24.593	27.85
6	747	28.38	30.79	31.325	30.522
3	834	16.187	27.698	23.022	28.417
10	968	47.004	66.942	59.194	70.351
14	1294	82.071	81.453	85.317	81.607
2	1434	53.208	43.724	48.187	42.19
11	2468	16.207	21.84	22.447	21.677
Average	10366	37.8642	43.2859	43.4015	43.4689

Northwestern Indiana, USA. This image scene has been studied extensively in the literature. It is very interesting in the sense that most pixels in the image scene are heavily mixed and and provides another excellent example for experiments. A detailed study on this scene was recently reported in Liu et al. (2006). Unfortunately, to the author's best knowledge, a comprehensive study on subpixels and mixed pixels in this scene is yet to be done.

Since the number of samples in each of 16 pattern classes varies in a wide range, the performance of correlation-weighted hyperspectral measures also varies. Tables 16.3 and 16.4 tabulate classification rates in percentage (%) of signature vector-based hyperspectral measures and correlation-weighted hyperspectral measures for 16 classes, respectively, where the classes are sorted in an increasing order of the number of samples and the last row calculated the averaged classification rates produced by various measures for each of 16 classes. Comparing Table 16.3 with Table 16.4, it is surprising to discover that the two MD-based hyperspectral measures, two MFD-based hyperspectral measures performed best among all the measures and the signature vector-based hyperspectral measured performed better than OSP-based hyperspectral measures.

Several observations can be made by Tables 16.3 and 16.4 and are worthwhile. The MFD-based hyperspectral measures, $\mathrm{MD}_{\mathrm{RX}}$ and $\mathrm{MD}_{\mathrm{CEM}}$ performed well when the number of samples is small in classes 9, 7, 1, 16, 13, 4, 16. Their performance was deteriorated with increasing samples as opposed to signature vector-based hyperspectral measures which performed increasingly better than did $\mathrm{MFD}_{\mathrm{RX}}$ and $\mathrm{MFD}_{\mathrm{CEM}}$ in classes 10, 14, 2, 11. This was due to the fact that the matching signatures to be used were increasingly affected by contaminated spectral correlation caused by more heavily mixed pixels, in which case, signature vector-based hyperspectral measures were not affected by sample spectral correlation. This was also witnessed by the performance of OSP-based hyperspectyral measures where the signature vectors used for the **U** were heavily mixed. Such mixed pixel information resulted in erroneous elimination of desired pixel information used by $\mathrm{ID}_{\mathrm{OSP},\Delta}$ and $\mathrm{ID}_{\mathrm{OSP}}$. However, it seemed that the performance of OSP-based hyperspectral measures, MD-based and MFD-based hyperspectral measures was little affected by number of samples. They yielded the best performance in general.

Table 16.4 Classification rates resulting from various correlation weighted-based hyperspectral measures

Class	Sample #	$\text{ID}_{\text{OSP},\Delta}$ (%)	ID_{OSP} (%)	MFD_{CEM} (%)	MFD_{RX} (%)	MD_{RX} (%)	MD_{CEM} (%)
9	20	95	75	100	100	100	100
7	26	69.231	73.077	100	100	100	100
1	54	22.222	37.037	98.1	98.148	85.185	85.185
16	95	100	92.632	100	100	89.474	90.526
13	212	97.642	95.755	100	100	99.528	99.528
4	234	0	1.2821	67.9	69.231	91.453	89.316
15	380	20.263	24.474	67.4	68.421	90	90.526
8	489	17.382	29.243	33.9	34.56	98.569	98.773
5	497	64.588	64.185	66.8	66.6	66.6	66.398
12	614	22.15	48.86	64	64.169	79.479	78.827
6	747	62.651	62.651	76.7	76.573	96.118	96.118
3	834	0.47962	5.7554	27.9	28.417	61.151	62.95
10	968	3.8223	21.591	50.7	50.413	79.959	79.649
14	1294	42.89	45.595	49.5	49.614	84.776	84.312
2	1434	2.4407	8.0893	22.6	22.106	76.081	76.011
11	2468	0.040519	12.358	13.4	13.574	57.091	57.131
Average	10366	19.9691	28.3523	41.515	41.607	75.6706	75.6705

In the past, many research efforts published in the literature have studied this image scene with background removed from the image scene for analysis. In the following experiments, we investigate such a scenario to see if the knowledge of background affects the performance of the correlation-weighted hyperspectral measures. Table 16.5 tabulates their classification rates in percentage (%) for 16 classes.

Comparing Table 16.5 with Table 16.4, their performances did not change drastically where both OSP-based hyperspectral measures and MD-based hyperspectral measures improved classification slightly in contrast to the MFD-based hyperspectral measures whose performance was slightly degraded. This was because the former had less interference caused by the mixed pixels in the background, while the latter required background pixels included in the sample correlation/covariance matrix to eliminate the effect incurred by the background.

Finally, Figure 16.3 plots the averaged performance of the four types of measures, sample-based, OSP-based, MD-based, and MFD-based hyperspectral measures in classification by averaging the results in Tables 16.3 and 16.4 with $\text{MD}_{\text{RX}} + \text{MD}_{\text{CEM}} \rightarrow$ MD-based hyperspectral measures, $\text{MFD}_{\text{RX}} + \text{MFD}_{\text{CEM}} \rightarrow$ MFD-based hyperspectral measures, $\text{ID}_{\text{OSP},\Delta}$, $\text{ID}_{\text{OSP}} \rightarrow$ OSP-based hyperspectral measures, $\text{ED} + \text{SAM} + \text{SID} + \text{OPD} \rightarrow$ signature vector-based hyperspectral measures.

It is interesting to find that MD-based hyperspectral measures yielded the best performance. The ability of the MFD-based hyperspectral measures in classification was also reasonably good with performance deteriorated as the number of samples was increased. The OSP-based hyperspectral measures performed better when the sample size was small. On average the performance of signature vector-based hyperspectral measures was the worst.

A concluding comment is noteworthy. On many occasions the correlation-weighted hyperspectral measures are easy to be confused with classifiers when they are applied to real images to perform spectral similarity such as experiments performed above for the HYDICE and Purdue's Indian Pine scene images. First, the correlation-weighted hyperspectral measures are not designed

Table 16.5 Classification rates resulting from various correlation-weighted hyperspectral measures with background removed

Class	Sample number	$\mathrm{ID}_{\mathrm{OSP},\Delta}$ (%)	$\mathrm{ID}_{\mathrm{OSP}}$ (%)	$\mathrm{MFD}_{\mathrm{CEM}}$ (%)	$\mathrm{MFD}_{\mathrm{RX}}$ (%)	$\mathrm{MD}_{\mathrm{RX}}$ (%)	$\mathrm{MD}_{\mathrm{CEM}}$ (%)
9	20	80	75	100	100	100	100
7	26	65.385	73.077	100	100	100	100
1	54	25.926	33.333	96.296	96.296	87.037	88.889
16	95	100	96.842	100	100	89.474	89.474
13	212	97.17	96.226	100	100	100	100
4	234	0	1.2821	66.239	67.094	91.026	89.316
15	380	21.842	31.316	73.421	73.158	87.632	86.579
8	489	12.27	29.448	56.442	55.828	98.773	98.773
5	497	63.38	64.789	70.423	70.423	75.453	74.447
12	614	22.964	45.44	52.769	53.094	82.248	81.922
6	747	66.667	74.565	72.557	72.825	96.118	95.85
3	834	0	6.1151	22.542	22.662	64.149	64.988
10	968	1.5496	18.905	35.021	35.021	80.992	80.785
14	1294	54.096	59.119	49.536	49.227	85.626	85.626
2	1434	0.06974	5.3696	18.55	17.643	77.964	77.964
11	2468	0	11.426	9.1977	9.2382	62.156	61.75
Average	10366	20.847	30.1949	38.5105	38.3851	78.1015	77.8989

for classifiers. Instead, they are designed to discriminate and identify signature vectors. Second, a classifier is a discrete p-value function which maps a data sample to a specific value that indicates the class to which it belongs. So, it is a class membership-labeling process and needs to know the number of classes, p *a priori*. Such prior knowledge is not required by the correlation-weighted hyperspectral measures. Third, a classifier generally requires training samples to provide its needed class information, while the correlation-weighted hyperspectral measures do not. Finally and most importantly, when a classifier operates on data sample vectors in the original data space, it usually implements a distance metric to measure similarity between two data sample vectors. So, when signature vector-based hyperspectral measures can be used for this distance metric, in which case they become classifiers. Classifiers of this type include ISODATA, nearest neighbor rule-based

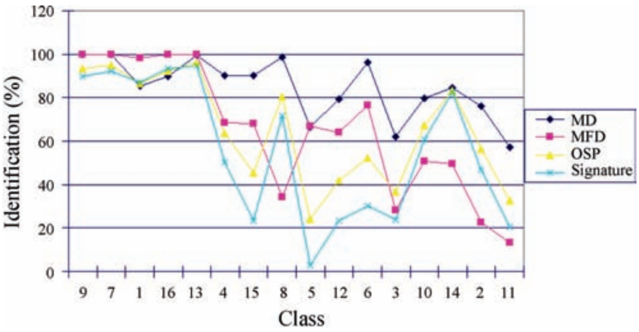


Figure 16.3 Averaged performance of four types of measures, signature vector-based, OSP-based, MD-based, and MFD-based hyperspectral measures.

classifiers. However, a good classifier generally extracts class feature information or takes advantage of training sample vectors to transform the original data space into a feature space in which it can perform classification more effectively on the extracted class features rather than data sample vectors. This is the main reason that classifiers of this type such as FLDA and SVM always perform better than the correlation-weighted hyperspectral measures used as classifiers.

16.5 Conclusions

A simplest unsupervised hyperspectral target analysis is to use a hyperspectral measure to perform target signature discrimination. When there is prior knowledge available, a hyperspectral measure can take advantage of it to be further used to accomplish tasks other than discrimination. For example, if there is a training data set for class membership available, a hyperspectral measure can work as a classifier either as a hard decision-made classifier or a soft decision-made quantifier. Moreover, if there is a database or spectral library available, a hyperspectral measure can be performed to be used for signature verification or identification. This chapter derives two categories of hyperspectral measures, signature vector-based hyperspectral measures and correlation matrix-weighted hyperspectral measures. While the former is generally considered as spectral similarity measures commonly used in remote sensing community, the latter has been used as various forms as detectors, classifiers, or identifiers due to the fact that they can take into account the correlation among data sample vectors to be used for various tasks. Such sample correlation information can be characterized by two types of information, *a priori* information and *a posteriori* information which can be used to design correlation matrix-weighted hyperspectral measures. As examples, if *a priori* information is provided by a set of training data for class membership, correlation matrix-weighted hyperspectral measures work as Mahalanobis classifier/maximum likelihood classifier. On the other hand, if *a posteriori* information is specified by the sample covariance matrix, correlation matrix-weighted hyperspectral measures can be considered as RX detector or matched filter. Interestingly, when the sample covariance/correlation matrix is assumed to be the identity matrix, the sample covariance matrix-weighted hyperspectral measures are reduced to Euclidean distance (ED) and the sample correlation matrix-derived matched filter hyperspectral measures become spectral angle mappers (SAM). However, there are limitations on an extent to which a hyperspectral measure can be stretched out, specifically, when no prior knowledge is available. Chapter 17 is developed to particularly address this issue.

17

Unsupervised Linear Hyperspectral Mixture Analysis

Chapter 16 provides the simplest unsupervised means of using hyperspectral measures to analyze data sample vectors for signature discrimination, classification, and identification without appealing for any algorithm. Consequently, its applications are rather limited. Specifically, when it comes to unsupervised linear spectral mixture analysis (LSMA), hyperspectral measures alone cannot do the same tricks as done in Chapter 16. *Unsupervised* linear spectral mixture analysis (ULSMA) is highly desirable in real-world applications due to the fact that prior knowledge is generally not available. Two of most challenging issues in ULSMA are (1) determining the number of signatures present in the data and (2) finding the signatures needed to perform spectral unmixing, both of which do not occur in *supervised* LSMA (SLSMA) since the latter generally assumes the target signatures to be known *a priori* or provided by prior knowledge. With recent advances in hyperspectral sensor technology many unknown and subtle signal sources that cannot be identified by prior knowledge or visual inspection can now be uncovered and revealed. In this case, using pre-assumed knowledge may not be reliable, accurate, or complete and, thus, the resulting unmixed results may be misleading. This chapter addresses these issues by introducing a new concept of sample spectral statistics generated by interband spectral information (IBSI) among a set of data sample vectors, *S*, denoted by IBSI(*S*), which can be used to categorize signatures into background (BKG) and target classes in terms of their sample spectral statistics. In order to extract these two types of signatures, two approaches, referred to as least squares (LS)-based ULSMA and component analysis-based (CA)-ULSMA, are developed to perform LSMA. It turns out that such unsupervised versions of LSMA can perform better than SLSMA in real-image experiments when obtaining true knowledge becomes difficult and incomplete.

17.1 Introduction

With high spectral/spatial resolution many unknown material substances can be revealed by hyperspectral imaging sensors for data exploitation, specifically LSMA where a set of signatures used to form a linear mixing model may not be known by prior knowledge or be identified visually. Under such circumstances performing SLSMA with assumed target knowledge assumed *a priori* or obtained by visual inspection may not be realistic or applicable to real-world problems. Therefore, it is highly desirable to obtain the desired signature knowledge directly from the data without appealing for prior knowledge. In doing so, two major issues need to be addressed: (1) the number

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

of signatures, denoted by p, used to form a linear mixing model and (2) a set of appropriate p signatures, $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p$, used to unmix data. Both issues are very challenging because determining the value of p and finding a desired set of p signatures $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p$ must be conducted by an unsupervised means.

Since a hyperspectral signature is obtained by hundreds of contiguous spectral channels, the spectral correlation across all the spectral bands is very crucial and useful for material identification. In this chapter, we introduce a new concept of so-called *spectral* targets to differentiate *spatial* targets commonly addressed in traditional image processing. In the traditional image processing there are no spectral bands involved. The targets of interest are generally defined and identified by their spatial properties such as size, shape, and texture. Accordingly, the targets of this type are considered as "*spatial*" targets. The techniques developed to recognize such spatial targets are referred to as spatial domain-based image processing techniques. On the other hand, due to use of spectral bands specified by a range of wavelengths a multispectral or hyperspectral data sample is actually a vector expressed as a column vector, of which a sample in a spectral band is produced by a particular wavelength. As a consequence, a single hyperspectral sample vector already contains abundant spectral information provided by hundreds of contiguous spectral bands that can be used for data exploitation. Such spectral information within a single data sample vector is referred to as interband spectral information (IBSI) in this chapter. By virtue of such IBSI two single data sample vectors can be discriminated, classified, and identified via a spectral similarity measure such as spectral measures presented in Chapter 16. In light of this interpretation a target is called "*spectral target*" if it is analyzed based on its spectral properties characterized by IBSI as opposed to "*spatial target*" analyzed by interpixel spatial information provided by spatial correlation among sample pixels.

More specifically, let $S = \{\mathbf{r}_i\}_{i=1}^N$ be a set of N data sample vectors where $\mathbf{r}_i = (r_{i1}, r_{i2}, \ldots, r_{iL})^T$ is the ith data sample vector in S and L is the total number of spectral bands. The spectral correlation across all the spectral bands within $\mathbf{r}_i$ is defined and referred to as interband spectral information of signature $\mathbf{r}_i$, denoted by IBSI($\mathbf{r}_i$). That is, the IBSI($\mathbf{r}_i$) is provided by spectral correlation among the L spectral values, $\{r_{ij}\}_{j=1}^L$ across spectral bands within the single data sample vector $\mathbf{r}_i$. For example, second-order statistics provided by IBSI($\mathbf{r}_i$) can be auto-correlation of $\mathbf{r}_i$, $\sum_{j=1}^L r_{ij}^2$, or cross correlation of $\mathbf{r}_i$, $\sum_{j=1,k=1,j\neq k}^L r_{ij}r_{ik}$. However, what we are really interested in is the sample statistics provided by a set of data sample vectors, $S = \{\mathbf{r}_i\}_{i=1}^N$, denoted by IBSI($S$), specifically, second-order statistics of IBSI(S) such as sample auto-correlation matrix of S, $\sum_{i=1}^N \mathbf{r}_i\mathbf{r}_i^T$ and sample cross-correlation matrix of S, $\sum_{i=1,j=1,i\neq j}^N \mathbf{r}_i\mathbf{r}_i^T$. It should be noted that the IBSI(S) is independent of intersample spatial correlation because IBSI(S) remains the same even the samples in S are reshuffled. This type of spectral information is opposite to sample spatial statistics commonly used in traditional image processing that takes into account spatial locations of a set of data sample vectors where reshuffling data sample vectors in a sample spatial correlation matrix can result in another different sample spatial correlation matrix because it alters spatial correlation when data sample vectors are re-arranged in different spatial coordinates.

One of major strengths resulting from a hyperspectral imaging sensor is its ability in uncovering and revealing subtle material substances that cannot be resolved by multispectral imager. Such target signal sources are very critical and vital to hyperspectral image analysts. Unfortunately, their presence is also limited to their sample size and spatial extent. One effective means is to take advantage of IBSI to calculate the sample statistics provided by these samples, denoted by a set S^{target}, in terms of IBSI(S^{target}). In other words, the sample size of S^{target} is generally very small compared to a large sample pool of background (BKG) signatures, denoted by S^{BKG}. As a consequence, the IBSI(S^{target}), can be better characterized by high order of statistics (HOS), while the IBSI(S^{BKG}) constitutes most of second-order statistics. In this case, we can define a background signature as a signature that is of no interest in applications and is usually characterized by second-order sample statistics provided by IBSI(S^{BKG}), and a target signature as a desired signature that is of major interest and can be mainly specified by high-order sample statistics provided by IBSI(S^{target}). Of course, when a signature exhibits both characteristics of second-order statistics and high-order statistics in terms of IBSI, it will be considered as a target signature. In hyperspectral image analysis this assertion seems reasonable because the spectral targets of interest in hyperspectral data exploitation are generally those that either occur with low probability or have small populations when they are present. In particular, these types of spectral targets are usually relatively small, appear in small population, and also occur with low probabilities, for example, special spices in agriculture and ecology, toxic wastes in environmental monitoring, rare minerals in geology, drug/smuggler trafficking in law enforcement, combat vehicles in the battlefield, man-made objects and anomalies in intelligence gathering, landmines in war zones, chemical/biological agents in bioterrorism, weapon concealment, and mass graves. These spectral targets are generally considered as insignificant objects in terms of IBSI(S) because of their very limited spatial information provided by a small sample pool S, but they are actually critical and crucial for defense applications and are insignificant compared to targets with large sample pools and generally hard to be identified by visual inspection. From a statistical point of view, the spectral information statistics of such special targets cannot be captured by second-order statistics of IBSI(S) but rather by HOS of IBSI(S).

Once hyperspectral signatures are categorized into background signatures and target signatures according to their sample spectral statistics characterized by IBSI, the next follow-up task is to design and develop algorithms to extract signatures from both categories that can be used to form a linear mixture model for the SLSMA to unmix target signatures in S^{target}, whereas the background signatures in S^{BKG} will be used for BKG suppression so as to enhance target detectability and discriminability. Two remaining issues needed to be resolved are (1) how to determine the numbers of signatures in the BKG as well as target classes and (2) how to find these two categories of signatures, BKG as well as target signatures. While the first issue can be addressed by the concept of virtual dimensionality (VD) recently developed in Chapter 5, the second issue is the major focus to be addressed in this chapter where two approaches, least squares-based ULMSA (LS-ULSMA) and component analysis-based ULSMA (CA-ULSMA), are developed to find a set of so-called virtual signatures (VSs) according to IBSI(S)-defined BKG and target signatures. The term of VS introduced here intends to distinguish it from the commonly used term of “endmember” that is assumed to be a pure signature that may not be a real data sample vector and also from the term of virtual endmembers used in Tompkins et al. (1997) and Bowles and Gilles (2007) in endmember extraction.

From the above IBSI(S)-defined BKG class the signatures in the BKG class are most likely characterized by second-order statistics of IBSI(S) compared to signatures of interest in the target class which will be more likely to be captured by HOS of IBSI(S) as outliners due to their small spatial presence. In this case, high-order spectral targets are assumed to be desired targets for image analysis, while second-order spectral targets are considered as undesired targets for which we would like to annihilate or suppress prior to data processing so as to improve image analysis. So, an unsupervised LS-based algorithm designed on second-order spectral statistics can only be used to extract spectral targets of second-order statistics of IBSI(S^{BKG}). In order for an LS-based algorithm to be able to extract HOS of IBSI(S^{target}), we sphere the data by removing the first- and second-order spectral statistics information from the original data so that the sphered data consist of only those data sample vectors characterized by high-order statistics, which contain desired targets. So, if an unsupervised LS-based algorithm operates on two data sets, original data and its sphered data, it can extract both second-order and high-order spectral targets. As a result, an LS-based algorithm can accomplish two goals, finding BKG VSs from the original data space and in the mean time it can also extract target VSs from the sphered data space. The LSMA makes use of these two sets of VSs, BKG and target VSs, to form a linear mixing model to perform spectral unmixing is referred to as LS-ULSMA.

In a parallel development to LS-ULSMA, an alternative approach is to develop component analysis (CA)-based techniques that statistically de-correlate the data into a set of spectral components so that various levels of target information can be captured and characterized in individual and separate spectral components. Unlike LS-ULSMA that operates the same LS-based algorithm on two data cubes, that is, the original data and sphered data, the CA-based approach operates different component analysis transforms on the same data cube to capture targets characterized by any order of statistics specified by IBSI(S). It is known that principal components analysis (PCA) is a second-order statistics-based transform, which uses an eigenmatrix made up of all eigenvectors to produce a set of ranked principal components (PCs) in accordance with the magnitude of data variances represented by eigenvalues. Since BKG signatures usually have a large population, which generally contributes data variances in spectral statistics in terms of IBSI(S^{BKG}), it is expected that the first few PCs should retain most BKG signatures of the data. Target signatures, in contrast, usually have a small sample pool size, which contributes very little to second order of spectral statistics. Consequently, these target signatures can be rather characterized by high orders of spectral statistics, IBSI(S^{target}). In order to capture these types of target signatures, a preprocessing is required to remove background signatures prior to extraction of target signatures. The independent component analysis (ICA) seems to be a perfect candidate for this task because ICA has been widely studied in hyperspectral imaging community. It performs data sphering to remove the first two orders of spectral statistics in terms of IBSI(S^{BKG}) and then produces a set of statistically independent components (ICs) for signal source separation. By means of ICA target signatures characterized by IBSI(S^{target}) can be extracted in separate and individual ICs. The only two issues that remain to be resolved are (1) how to determine the numbers of PCs and ICs and (2) how to extract BKG signatures from PCs and target signatures from ICs. The concept of VD once again provides a feasible solution to the first issue. As for the second issue it can be solved by finding background signal sources corresponding to the maximal projections of each of PCs and target signal sources corresponding to maximal and minimal projections of each of ICs. To meet this need a CA-based unsupervised virtual signature finding algorithm (CA-UVSFA) is particularly developed to allow users to find both BKG VSs and target VSs that can be used to form a linear mixing model for SLSMA to unmix data. Such an SLSMA that makes use of the signatures found by the UVSFA as signature knowledge to perform linear spectral unmixing is referred to as CA-based ULSMA.

In order to substantiate the two developed techniques to perform ULSMA, LS-ULSMA, and CA-ULSMA, synthetic image experiments are first used to conduct a quantitative study between SLSMA and LS-ULSMA/CA-ULSMA. It is then followed by real data experiments to evaluate performance analysis in comparison with SLSMA for which the signature knowledge is provided *a priori* either by ground truth or visual inspection. As demonstrated by experimental results, when SLSMA uses accurate signature knowledge SLSMA would perform better than LS-ULSMA/CA-ULSMA. Otherwise, LS-ULSMA/CA-ULSMA is a better option than SLSMA.

17.2 Least Squares-Based ULSMA

An LS-based approach designs an LS-based algorithm that can be first applied to the original data to extract data sample vectors characterized by second-order statistics of IBSI(S) as BKG signatures and then is applied again to the sphered data to capture data sample vectors characterized by HOS of IBSI(S) as target signatures. The task of data sphering is designed to remove the data sample mean and co-variances while making data variances unity so that data sample vectors completely characterized by second-order statistics of IBSI(S) will be forced on the sphere and all other data sample vectors that are characterized by HOS of IBSI(S) are either inside (sub-Gaussian samples) or outside the sphere (super-Gaussian samples). As a consequence, data sample vectors characterized by IBSI(S) of orders higher than 2 can be extracted from inside or outside the sphere. Interestingly, the idea of using the same algorithm applied to different data sets resulting from the same data set to be processed has never been explored until Chang et al. (2010, 2011).

In what follows, three least squares (LS)-based algorithms developed for SQ-EEAs in Chapter 8 can be used for the purpose of finding VSs directly from the data. The first algorithm is ATGP that is an orthogonal subspace projection (OSP)-based algorithm. Since the OSP is a least squares-based criterion, the ATGP can be also viewed as an unsupervised version of an unconstrained LS-based LSMA method. A second LS-based algorithm is an unsupervised version of a partially abundance-constrained least squares NCLS, unsupervised NCLS (UNCLS) as opposed to a third LS-based algorithm that is an unsupervised version of fully abundance least squares-based FCLS, unsupervised FCLS (UFCLS). When these three unsupervised LS-based algorithms are implemented, a prescribed error ε determined by various applications is required to terminate the algorithms. In general, it is done by visual inspection on a trial-and-error basis which is not practical for our purpose. Therefore, instead of using ε as a stopping rule, we use VD as an alternative stopping rule to determine how many targets are needed to be generated. This is because VD is generally found by methods regardless of applications, which do not appeal for any algorithm, such as the Harsanyi–Farrand–Chang (HFC) method, SSE/HySime in Chapter 5.

In order for the proposed LS-based algorithms to be successful, we also assume that the most BKG data sample vectors are characterized by a large number of uninteresting data sample vectors in the sample pool *S* that can be characterized by second-order statistics of IBSI(S) as opposed to target sample vectors that can be captured by higher order statistics of IBSI(S) due to a small number of sample vectors in *S*. As a result of this assumption two sets of data sample vectors can be derived from the original data. One set is the original data and the other set is the sphered data that has the mean and covariance removed from the original data for data processing. We then apply the three unsupervised LS-based algorithms to these two data sets to extract second-order BKG data sample vectors as well as high-order target data sample vectors. However, if a data sample vector shows strong signal statistics in both original and sphered data sets, it will be considered as a target sample vector and can be removed from the BKG class.

A detailed implementation of LS-based unsupervised VS finding algorithm (LS-UVSFA) can be briefly described below where the LS-based unsupervised algorithm used in LS-UVSFA can be one of the three LS unsupervised algorithms, ATGP, UNCLS, and UFCLS described above.

LS-based Unsupervised VS Finding Algorithm (LS-UVSFA)

1. Apply VD on the image data to determine the number of spectrally distinct signatures, n_{VD} required for an LS-based algorithm to generate.
2. Apply an LS-based algorithm to the original image data and find n_{VD} data sample vectors in the BKG class, $\text{S}^{\text{BKG}} = \left\{\mathbf{b}_j^{\text{LS}}\right\}_{j=1}^{n_{\text{VD}}}$.
3. Apply the same LS-based algorithm to the sphered data and find n_{VD} target sample vectors in the target class, $\text{S}^{\text{target}} = \left\{\mathbf{t}_j^{\text{LS}}\right\}_{j=1}^{n_{\text{VD}}}$.
4. Since there may be some sample vectors in S^{BKG} whose spectra are very close to those that also appear in S^{target}, a spectral measure such as SAM is applied to extract these sample vectors that will be removed from S^{BKG}. Let the resulting BKG class be denoted by $\tilde{S}^{\text{BKG}} = \left\{\tilde{\mathbf{b}}_i^{\text{LS}}\right\}_{i=1}^{n_{\text{BKG}}}$ where n_{BKG} is the total number of remaining BKG sample vectors in $\tilde{\text{S}}^{\text{BKG}}$ after the common sample vectors in $\text{S}^{\text{target}} \cap \text{S}^{\text{BKG}}$ are removed.

5. Form a set of VSs, S^{VS} by merging $\tilde{S}^{\mathrm{BKG}}$ and $\mathrm{S}^{\mathrm{target}}$, that is, grouping all the sample vectors in $\left\{\tilde{\mathbf{b}}_i^{\mathrm{LS}}\right\}_{i=1}^{n_{\mathrm{BKG}}} \cup \left\{\mathbf{t}_j^{\mathrm{LS}}\right\}_{j=1}^{n_{\mathrm{VD}}}$. It should be noted that the number of pixels in S^{VS} is between n_{VD} and $2n_{\mathrm{VD}}$, that is, $n_{\mathrm{VD}} \leq n_{\mathrm{VD}} + n_{\mathrm{BKG}} \leq 2n_{\mathrm{VD}}$.
6. Apply an SLSMA method such as abundance-unconstrained classifier LSOSP, abundance nonnegativity constrained classifier NCLS and abundance fully constrained classifier FCLS to perform spectral unmixing where only the target sample vectors in $\mathrm{S}^{\mathrm{target}}$ will be unmixed by their corresponding abundance fractions while the sample vectors in $\tilde{S}^{\mathrm{BKG}} = \left\{\tilde{\mathbf{b}}_i^{\mathrm{LS}}\right\}_{i=1}^{n_{\mathrm{BKG}}}$ will be used for BKG suppression.

It should be noted that when a specific LS-based algorithm is used, the superscript "LS" in the above algorithm will be replaced with this particular algorithm. For example, if ATGP is used for LS-UVSFA, it is then called ATGP-UVSFA.

17.3 Component Analysis-Based ULSMA

As noted in the introduction, hyperspectral signatures can be categorized into background signatures characterized by second-order statistics of IBSI(S) and target signatures characterized by HOS of IBSI(S). Recall that the commonly used PCA is a second-order statistics-based transform that uses a set of PCs to represent the data where eigenvectors are projection vectors to specify PCs with eigenvalues being data variances. In this case, PCA can be then used to extract background signatures characterized by second-order statistics of IBSI(S) in PCs. On the other hand, ICA is an HOS-based transform that uses mutual information to generate a set of ICs to represent data. Therefore, ICA can be used to find desired target signatures characterized by HOS of IBSI(S) in ICs. In both cases, VD is again used to determine how many PCs and ICs are required to extract signatures. Since PCs and ICs are obtained by mapping all data samples onto the projection vectors, the projection values of data samples are real values. So, an issue arises: how many data sample vectors should be selected from each PC and each IC? Two sample values in each IC are of major interest: one with maximal projection value and the other with minimal projection value. These two samples represent maximal projections in two opposite directions of a projection vector that specifies an IC. They both indicate their importance in data analysis. This idea was previously explored in pixel purity index (PPI) in Chapter 7 where the most likely endmembers are those samples with either maximal or minimal projections on each randomly generated vectors referred to as skewers. However, it is worth noting that there is no similar selection of sample vectors with minimal projections in PCs due to the fact that PCA is a transformation of second-order statistics with variance representing signal energy. Those samples with minimal projections, that is, variances are supposed to correspond to noisy samples. In this case, there is no reason to select data sample vectors with minimal projections in PCs as desired samples similar to the case described above for ICA where noisy samples with small variances have been removed by sphering. All such data sample vectors extracted from PCs and ICs are considered as VSs.

Using VD again to determine the numbers of PCs and ICs along with data samples selected from the first n_{VD} PCs and data sample vectors with maximal and minimal projections in the first n_{VD} ICs a CA-based unsupervised VS finding algorithm (CA-UVSFA) can be described as follows.

CA-UVSFA

1. Use VD to determine the number of components required to generate, denoted by $n_{\mathrm{VD}} = p$.
2. Apply PCA to the original image data and find p PCs and extract the brightest pixels (i.e., data sample vectors with maximal values in PCs) as a VS from each of p PCs to form $\mathrm{S}^{\mathrm{BKG}} = \left\{\mathbf{b}_j^{\mathrm{PCA}}\right\}_{j=1}^{p}$.

3. Apply ICA to the sphered data and find p ICs and extract points with the maximal and minimal projections from each of p ICs as VSs to form $\mathrm{S}^{\text{target}} = \left\{\mathbf{t}_j^{\text{ICA}}\right\}_{j=1}^{2p}$. Since some sample vectors with minimal projections may be close to certain sample vectors with maximal projections in other ICs, a spectral measure such as SAM is applied to identify these sample vectors and eliminate them from $\mathrm{S}^{\text{target}}$ to form $\tilde{S}^{\text{target}} = \left\{\tilde{\mathbf{t}}_j^{\text{ICA}}\right\}$. It should be noted that the ICA used here is the one developed in Hyvarinen and Oja (1997), called FastICA, that is actually based on a criterion derived from a combination of third- and fourth-order statistics; see Equation (5.35), p. 115 (Hyvarinen, 2001).
4. Furthermore, there may be some target sample vectors extracted in $\tilde{S}^{\text{target}} = \left\{\tilde{\mathbf{t}}_j^{\text{ICA}}\right\}$ that also exhibit strong energies in the PCs where pixels corresponding to these sample vectors may be also extracted in S^{BKG}. In this case, SAM is also used to extract these sample vectors and remove them from S^{BKG}. Let the remaining background sample set be denoted by $\tilde{\mathrm{S}}^{\text{BKG}} = \left\{\tilde{\mathbf{b}}_j^{\text{PCA}}\right\}$.
5. Construct a set of VSs, $\mathrm{S}^{\text{VS}} = \left\{\tilde{\mathbf{b}}_j^{\text{PCA}}\right\} \cup \left\{\tilde{\mathbf{t}}_j^{\text{ICA}}\right\}$ by merging $\tilde{\mathrm{S}}^{\text{BKG}}$ and $\tilde{\mathrm{S}}^{\text{target}}$ for spectral unmixing. It should be noted that the number of VSs in S^{VS} is between p and $3p$.

A comment on that each of PCA and ICA is required to generate p components is noteworthy. It is often the case that target sample vectors may also show up in either PCs or ICs, but not both. In order to make sure that no matter which scenario will be, using p PCs and p ICs should have sufficient components to capture all these target sample vectors. A detailed step-by-step procedure of CA-ULSMA can be summarized as follows.

CA-ULSMA

1. Use HFC/NWHFC method to determine VD and let $n_{VD} = p$
2. Implement CA-UVSFA to produce a set of VSs, S^{VS}.
3. Apply an SLSMA technique such as unconstraint classifier LSOSP, non-negativity constrained classifier NCLS, and fully constrained classifier FCLS to perform spectral unmixing where only the target pixels in $\tilde{\mathrm{S}}^{\text{target}}$ will be unmixed, while the target pixels in $\tilde{\mathrm{S}}^{\text{BKG}}$ will be used for background suppression.

In Step 1 of CA-ULSMA, in order for the HFC/NWHFC method to work effectively, the spectrally distinct signatures defined by VD are those that do not have significant contribution to data variances. Such signatures are generally characterized by three unique features. Firstly, the probabilities of their occurrence are usually low. Secondly, when such signatures are present, there are not too many samples. Thirdly, as a result, the variances of such signatures are generally very small and can be considered to be negligible. So, when the HFC/NWHFC method is used to estimate VD that determines the number of VSs used for LSMA, two assumptions are made on the VSs. The first and foremost assumption is that all the VSs in S^{VS} are assumed to have the least intersample spectral correlation. This is a reasonable assumption since different VSs should have the least spectral correlation among all the data sample vectors and should also be as distinct as possible in terms of spectral characteristics via IBSI(S). Another is that the number of samples specified by VS should be relatively small since they represent most spectrally distinctive signatures. With these two assumptions in mind a VS is only contributed to the sample mean but not variance of each spectral band. This is also the key idea used to develop the HFC/NWHFC method and explains why the HFC/NWHFC-estimated VD works very effectively for HYDICE data in the following experiments where the number of target panel pixels specified by each of five panel signatures is very small. When these two assumptions are violated, the HFC/NWHFC-estimated VD may not be accurate. So, when the HFC/NWHFC method is used to estimate VD we should be aware of these assumptions. Finally, we conclude a noteworthy comment on the use of twice VD value, $2n_{\mathrm{VD}} = 2p$ for LS-ULSMA and CA-ULSMA to extract VSs. Such selection is not arbitrary. It is actually based on the concept, called dynamic dimensionality allocation (DDA) derived in Chapter 22. For more details we refer readers to this chapter.

17.4 Synthetic Image Experiments

Two of six synthetic image-based scenarios in Chapter 4, TI2 in Section 4.3.2.2 and TE2 in Section 4.3.3.2, are particularly selected for experiments and re-described as follows.

Target Implantation 2

The target implantation 2 (TI2) inserts a number of panel pixels into the image by replacing their corresponding BKG pixels. So, the resulting synthetic image has clean panel pixels with perfect knowledge implanted in a noisy BKG corrupted an additive Gaussian noise with a certain level of SNR. TI2 is primarily designed to simulate scenarios with pure pixels implanted as pure signatures to represent true endmembers to evaluate the quantitative performance of SLSMA.

Target Embededness 2

As opposed to TI, the second type of target insertion is target embededness 2 (TE2) that is the same as the TI2 described above except the way the panel pixels are inserted. The BKG pixels were not removed to accommodate the inserted panel pixels as they are done in TI2, but were rather superimposed over the inserted panel pixels. So, in this case, the resulting synthetic image has clean panel pixels embedded in a noisy BKG. The TE2 is particularly designed to simulate scenarios where there are no pure pixels present in the data. As a result, no real true endmembers can be used for LSMA. So, TE2 is more realistic than TI2. Instead, the VSs must be found from TE2. Nevertheless, the complete knowledge of inserted panels and BKG signature is still available for quantitative study and analysis. TE2 is specifically designed to demonstrate the ability of ULSMA in finding VSs.

The synthetic images to be used to simulate TI2 and TE2 for experiments has a size of 200 × 200 pixel vectors with 25 panels of various sizes that are arranged in a 5 × 5 matrix and located at the center of the scene shown in Figure 17.1 where there are a total of 130 panel pixels present in the scene, 80 pure panel pixels in the first column and 20 pure panel pixels in the second column simulated by the five mineral signatures, A, B, C, K, and M in Figure 1.12(c), 20 mixed panel pixels in the third column simulated by 50% of one of five mineral signatures plus 50% of the other four signatures, five 50%-abundaunce subpanel pixels in the fourth column simulated by 50% of one of five mineral signatures plus BKG signature, and five 25%-abundaunce subpanel pixels in the fifth column simulated by 25% one of five mineral signatures plus 75% of the BKG signature.

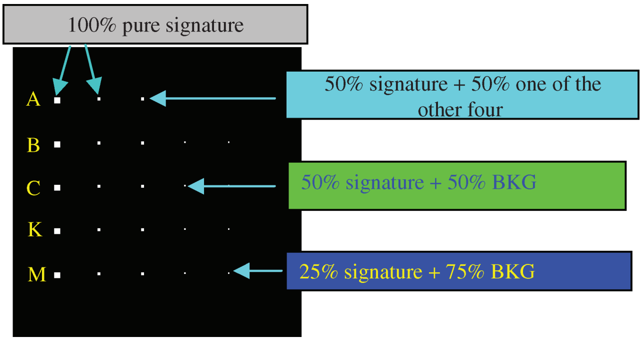


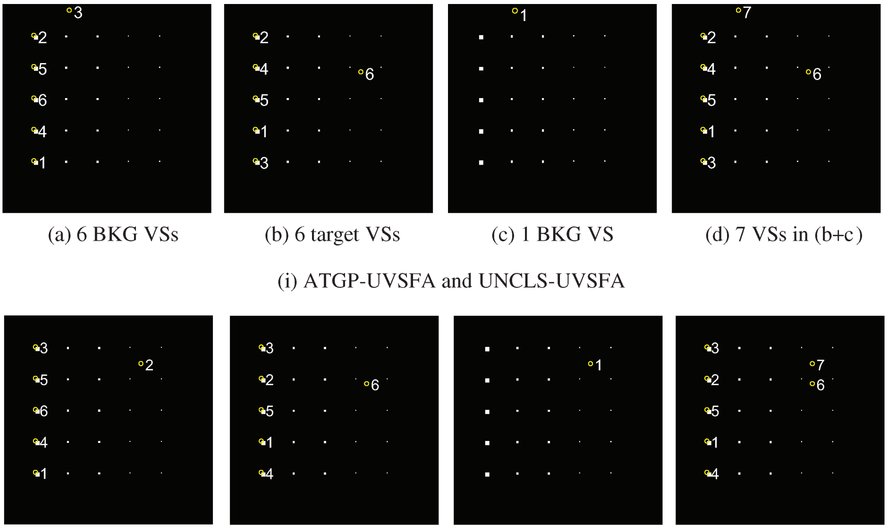
Figure 17.1 A set of 25 panels simulated by A, B, C, K, and M in Figure 1.12(c).

The image BKG in Figure 17.1 is simulated by the sample mean signature in Figure 1.12(a) corrupted by an additive Gaussian noise to achieve a certain signal-to-noise ratio (SNR) that was defined as 50% signature (i.e., reflectance/radiance) divided by the standard deviation of the noise in Harsanyi and Chang (1994).

17.4.1 LS-ULSMA

First of all, assume that no prior knowledge about the scenarios of TI2 and TE2 was provided. In both scenarios the VD-estimated value, n_{VD} was 6 as long as the false alarm probability $P_{\text{F}} \leq 10^{-1}$. Therefore, $n_{\text{VD}} = 6$ was used for the value of p throughout the experiments. Figures 17.2(a)–(d) and 17.3(a)–(d) show the VSs found by the three LS-based methods, referred to as ATGP-UVSFA, UNCLS-UVSFA, and UFCLS-UVSFA with $n_{\text{VD}} = 6$ for the TI2 and TE2 scenarios, respectively, where (a) the second-order BKG VSs were obtained by applying an unsupervised LS-based algorithm to the original data; (b) high-order target VSs were obtained by applying the same algorithm to the sphered data; (c) the remaining BKG VSs in (a) were obtained after removing those BKG VSs that were also identified as target VSs in (b); (d) total desired VSs obtained by merging the VSs in (b) and (c).

According to the results obtained for the TI2 and TE2 scenarios in Figures 17.2 and 17.3, the VSs found by an LS-based UVSFA can be categorized into $\left\{\tilde{\mathbf{b}}_i^{\text{LS}}\right\}_{i=1}^{n_{\text{BKG}}}$ as BKG VSs and $\left\{\mathbf{t}_j^{\text{LS}}\right\}_{j=1}^{n_{\text{VD}}}$ as target VSs. Then the BKG and target VSs were used to form a desired VS matrix $\mathbf{M} = \left[\mathbf{t}_1^{\text{LS}}\mathbf{t}_2^{\text{LS}} \cdots \mathbf{t}_{n_{\text{VD}}}^{\text{LS}}\tilde{\mathbf{b}}_1^{\text{LS}}\tilde{\mathbf{b}}_2^{\text{LS}} \cdots \tilde{\mathbf{b}}_{n_{\text{BKG}}}^{\text{LS}}\right]$ to unmix any given image pixel vector $\mathbf{r}$. Figures 17.4 and 17.5 show the unmixed results of the entire image with each of image pixel vectors unmixed by a set of UVSFA-found n_{VD} VSs,$\left\{\mathbf{t}_j^{\text{LS}}\right\}_{j=1}^{n_{\text{VD}}}$ into a set of n_{VD} abundance fraction maps to represent n_{VD}



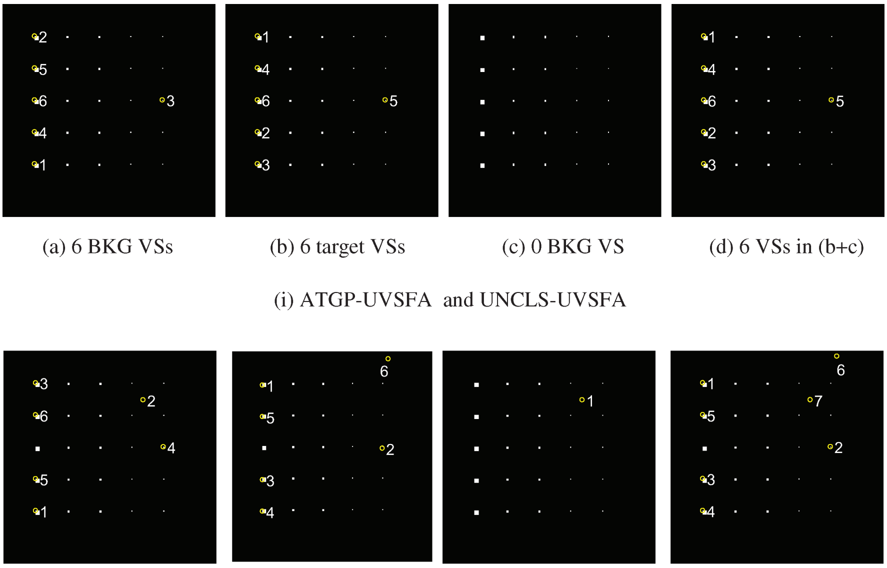
(a) 6 BKG VSs (b) 6 target VSs (c) 1 BKG VS (d) 7 VSs in (b+c)

(i) ATGP-UVSFA and UNCLS-UVSFA

(a) 6 BKG VSs (b) 6 target VSs (c) 1 BKG VS (d) 7 VSs in (b+c)

(ii) UFCLS-UVSFA

Figure 17.2 Target VSs extracted by three unsupervised algorithms ATGP-UVSFA, UNCLS-UVSFA, and UFCLS-UVSFA for scenario TI2.



(a) 6 BKG VSs (b) 6 target VSs (c) 0 BKG VS (d) 6 VSs in (b+c)

(i) ATGP-UVSFA and UNCLS-UVSFA

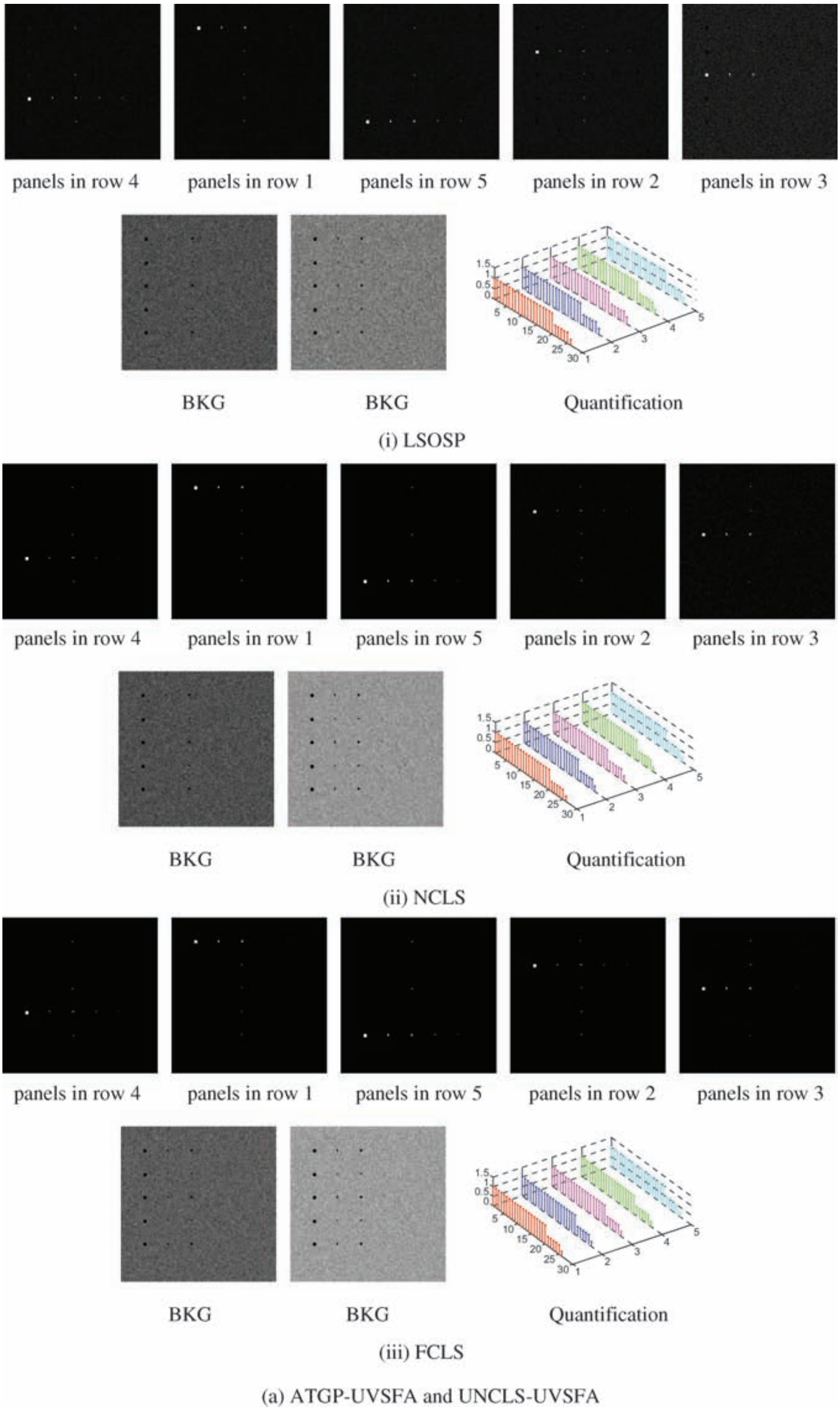
(a) 6 BKG VSs (b) 6 target VSs (c) 1 BKG VS (d) 7 VSs in (b+c)

(ii) UFCLS-UVSFA

Figure 17.3 Target VSs extracted by three unsupervised algorithms ATGP-UVSFA, UNCLS-UVSFA, and UFCLS-UVSFA for scenario TE2.

spectral classes for the TI2 and TE2 scenarios, respectively. The LS-based algorithms used by the UVSFA were ATGP, UNCLS and UFCLS for finding VSs. The mixed pixel classification was performed by three linear spectral unmixing methods, least-squares orthogonal subspace projection (LSOSP), non-negativity constrained least squares (NCLS), and fully constrained least squares (FCLS). The results in (i), (ii), and (iii) of Figures 17.4 and 17.5 were obtained by using LSOSP, NCLS, and FCLS to unmix data samples in TI2 and TE2 scenarios via the VS matrix **M** formed by the target VSs found in Figures 17.2(d) and 17.3(d), respectively, where VSs were identified by the ground truth along with their quantification results for comparison. It should be noted that each figure is arranged in the order of VSs extracted by the UVSFA in Figures 17.2(d) and 17.3(d). Since the whole process was unsupervised the data sample vectors were unmixed by using all VSs including the BKG VSs. Figure 17.5 shows the unmixed results of the scenario TE2 using the target VSs found in Figure 17.3(d). Due to the use of a subpanel pixel $\mathbf{p}_{5,1}^{3}$ in Figure 17.3(d) as one of the target VSs to unmix the TE2 scenario the resulting abundance fractions for the two subpanel pixels were 100% in Figure 17.5(b). These results were different from Figure 17.5(a) where the target and BKG VSs found by ATGP and UNCLS were used for spectral unmixing and the two subpanel pixels were unmixed into their correct abundance fractions of 50% and 25% of Calcite, respectively. Because LSOSP is unconstrained, the 20 pure panel pixels in row 3 were overestimated in Figure 17.5(b)–(i) by using the subpanel pixel $\mathbf{p}_{5,1}^{3}$ as a VS for spectral unmixing.

In order to further investigate the above finding, experiments were conducted by using the prior knowledge of the five mineral signatures in Figure 1.12(c) plus the sample mean signature of Figure 1.12(a) to form the signature matrix **M** for SLSMA to perform spectral unmixing of TI2 and TE2 scenarios using the same three LSMA methods. Figures 17.6(a)–(c) and 17.7(a)–(c) show



panels in row 4 panels in row 1 panels in row 5 panels in row 2 panels in row 3

BKG BKG Quantification

(i) LSOSP

panels in row 4 panels in row 1 panels in row 5 panels in row 2 panels in row 3

BKG BKG Quantification

(ii) NCLS

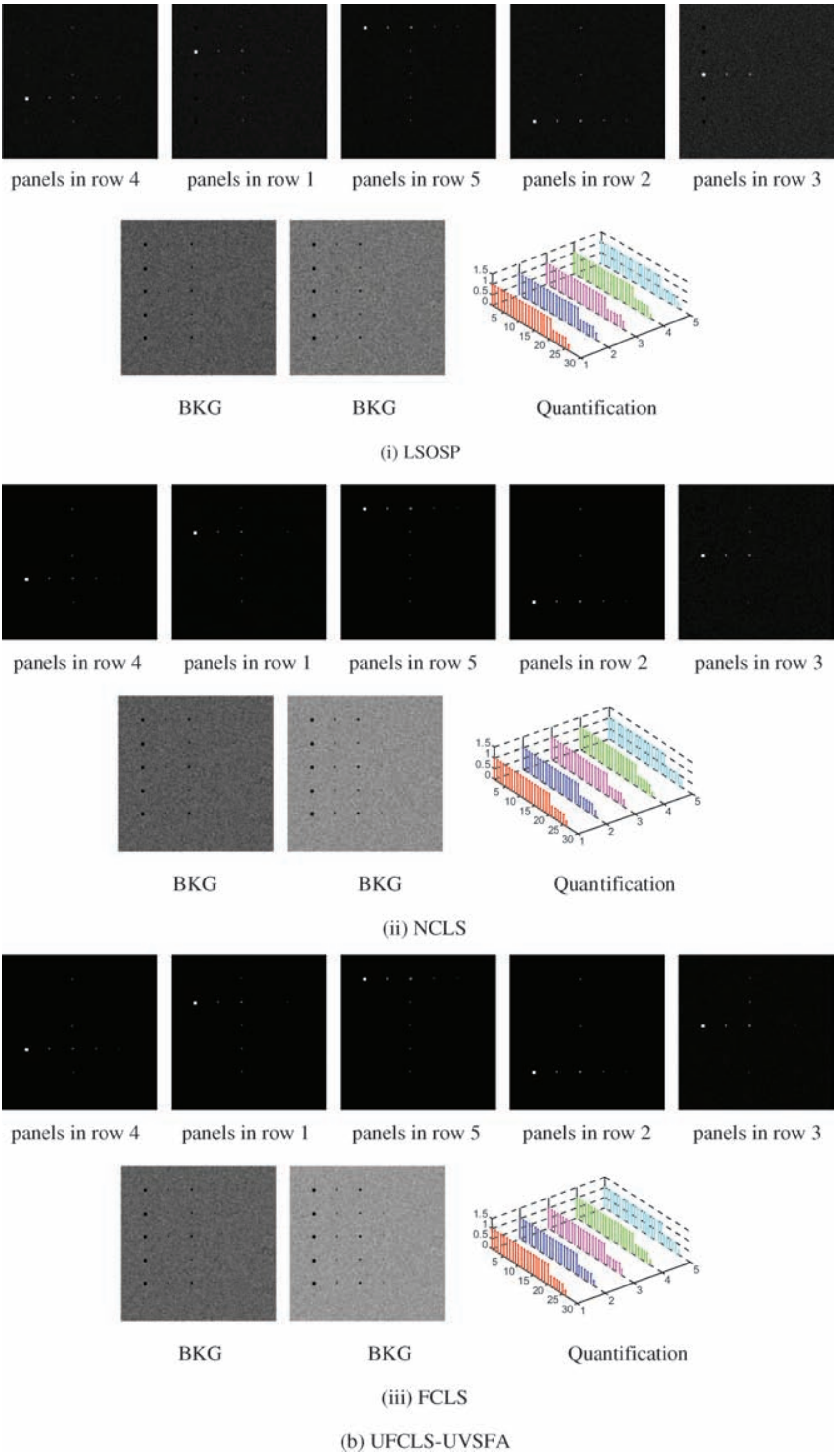
panels in row 4 panels in row 1 panels in row 5 panels in row 2 panels in row 3

BKG BKG Quantification

(iii) FCLS

(a) ATGP-UVSFA and UNCLS-UVSFA

Figure 17.4 Results of using LSOSP, NCLS, and FCLS to unmix TI2 via the target/VSs found in Figure 4(d).



panels in row 4 panels in row 1 panels in row 5 panels in row 2 panels in row 3

BKG BKG Quantification

(i) LSOSP

panels in row 4 panels in row 1 panels in row 5 panels in row 2 panels in row 3

BKG BKG Quantification

(ii) NCLS

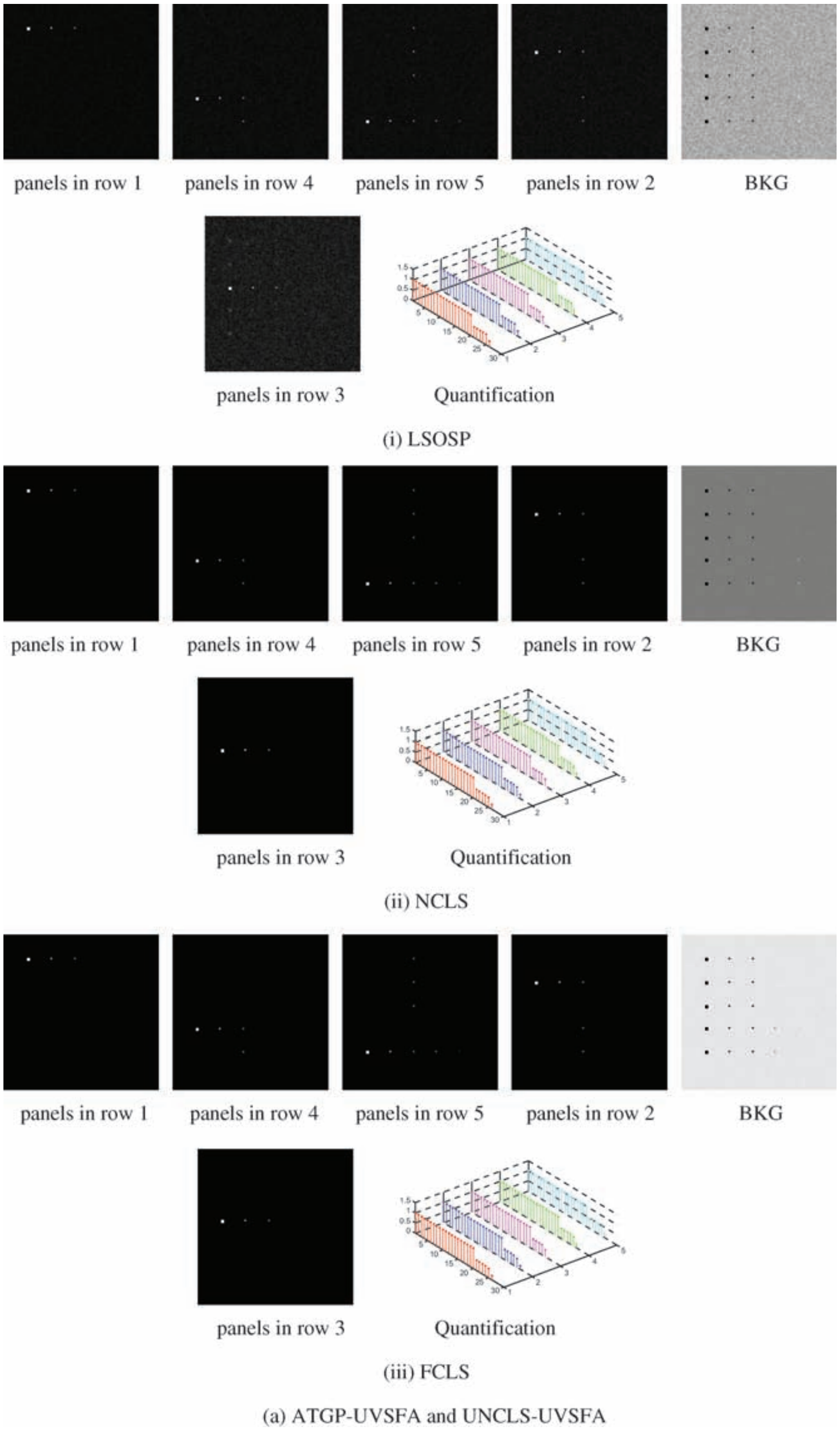
panels in row 4 panels in row 1 panels in row 5 panels in row 2 panels in row 3

BKG BKG Quantification

(iii) FCLS

(b) UFCLS-UVSFA

Figure 17.4 (*Continued*)



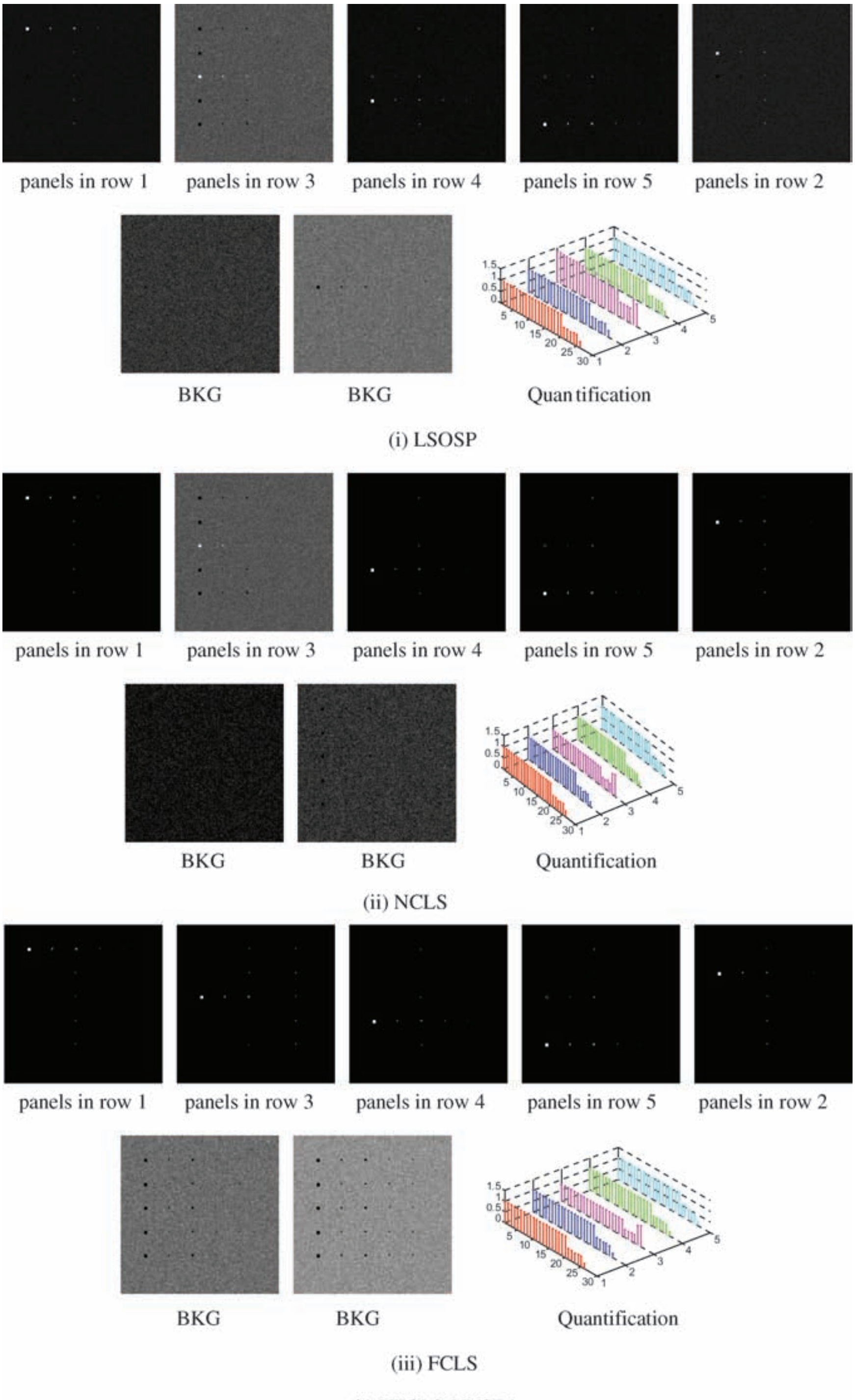
(i) LSOSP

(ii) NCLS

(iii) FCLS

(a) ATGP-UVSFA and UNCLS-UVSFA

Figure 17.5 Results of using LSOSP, NCLS, and FCLS to unmix TE2 via the target/VSs found in Figure 17.1(d).



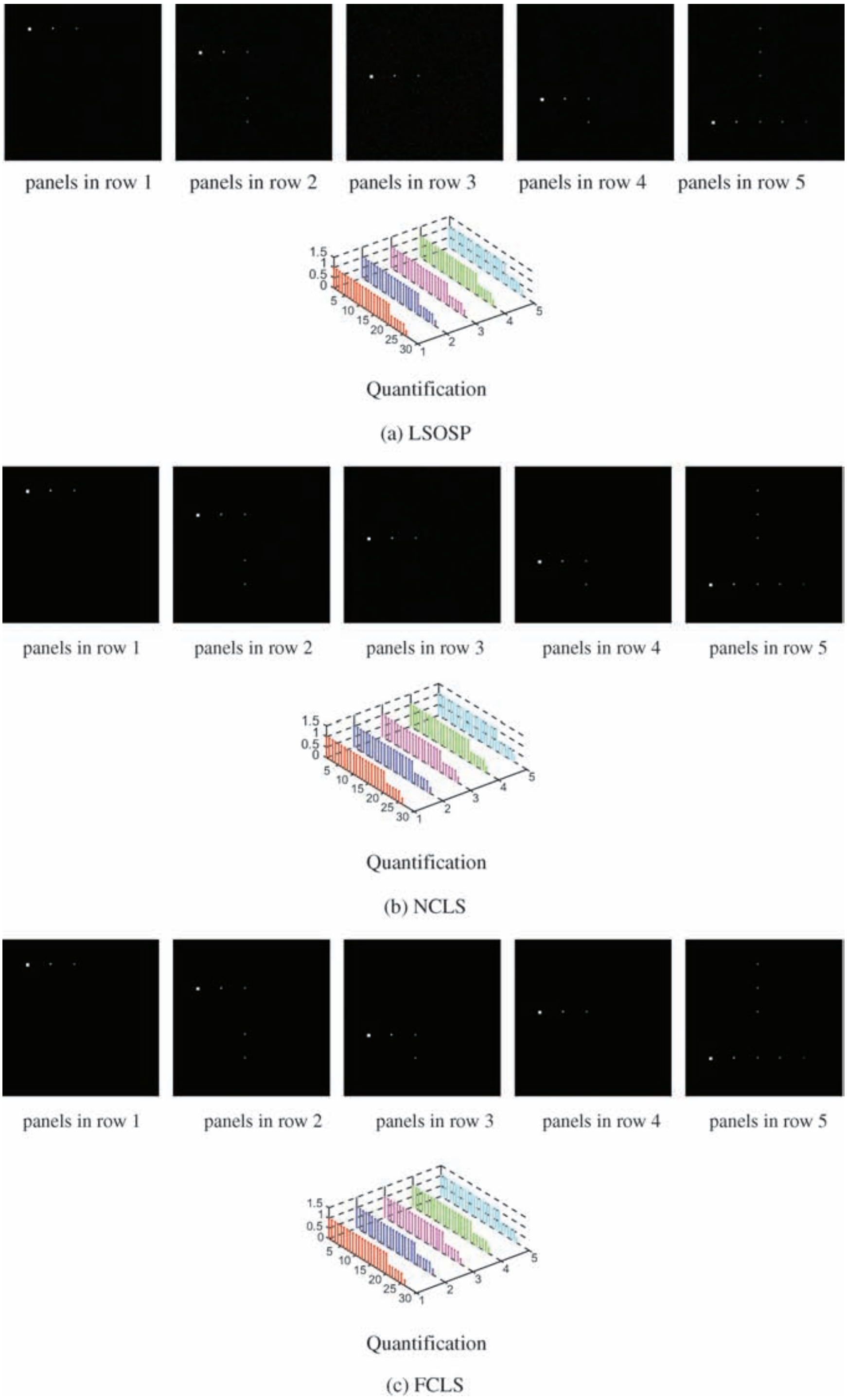
(i) LSOSP

(ii) NCLS

(iii) FCLS

(b) UFCLS-UVSFA

Figure 17.5 (*Continued*)

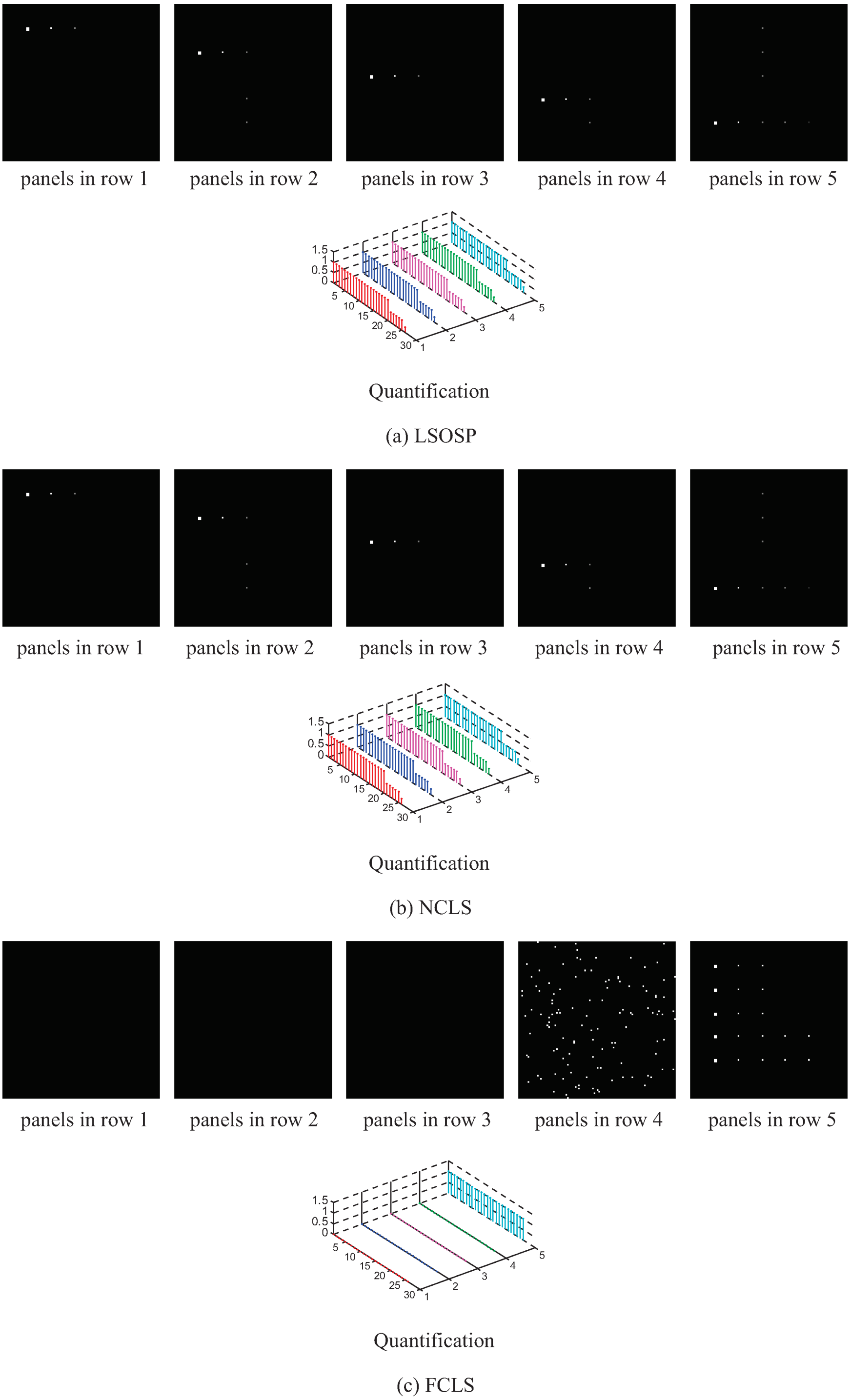


(a) LSOSP

(b) NCLS

(c) FCLS

Figure 17.6 Results using LSOSP, NCLS, and FCLS to unmix TI2 with assuming prior signature knowledge.



(a) LSOSP

(b) NCLS

(c) FCLS

Figure 17.7 Results using LSOSP, NCLS, and FCLS to unmix TE2 with assuming prior signature knowledge.

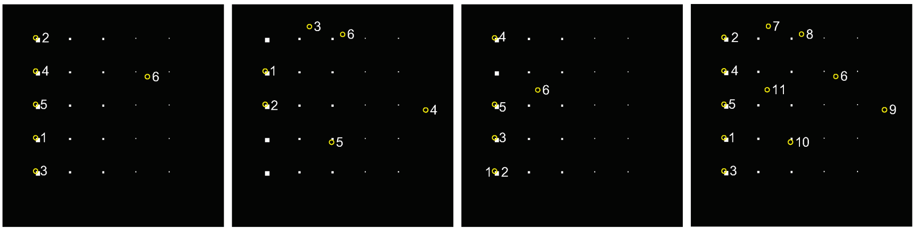
their unmixed results for all the 130 panel pixels along with their detected abundance fractions for the TI2 and TE2 scenarios, respectively.

Comparing the results in Figures 17.6 and 17.7 obtained by SLSMA with Figures 17.4 and 17.5 obtained by USLMA, both SLSMA and ULSMA produced comparable results in terms of quantifying 130 panel pixels for both TI2 and TE2 scenarios except one case of using FCLS to perform SLSMA to unmix TE2 scenario where the resulting abundance fractions for every single panel pixel in row 5 were estimated to be 100% as opposed to zero for every pixel in rows 1–4 as shown in Figure 17.7(c). The reasons for this can be explained as follows. First of all, the target panel pixels in TE2 scenario were superimposed over the BKG pixels so that the abundance fractions of panel pixels and BKG pixels were not summed up to one. However, even though the sum-to-one constraint assumption was violated in TE scenario, FCLS still tried to impose the constraint by giving all abundance fractions to the most distinct spectral signature, Muscovite, that was used to simulate panel pixels in row 5. For this particular case, the ULSMA was superior to the SLSMA because USLMA obtains target knowledge directly from the data where the target panel pixels were not pure anymore and they were actually mixed with the BKG signatures. This knowledge may be more realistic and accurate than the prior knowledge used by the SLSMA where the target panel pixels were assumed to be pure but certainly not true in TE scenario. However, for TI2 scenario the target panel pixels were implanted into the BKG with the corresponding BKG pixels removed to accommodate the inserted target panel pixels in which case the abundance sum-to-one constraint was still valid. As a result, FCLS performed well regardless of whether LSMA was performed in a supervised or an unsupervised manner. Since LSOSP and NCLS did not impose the sum-to-one constraint they also performed well for both TI2 and TE2 scenarios. These experiments also provided strong evidence of importance of using synthetic images to substantiate certain scenarios that were nearly impossible to use real image data to conduct experiments. In addition, with no complete ground truth available real images cannot be used for quantitative data analyses.

17.4.2 CA-ULSMA

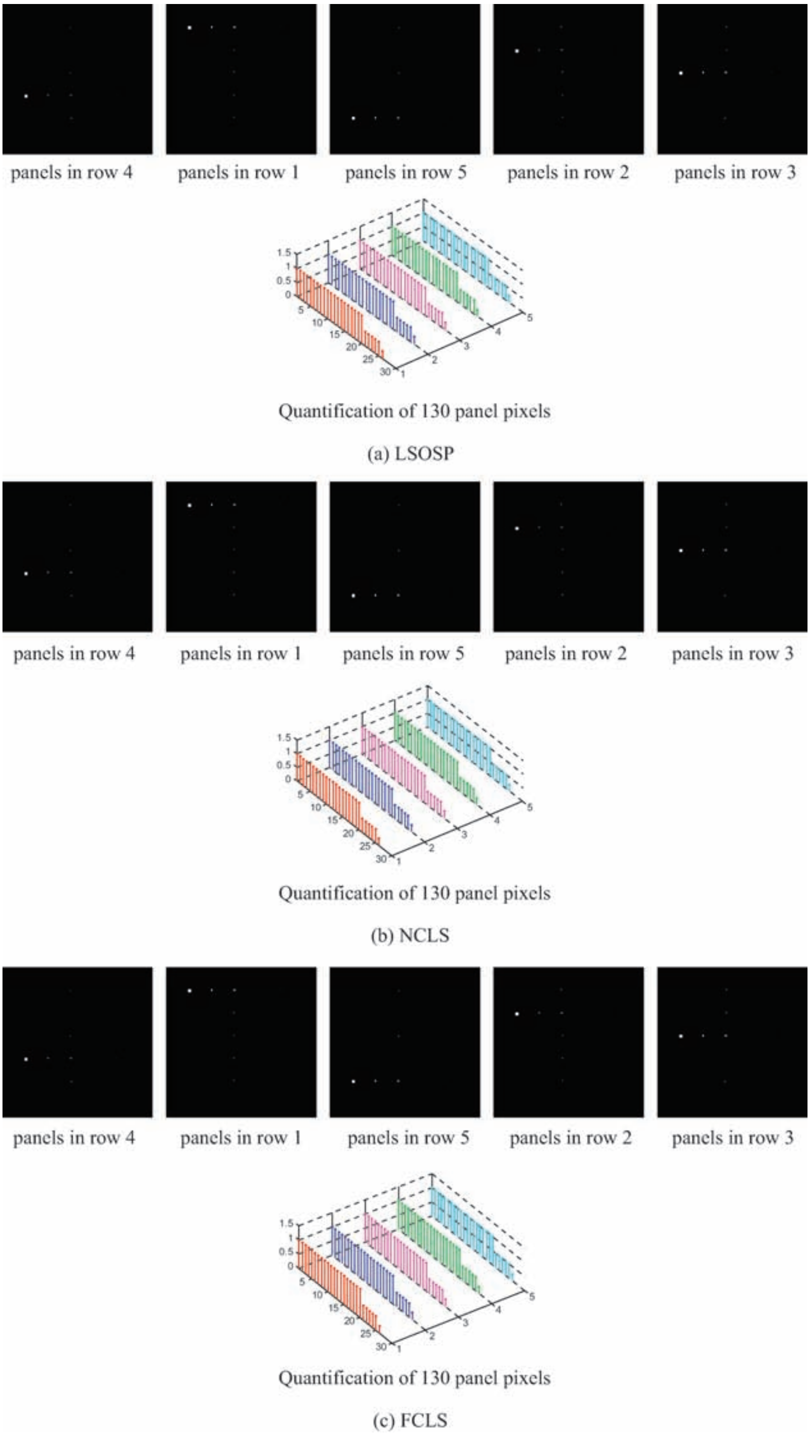
As an interesting alternative, this section conducted the same experiments performed by LS-based approaches for comparison. Figure 17.8(a) and (b) shows the six target VSs extracted by maximal IC projections and minimal IC projections respectively for TI scenario. Figure 17.8(c) shows the six BKG VSs extracted by maximum PC projections, while Figure 17.8(d) shows a total of 11 VSs obtained by merging the VSs generated in Figure 17.8(a)–(c).

The 11 VSs generated in Figure 17.8(d) were then used to form a VS matrix to perform linear spectral unmixing. Figure 17.9(a)–(c) is results of three LSMA methods, LSOSP, NCLS, and FCLS where only five target VSs in $\tilde{S}^{\text{target}} = \left\{\tilde{\mathbf{t}}_j^{\text{ICA}}\right\}$ corresponding to the five mineral signatures



(a) 6 IC Max VSs (b) 6 IC Min VSs (c) 6 PC Max VSs (d) 11 VSs used for unmixing

Figure 17.8 Targets found for scenario TI2: (a) six IC Max VSs; (b) six IC Min VSs; (c) six PC Max VSs; (d) 11 VSs to be used for unmixing.

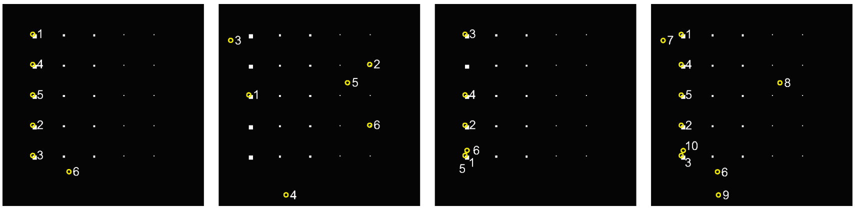


(a) LSOSP

(b) NCLS

(c) FCLS

Figure 17.9 Results of using LSOSP, NCLS, and FCLS to unmix scenario TI2 via the target pixels found in Figure 17.8(d).



(a) 6 IC Max VSs (b) 6 IC Min VSs (c) 6 PC Max VSs (d) 10 VSs used for unmixing

Figure 17.10 Targets found for scenario TE2: (a) six IC Max VSs; (b) six IC Min VSs; (c) six PC Max VSs; (d) 10 VSs to be used for unmixing.

were unmixed. Since the 10 subpanel pixels in fourth and fifth columns are not visible by inspection due to their size smaller than the spatial resolution, the quantification results of the 130 panel pixels are plotted for quantitative analysis where numerals 1–5 represent the number of rows and numerals 1–26 represent panel pixels in a particular row. For example, numbers 1–16, 17–20, 21–24, 25, and 26 represent 16 pixels in the first column, four pixels in the second column, four in the third column, one pixel in the fourth column, and one pixel in the fifth column, respectively. As shown in Figure 17.9 all the three methods performed very well including quantifying the abundances of 130 panel pixels.

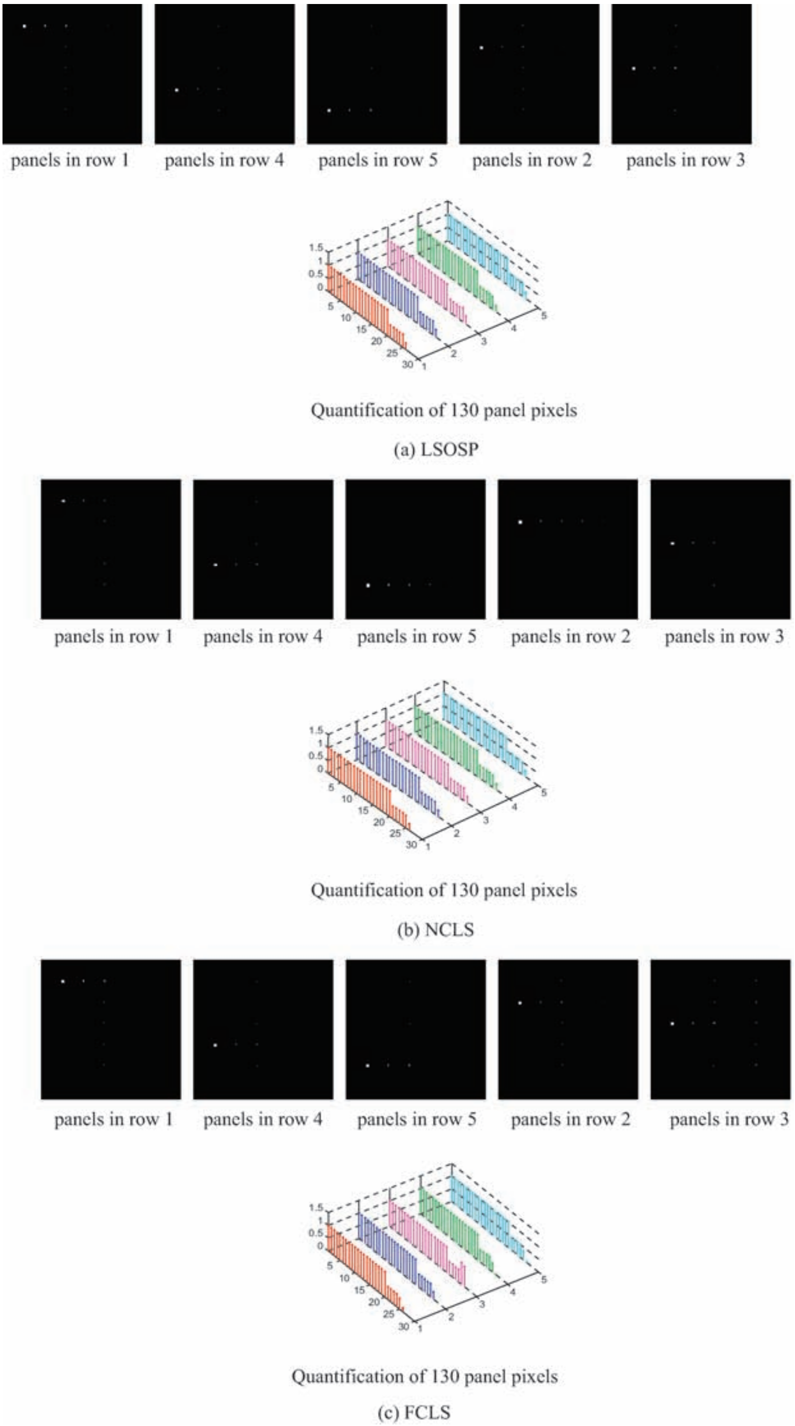
Similar experiments conducted for scenario TI2 were also performed for scenarios TE. Figure 17.10(a)–(c) shows VSs extracted by maximum IC projections, 6 minimal IC projections and 6 maximal PC projections, respectively. Figure 17.10(d) shows 10 VSs obtained by merging the VSs in Figure 17.10(a)–(c).

The 10 VSs generated in Figure 17.10(d) were then used to form a VS matrix to perform spectral unmixing. Figure 17.11(a)–(c) shows results of LSOSP, NCLS, and FCLS along with plotted quantification results of the 130 panel pixels where only 5 target VSs in $\tilde{S}^{\text{target}} = \left\{\tilde{\mathbf{t}}_j^{\text{ICA}}\right\}$ corresponding to the five mineral signatures were unmixed.

Comparing Figure 17.11 with 17.9 the unmixed results obtained for both scenarios TI2 and TE2 were very close except the two subpanel pixels in row 3 whose FCLS-estimated abundance fractions were 75% way beyond their true abundances. This may due to the fact that TE2 has panel pixels superimposed over the BKG pixels that violated the abundance sum-to-one constraint. This phenomenon will be more evidential in the following SLSMA experiments conducted for the scenario TE2 (Figure 17.13). Other than that the proposed CA-ULSMA performed really well for both scenarios.

In order to conduct a comparative analysis between CA-ULSMA and the SLSMA, we compare the CA-ULSMA results in Figures 17.9 and 17.11 to the results obtained by SLSMA in Figures 17.6 and 17.7. Like LS-ULSMA CA-LSMA also performed as well as LS-LSMA did and also comparably to SLSMA in terms of quantification for both scenarios TI2 and TE2 except the case of TE2 scenario that has been explained in the TI2 experiments where the FCLS-unmixed abundance fractions were unmixed to be 100% for every single panel pixel in row 5 while zero for every pixel in rows of 1–4 as shown in Figure 17.7(c).

As a concluding remark, it is important to realize the importance of the synthetic image-based experiments. With provided complete knowledge of abundance fractions simulated for each of 130 panel pixels we were able to conduct study and analysis on the quantitative performance of LSMA in spectral unmixing which real image experiments generally cannot provide. As shown in experiments, ULSMA using the VSs found by LS-based and CA-based



Quantification of 130 panel pixels

(a) LSOSP

Quantification of 130 panel pixels

(b) NCLS

Quantification of 130 panel pixels

(c) FCLS

Figure 17.11 Results of using LSOSP, NCLS, and FCLS to unmix scenario TE2 via the target VSs found in Figure 17.10(d).

unsupervised VS finding algorithms to perform spectral unmixing could be as at least effectively as SLSMA using real true endmembers. On some occasions such as the case where no real true endmembers are present ULSMA actually outperformed SLSMA due to the fact that the VSs used for ULSMA were real data sample vectors that were more realistic and reliable than the endmembers assumed by prior knowledge or visual inspection. Two additional comments are also worthwhile.

1. Although the two synthetic image scenarios seem simple, the value of the experiments should be appreciated. These two scenarios provide an objective validation of any designed algorithm under a fully controllable environment with complete ground truth. A good example is illustrated by Figure 17.7(c) where FCLS completely failed in TE2 because the sum-to-one abundance constraint was violated. If it had been applied to real data, we would not have known that a fully abundance constrained LSMA could not be used as a signal detection technique when the linear mixing model was used as a signal/noise detection model in which case a signal is embedded in a pixel corrupted by an additive noise like TE2 so that the abundance fractions of the signal and noise were not summed up to one. If an algorithm does not pass the synthetic image experiments, it will be very likely that it may not work in real data.
2. Due to significantly improved spectral resolution provided by hyperspectral imaging sensors hyperspectral imaging generally performs "*target*"-based spectral analysis rather than "*class-map/pattern*"-based spatial analysis as conduced in traditional image processing. Therefore, BKG VSs are usually not of major interest and no BKG analysis is necessary for hyperspectral imaging. Instead, they are only used for BKG suppression to improve target detection and classification. Because of that TI2 and TE2 scenarios suffice to serve this purpose where only complete knowledge of target panel pixels is required for target analysis and BKG can be made as simple as possible by adding Gaussian noise for suppression.

17.5 Real-Image Experiments

The HYDICE image scene in Figure 1.15 was chosen for real-image experiments because the ground truth of 15 panels specified by 19 R pixels is completely available for LSMA performance evaluation. Nevertheless, this ground truth should be only used to serve a reference since the scene is real data where uncharacterized spectral variations may affect the performance. Specifically, those pixels that are identified by a ground crew as panel center pixels may not actually pure pixels as demonstrated by Chang et al. (2004) and will be also shown in the following experiments. These experiments indicate that the prior knowledge may not be as reliable as it is supposed to be.

17.5.1 LS-ULSMA

First of all, the VD estimated for this scene, n_{VD}, is 9 with the false alarm probability $P_F \leq 10^{-3}$. Figure 17.12(a) shows the 9 target VSs that were extracted directly from the original data by the ATGP and were considered as a set of BKG VSs $\text{S}^{\text{BKG}} = \left\{\mathbf{b}_j^{\text{ATGP}}\right\}_{j=1}^{9}$ that included three panel pixels from rows 1, 3, and 5. Figure 17.12(b) shows the 9 target VSs extracted from the sphered data by ATGP that included five panel panels extracted from each of five rows and were considered as a set of target VSs, $\text{S}^{\text{target}} = \left\{\mathbf{t}_j^{\text{ATGP}}\right\}_{j=1}^{9}$. Figure 17.12(c) singles out the five VSs that were identified as BKG VSs, $\tilde{S}^{\text{BKG}} = \left\{\tilde{\mathbf{b}}_i^{\text{ATGP}}\right\}$ by removing the four target VSs using a similarity measure such as SAM, and Figure 17.12(d) shows a total number of 14 VSs obtained by combining the

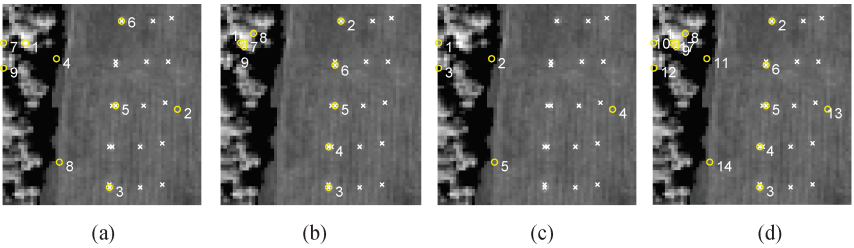


Figure 17.12 ATGP-generated BKG and target VSs: (a) nine BKG VSs in original data; (b) nine target VSs in sphered data; (c) five BKG VSs not identified as target VSs; (d) 14 VSs obtained by merging the VSs in (b–c).

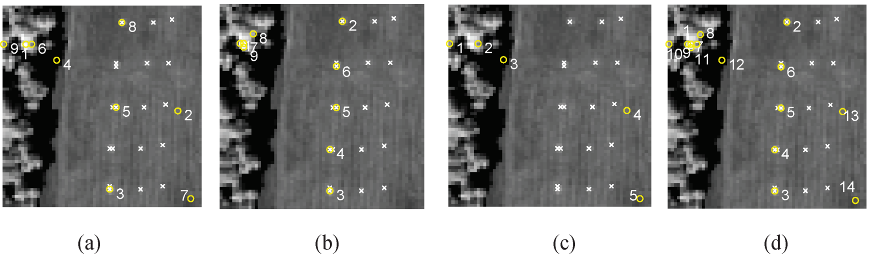


Figure 17.13 UNCLS-generated BKG and target VSs: (a) nine BKG VSs in original data; (b) nine target VSs in sphered data; (c) five BKG VSs not identified as target VSs; (d) 14 VSs obtained by combining the VSs in (b–c).

BKG VS set $\tilde{S}^{\mathrm{BKG}}$ in Figure 17.12(c) with the target VS set S^{target} in Figure 17.12(b) into a BKG-target VS merged set $\tilde{S}^{\mathrm{BKG}} \cup S^{\mathrm{target}}$ to be used for spectral unmixing where the numbers in the figures indicated the orders of VSs extracted by the ATGP.

Similarly, Figures 17.13(d)–17.14(d) also show 14 VSs produced by the UNCLS including nine target VSs and five BKG VSs and 15 VSs extracted by the UFCLS including nine target VSs and six BKG VSs. Since the target VSs of interest were those extracted by the three LS-based algorithms in Figures 17.12(b), 17.13(b), and 17.14(b) from the sphered data they should have included five pure targets VSs that corresponded to all the five pure panel signatures. This was exactly the case where these five pure panel pixels, p_{11}, p_{221}, p_{312}, p_{411}, and p_{521} were found and identical in

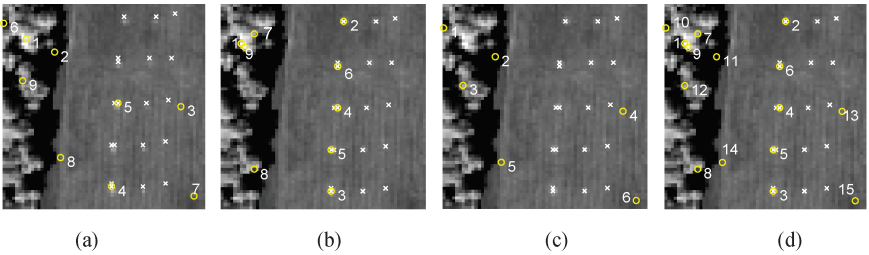


Figure 17.14 UFCLS-generated BKG and target VSs: (a) nine BKG VSs in original data; (b) nine target VSs in sphered data; (c) six BKG VSs not identified as target VSs; (d) 15 VSs obtained by merging the VSs in (b–c).

Figures 17.12(b), 17.13(b), and 17.14(b). Additionally, among these five pure panel pixels, p_{11}, p_{312}, and p_{521} were the only three target VSs extracted as BKG VSs in the original data in Figures 17.12(a), 17.13(a), and only two panel pixels p_{312} and p_{521} extracted in 17.14(a). This is due to the fact that the panel pixels in rows 2 and 4 have very similar signatures to those in rows 3 and 5, respectively, according to the ground truth in which case they were not extracted as endmembers. Interestingly, these three pure target VSs were the only endmembers extracted by any endmember extraction algorithm except the case when dimensionality reduction is performed by the independent component analysis (ICA) as shown in the following section.

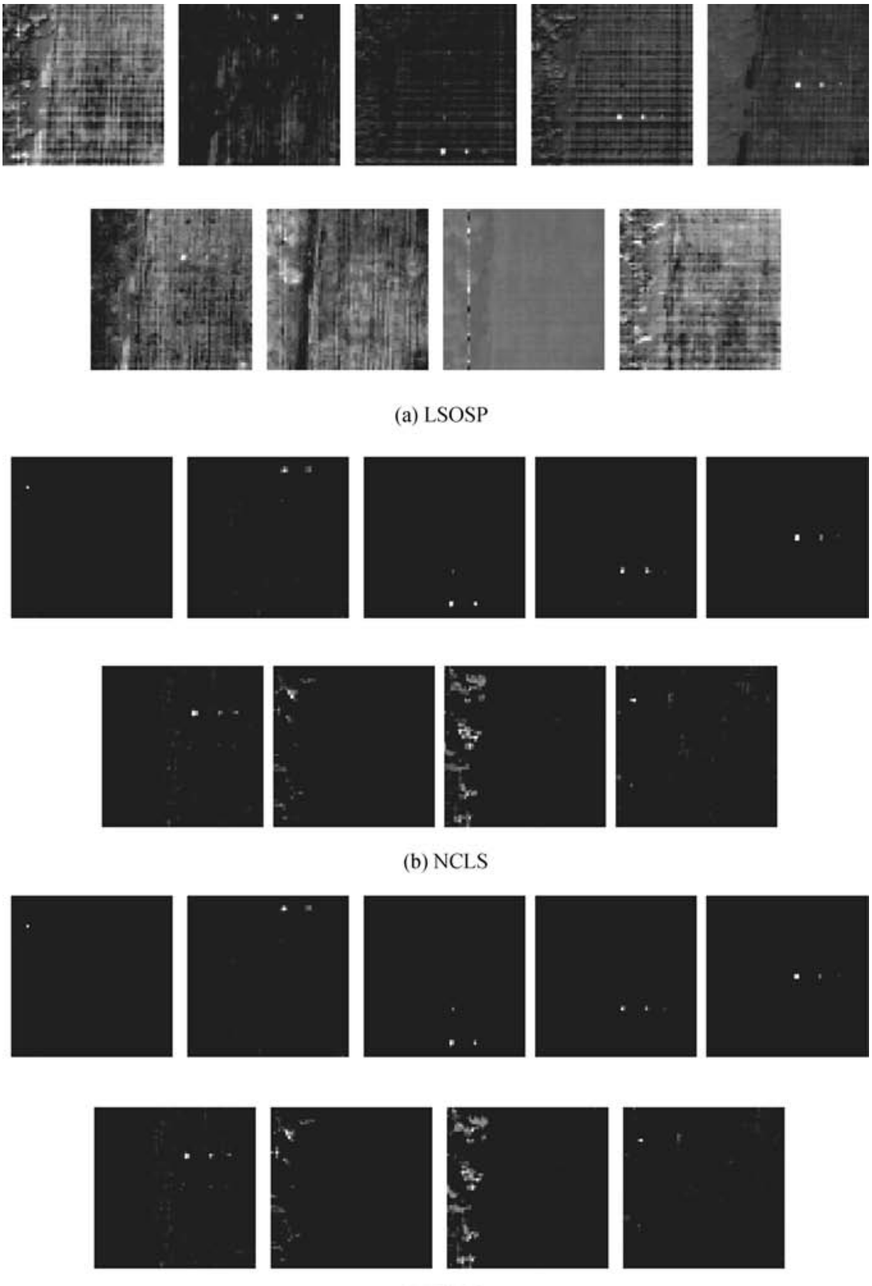
In order for LSMA to perform effectively, the targets signatures used to form the signature matrix **M** must include all the targets VSs $\left\{\mathbf{t}_j^{\mathrm{LS}}\right\}_{j=1}^{n_{\mathrm{VD}}}$ and BKG VSs $\left\{\tilde{\mathbf{b}}_i^{\mathrm{LS}}\right\}_{i=1}^{n_{\mathrm{BKG}}}$ to represent the entire data where the target VSs $\left\{\mathbf{t}_j^{\mathrm{LS}}\right\}_{j=1}^{n_{\mathrm{VD}}}$ are the target signatures we would like to unmix and the BKG VSs $\left\{\tilde{\mathbf{b}}_i^{\mathrm{LS}}\right\}_{i=1}^{n_{\mathrm{BKG}}}$ are considered as undesired signatures that can be suppressed to enhance target classification performance. The three LSMA methods, LSOSP, NCLS, and FCLS, were used to unmix high-order target VSs $\left\{\mathbf{t}_j^{\mathrm{LS}}\right\}_{j=1}^{n_{\mathrm{VD}}}$ extracted from the sphered data, each of which was considered to represent one specific target class. Figures 17.15(a)–(c) to 17.17(a)–(c) show their corresponding results where figures labeled by (a), (b), and (c) are unmixed results by LSOSP, NCLS, and FCLS, respectively.

Obviously, the results obtained by NCLS and FCLS performed better than that produced by LSOSP in Figures 17.15–17.17 due to the imposed constraints on abundance fractions. Interestingly, while the results in Figure 17.15(b) and (c) were similar to the results obtained in Figures 17.16(b) and (c) and 17.17(b) and (c), the umixed results obtained by LSOSP in Figure 17.15(a) were slightly better than those in Figures 17.15(a) and 17.16(a) in terms of detection of 15 panels in five rows. This improvement was mainly due to the fact that UFCLS produced six BKG VSs to perform better BKG suppression rather than five BKG VSs produced ATGP and UNCLS. Also, it should be noted that there were of course more BKG pixels that could be used for this purpose. As a matter of fact, in Heinz and Chang (2001) and Chang (2003a, Chapter 5) there were 34 pixels found by the unsupervised FCLS for spectral unmixing. The results were similar to those presented in our experiments using only 9 image endmembers. In this case, "9" is probably sufficiently enough for LSMA to perform spectral unmixing well.

17.5.2 CA-ULSMA

Once again the value of VD was set to $p = n_{\mathrm{VD}} = 9$ for CA-UVSFA to find nine signatures from nine PCs and signatures with maximal and minimal projections in nine ICs. Figure 17.18(a) shows the nine target VSs extracted by maximal IC projection. Figure 17.18(b) shows the nine target VSs extracted by minimal IC projection. Figure 17.18(c) shows the nine BKG VSs extracted by maximal PC projections, and Figure 17.18(d) shows the total 19 VSs obtained by merging signatures in Figure 17.18(a)–(c). The 19 VSs obtained in Figure 17.18(d) are the used as VSs for the signature matrix **M** to form a linear mixture model and LSOSP, NCLS, and FCLS were implemented to perform unmixing. Figure 17.19(a)–(c) shows their abundance maps of unmixed results, respectively.

Obviously, the unmixed results obtained by NCLS in Figure 17.19(b) and FCLS in Figure 17.19(c) were much better than those obtained by LSOSP in Figure 17.19(a) due to the imposed abundance constraints. Interestingly, the pixels in Figures 17.12(b), 17.3(b), and

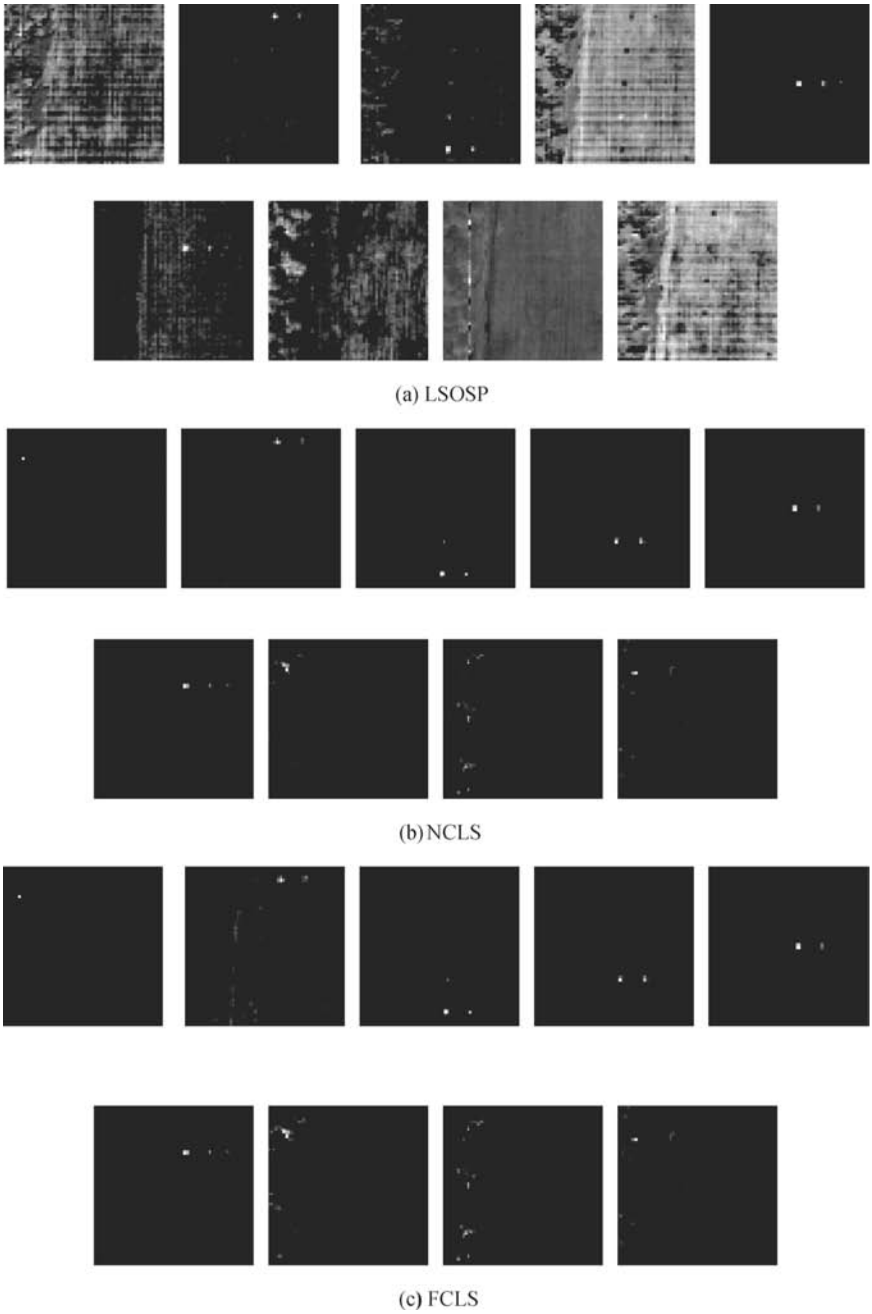


(a) LSOSP

(b) NCLS

(c) FCLS

Figure 17.15 Nine target classes obtained by LSOSP, NCLS, and FCLS using the target pixels generated by ATGP-UVSFA.

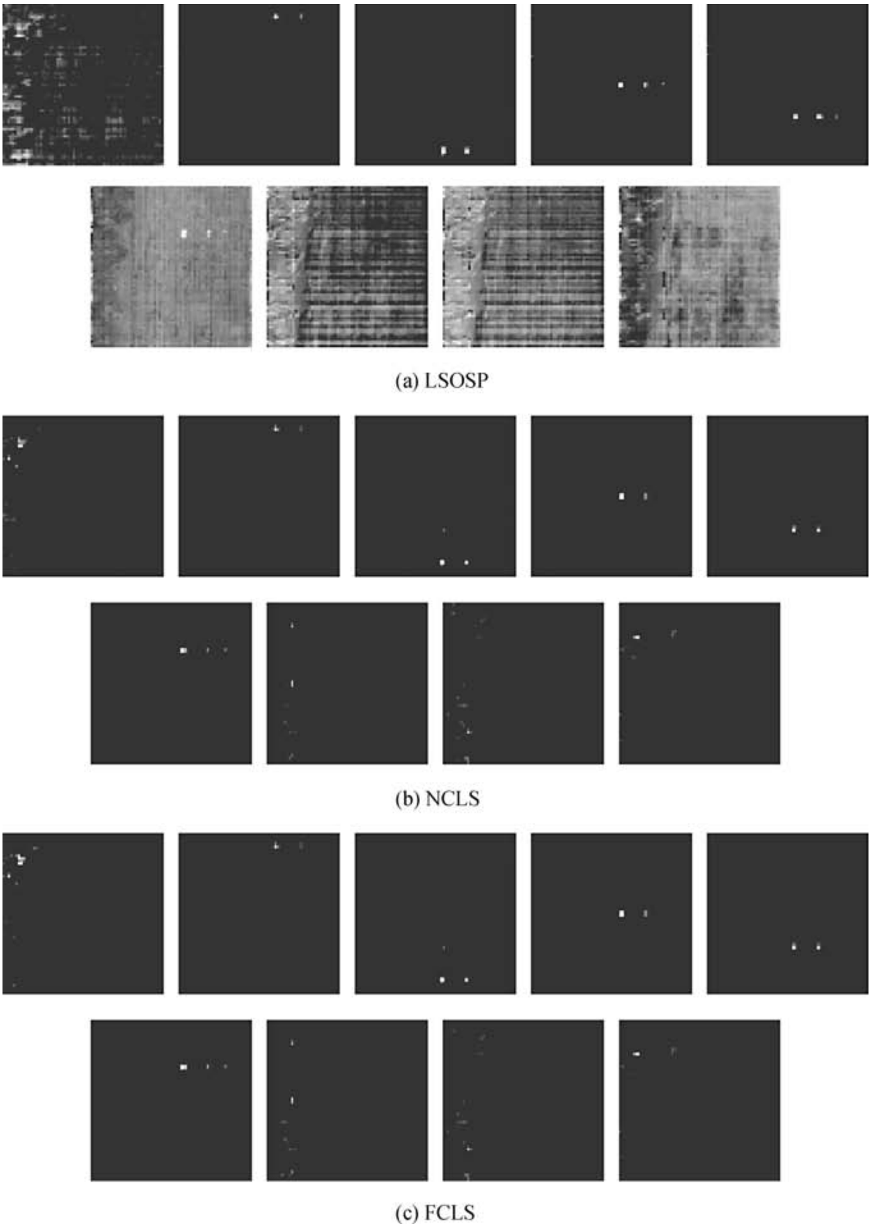


(a) LSOSP

(b) NCLS

(c) FCLS

Figure 17.16 Nine target classes obtained by LSOSP, NCLS, and FCLS using the target pixels generated by UNCLS-UVSFA.



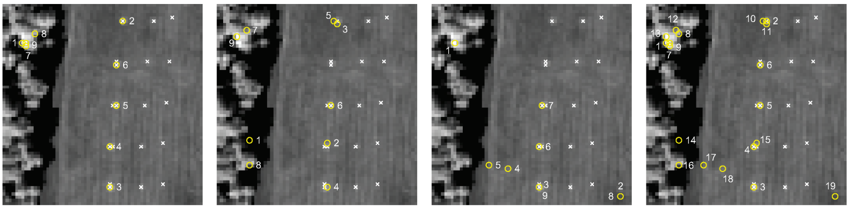
(a) LSOSP

(b) NCLS

(c) FCLS

Figure 17.17 Nine target classes obtained by LSOSP, NCLS, and FCLS using the target pixels generated by UFCLS-UVSFA.

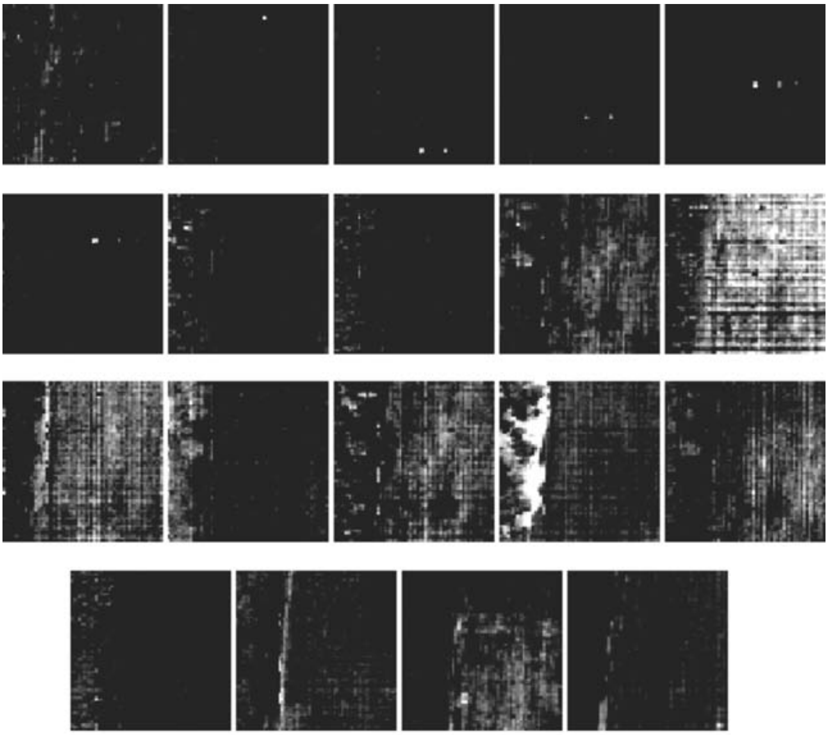
17.4(b) found by LS-UVSFA via the sphered data and pixels in Figure 18(a) found by CA-UVSFA via ICA included five identical panel pixels, p_{11}, p_{221}, p_{312}, p_{411}, and p_{521} that specify five distinct panel signatures. If we assume that these five pixels were the main target signatures of interest and pixels other than target signatures were considered as BKG signatures, then the only differences between these two algorithms was the BKG signature extraction



(a) 9 IC Max pixels (b) 9 IC Min pixels (c) 9 PC Max pixels (d) 19 pixels merged by (a)–(c)

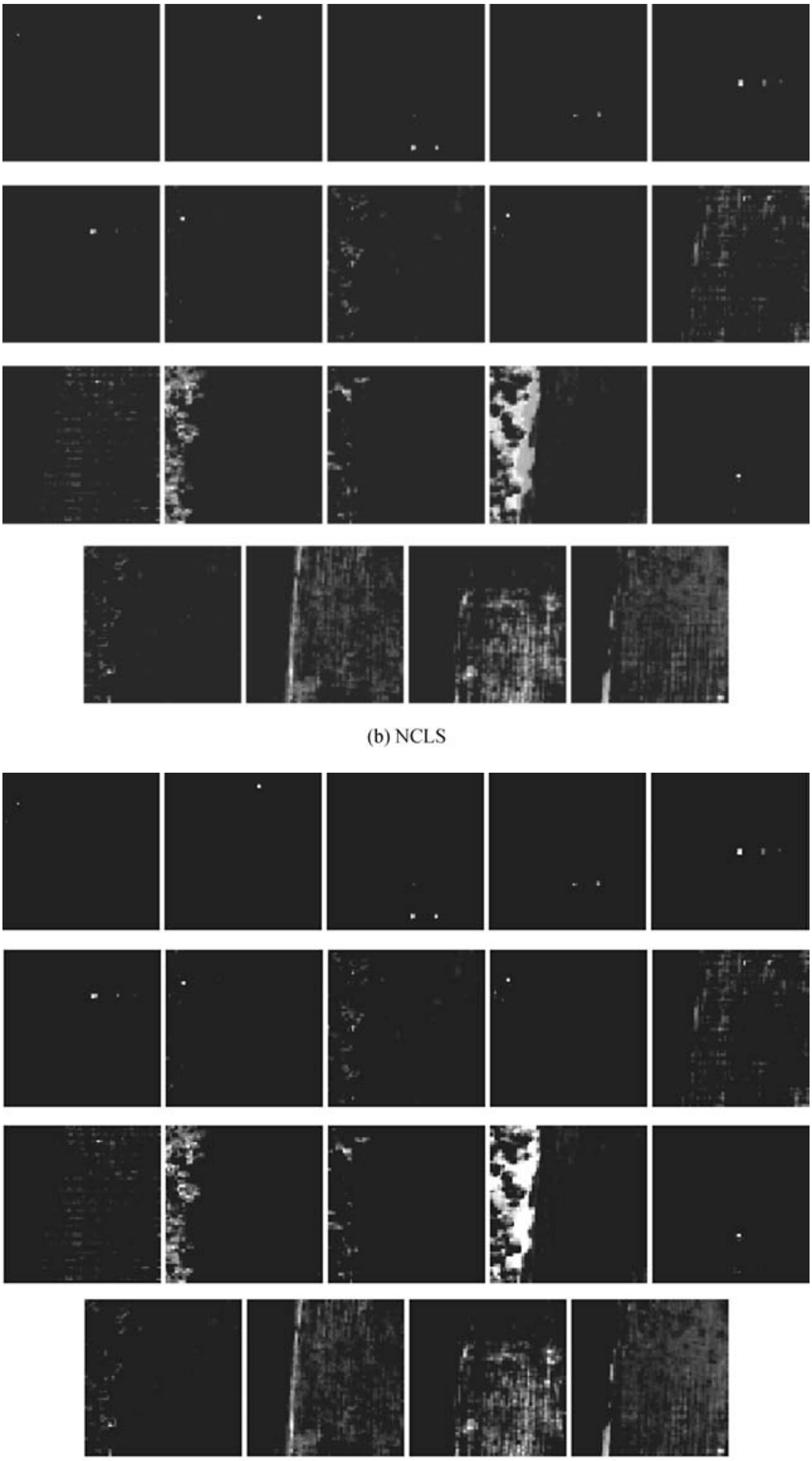
Figure 17.18 CA-UTFA results for HYDICE scene: (a) nine IC Max VSs; (b) nine IC Min VSs; (c) nine PC Max VSs; (d) 19 VSs found by merging results in (a)–(c).

where more BKG signatures were produced by CA-UVSFA than LS-UVSFA for LSMA to perform BKG suppression. The fact that more BKG signatures were used by CA-LSMA for BKG suppression than that by LS-LSMA our visual inspection showed that the unmixed results of panel pixels in five rows in Figure 17.19 by CA-LSMA were more cleaner and clearer than those in Figures 17.15–17.17 unmixed by LS-LSMA. However, did this also imply that CA-LSMA produced more accurate quantification results of 19 R pixels than LS-LSMA? Interestingly, the answer is “not necessarily true.” The following section is included to address this issue.



(a) LSOSP

Figure 17.19 Nineteen classes obtained by LSOSP, NCLS, and FCLS.



(b) NCLS

(c) FCLS

Figure 17.19 (*Continued*)

17.5.3 Qualitative and Quantitative Analyses between ULSMA and SLSMA

Two important factors affect the performance of LSMA: capability in unmixng and ability in BKG suppression. An early development of orthogonal subspace projection (OSP) (Harsanyi and Chang, 1994) was designed for this purpose to make LSMA work more effectively by separating signatures into desired signatures to be considered as target signatures and undesired signatures to be considered as BKG signatures. This concept is further confirmed by ULSMA presented in Sections 17.2 and 17.3. This section further demonstrates that the performance of ULSMA using the designed UVSFA is superior to that of SLSMA using assumed prior signature knowledge in sense of target signature unmixing and BKG suppression.

In order to make SLSMA work more effectively, we must include other BKG signatures in a linear mixing model so that a better BKG suppression can result in a better spectral unmixing. Using full knowledge of the scene by visual inspection and prior knowledge provided by ground truth there were at least three identified BKG signatures, grass, tree, and road shown in Figure 17.20 in addition to 19 center panel pixels marked by red shown in Figure 1.16(b).

There is an interesting observation in the scene worth noting. It has been shown in many experiments that there was an interferer marked in Figure 17.20 that generally could not be identified by visual inspection or prior knowledge but has very strong energy that can be always extracted by any unsupervised target detection algorithm such as algorithms developed in Chang (2003a). If this interferer signature is included with the five panel signatures in Figure 1.16 and the other three BKG signatures to make up five target classes representing five panel signatures and four BKG classes to form a signature matrix **M** for spectral unmixing, there are exactly nine signatures estimated by VD. This fact provides further evidence that VD is an effective estimation method to estimate the number of image endmembers for spectral unmixing. However, in order to see if including the interferer in the signature matrix **M** makes a difference, Figures 17.21(a)–(c) and 17.22(a)–(c) show respective unmixed results of 19 panel pixels via abundance fractional maps produced by three SLSMA methods, LSOSP, NCLS, and FCLS using an 8-signature matrix $\mathbf{M} = [\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4,\mathbf{p}_5,$ grass,tree,road$]$ without the interferer and a 9-signature matrix $\mathbf{M} = [\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4,\mathbf{p}_5,$ grass,tree,road,interferer$]$, both of which use the five panel signatures in Figure 1.16 to represent $\mathbf{p}_1,\mathbf{p}_2,\mathbf{p}_3,\mathbf{p}_4,\mathbf{p}_5$. Since the BKG classes are not of major interest, the unmixed results for the four BKG classes are not included in the figures.

Comparing the results in Figure 17.22 to that in Figure 17.21, it was apparent that the effect of the interferer was very significant, specifically for LSOSP. If we further compare the results in Figures 17.21–17.22 to Figures 17.15–17.17 and Figure 17.19 it was also clear that ULSMA performed significantly better than its supervised counterpart, SLSMA, in terms of unmixing the 19 R

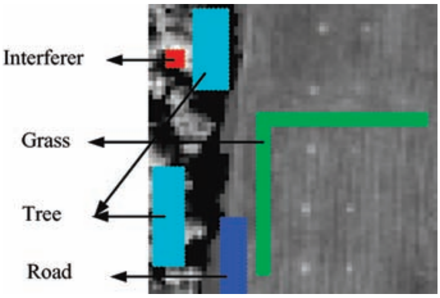


Figure 17.20 Areas identified by ground truth and marked by three BKG signatures, grass, tree, and road plus an interferer.

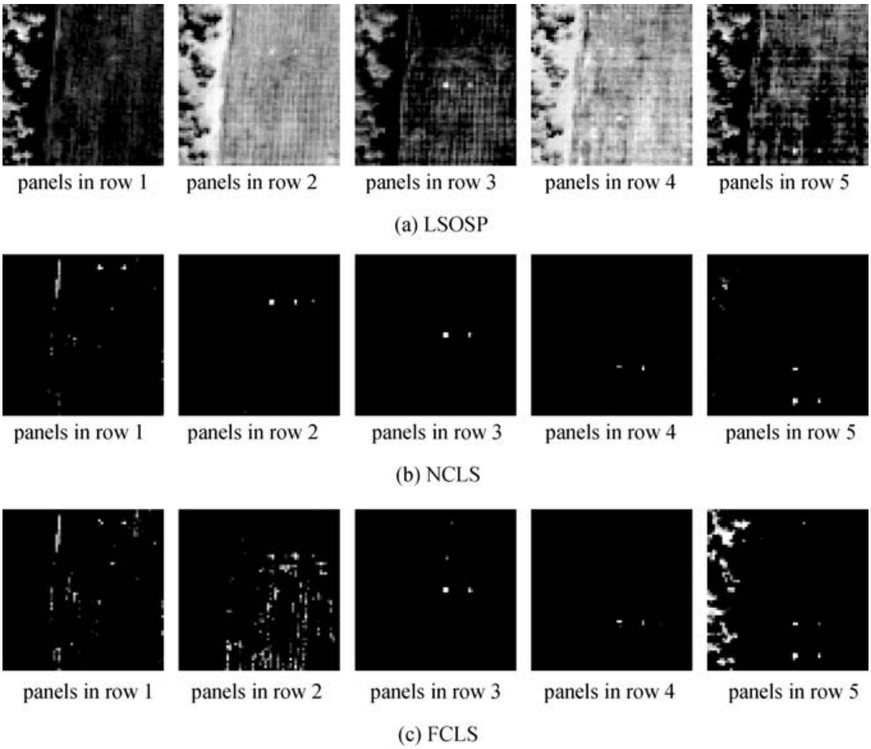
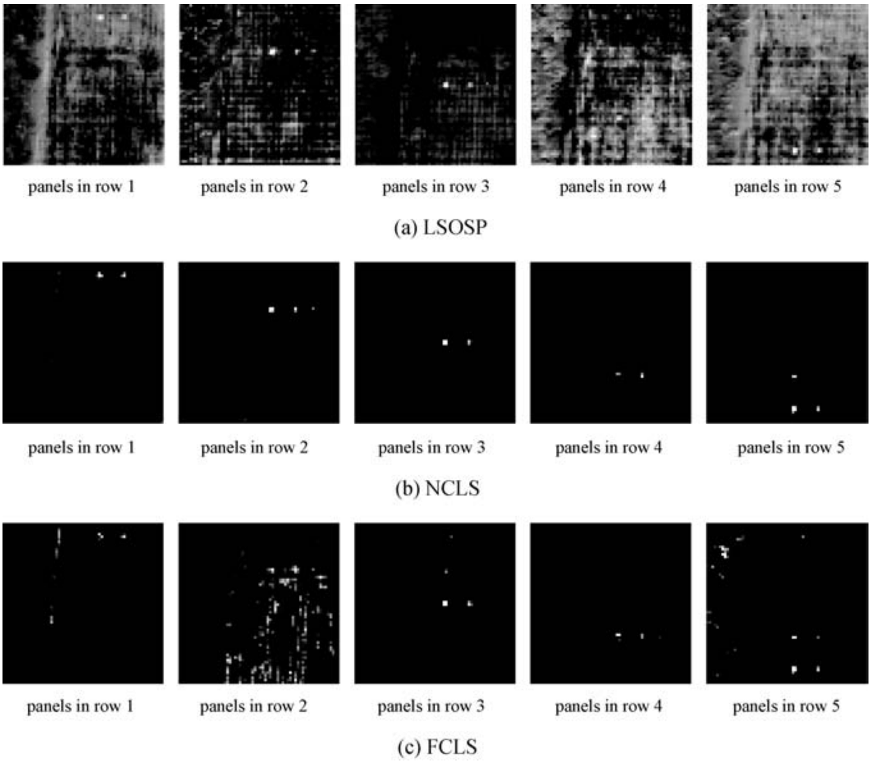


Figure 17.21 Unmixed results of 15 panels with 19 panel pixels by LSOSP, NCLS, and FCLS using five panel signatures in Figure 1.16 and 3 BKG signatures, grass, tree, and road obtained by marked areas inn Figure 17.20.

panel pixels, specifically panel pixels in rows 3 and 5. These experiments further demonstrated that SLSMA was not as effective as ULSMA when it came to real data mainly due to the unknown knowledge about the BKG that played a key role in BKG suppression for LSMA.

The above unmixed results are evaluated qualitatively by visual assessment. The conclusions may not be objective. So, Table 17.1 further provides quantification results in Figures 17.15–17.17 and tabulates the unmixed abundance fractions of 19 R panel pixels in Figures 17.15–17.17 produced by LSOSP, NCLS, and FCLS using VSs found by the three LS-based UVSFAs, ATGP, UNCLS, and UFCLS. The bottom row in the table calculated the total error for each of methods based on the sum of squared errors between the estimated abundance fractions of 19 R panel pixels and their ground truth provided in the second column. Since the 5 R panel pixels in the third column are subpanel pixels, their ground truth panel abundance fractions were calculated based on the ratio of their size to the pixel size, which is $(1\,\mathrm{m} \times 1\,\mathrm{m})/(1.56\,\mathrm{m} \times 1.56\,\mathrm{m}) \approx 10/25 = 0.4$. According to Table 17.1 the abundance fractions of the 19 R panel pixels estimated by NCLS and FCLS were very close. Surprisingly, the total error resulting from LSOSP was smallest, while FCLS resulted in the largest total errors. This conclusion was completely reversed by visual assessment based on Figures 17.15–17.17. However, if we compare the unixed abundance fractions of the individual 19 R panel pixels against their respective ground truth FCLS was indeed the best compared to LSOSP that was still the worst. This simple example indicated that simple total quantification errors did not provide a complete picture of how effective LSMA may perform. In



(a) LSOSP

(b) NCLS

(c) FCLS

Figure 17.22 Unmixed results of 15 panels with 19 panel pixels by LSOSP, NCLS, and FCLS using five panel signatures in Figure 1.16 and 4 BKG signatures, grass, tree, road, and interferer obtained by marked areas in Figure 17.20.

particular, the quantification errors were only calculated only based on 19 R panel pixels by completly discarding the effect resulting from BKG suppression which can be clearly seen in Figures 17.15–17.17.

Further, Table 17.2 also tabulates the unmixed abundance fractions of 19 R panel pixels in Figure 17.19(a)–(c) produced by LSOSP, NCLS, and FCLS using VSs found by CA-based UVSFA where the total error for each of methods was calculated based on the sum of squared errors between the unmixed abundance fractions of 19 R panel pixels and their ground truth provided in the second column. The conclusions drawn from this table were the same as what we concluded for Table 17.2 where the total error resulting from LSOSP was smallest compared to the largest total error produced by FCLS. Similarly, FCLS performed generally better than LSOSP if each of the individual 19 R panel pixels is taken into consideration for comparison.

Comparing the results in Table 17.2 to the results in Table 17.1 the experiments suggested that LS-LSMA generally performed a little bit better than CA-LSMA in terms of total error. Specifically, the unmixed abundance fraction of p_{13} produced by NCLS and FCLS was zero in Table 17.2 as opposed to nonzero unmixed abundance fractions of p_{13} in Table 17.1.

To compare the quantification results produced in Table 17.1 by LS-ULSMA and Table 17.2 by CA-ULSMA, Table 17.3 also tabulates unmixed abundance fractions of 19 panel pixels in Figures 17.23 and 17.24 produced by SLSMA using prior knowledge provided by Figures 1.16

Table 17.1 Estimated abundance fractions of 19 panel pixels produced by LSOSP, NCLS, and FCLS using BKG and target VSs found in Figures 17.15–17.17 by ATGP-UVSFA, UNCLS-UVSFA, and UFCLS-UVSFA

Panel pixel	Ground truth	$\left\{\tilde{\mathbf{b}}_i^{\text{ATGP}}\right\} \cup \left\{\mathbf{t}_j^{\text{ATGP}}\right\}_{j=1}^{9}$			$\left\{\tilde{\mathbf{b}}_i^{\text{UNCLS}}\right\} \cup \left\{\mathbf{t}_j^{\text{UNCLS}}\right\}_{j=1}^{9}$			$\left\{\tilde{\mathbf{b}}_i^{\text{UFCLS}}\right\} \cup \left\{\mathbf{t}_j^{\text{UFCLS}}\right\}_{j=1}^{9}$		
		LSOSP	NCLS	FCLS	LSOSP	NCLS	FCLS	LSOSP	NCLS	FCLS
p_{11}	1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
p_{12}	1	0.4096	0.4332	0.4120	0.3562	0.4165	0.4001	0.4085	0.4148	0.3850
p_{13}	0.4	0.0002	0.0887	0.0841	−0.1073	0.0308	0.0465	0.0142	0.0307	0.0250
p_{211}	1	0.8421	0.8403	0.8404	0.9180	0.8413	0.8209	0.8648	0.8384	0.8453
p_{221}	1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
p_{22}	1	0.6164	0.6257	0.7308	0.6351	0.6607	0.7127	0.6990	0.6126	0.7405
p_{23}	0.4	0.5525	0.4774	0.4724	0.3478	0.4168	0.4153	0.3798	0.4471	0.4498
p_{311}	1	0.8741	0.8674	0.8627	0.9094	0.8674	0.8628	0.8969	0.8671	0.8634
p_{321}	1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
p_{32}	1	0.5027	0.4249	0.4192	0.5906	0.4713	0.4727	0.5925	0.5149	0.4922
p_{33}	0.4	0.2516	0.2614	0.2655	0.3541	0.2959	0.2929	0.3388	0.2880	0.2886
p_{411}	1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
p_{412}	1	0.7685	0.3137	0.3876	0.5827	0.3222	0.3605	0.7976	0.3407	0.3923
p_{42}	1	0.8085	0.6761	0.6657	0.7965	0.7495	0.7485	0.8414	0.7480	0.7477
p_{43}	0.4	0.2363	0.1789	0.1473	0.5047	0.2851	0.2633	0.2790	0.1227	0.1542
p_{511}	1	0.7204	0.7224	0.7215	0.6954	0.7245	0.7198	0.6973	0.7213	0.7235
p_{521}	1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
p_{52}	1	0.7645	0.7770	0.7689	0.7027	0.7460	0.7244	0.7228	0.7753	0.7740
p_{53}	0.4	0.1452	0.1545	0.1537	−0.0144	0.0000	0.0017	0.1215	0.1471	0.1554
Total errors		1.305	1.769	1.665	1.572	1.761	1.712	1.115	1.690	1.582

Table 17.2 Quantification results of abundance fractions of 19 panels estimated by CA-ULSMA

Panel pixels	Ground truth	LSOSP	NCLS	FCLS
p_{11}	1	1.000	1.000	1.000
p_{12}	1	0.323	0.357	0.313
p_{13}	0.4	−0.190	0.000	0.000
p_{211}	1	0.810	0.800	0.800
p_{221}	1	1.000	1.000	1.000
p_{22}	1	0.623	0.657	0.777
p_{23}	0.4	0.376	0.456	0.454
p_{311}	1	0.864	0.869	0.864
p_{321}	1	1.000	1.000	1.000
p_{32}	1	0.587	0.511	0.513
p_{33}	0.4	0.430	0.374	0.374
p_{411}	1	1.000	1.000	1.000
p_{412}	1	0.756	0.308	0.372
p_{42}	1	0.781	0.734	0.740
p_{43}	0.4	0.281	0.126	0.213
p_{511}	1	0.682	0.716	0.721
p_{521}	1	1.000	1.000	1.000
p_{52}	1	0.747	0.783	0.777
p_{53}	0.4	0.096	0.132	0.144
Total errors		1.556	1.816	1.672

Table 17.3 Estimated abundance fractions of 19 panel pixels in Figures 17.21–17.22 produced by LSOSP, NCLS, and FCLS using five panel signatures obtained by averaging all R pixels in the first three columns using (three BKG signatures/four BKG signatures)

	LSOSP	NCLS	FCLS
p_{11}	1.3876/1.4475	0.8420/0.8309	0.0177/0.0177
p_{12}	0.9377/0.9155	0.8821/0.8510	0.8813/0.8199
p_{13}	0.6747/0.6370	0.2735/0.2735	0.3349/0.1940
p_{211}	1.1520/1.2384	0.8115/0.8115	0.5474/0.5474
p_{221}	1.1516/1.3146	0.7945/0.7945	0.3455/0.3455
p_{22}	0.9771/0.8558	0.8558/0.8558	0.8522/0.8522
p_{23}	0.7193/0.5912	0.4843/0.4843	0.4992/0.4992
p_{311}	1.2467/1.2482	0.8770/0.8809	0.8552/0.8299
p_{321}	1.5336/1.4713	0.9149/0.8953	0.7960/0.7913
p_{32}	0.7966/0.8240	0.5935/0.5935	0.7388/0.7388
p_{33}	0.4231/0.4565	0.2761/0.2761	0.2710/0.2710
p_{411}	1.1220/1.2356	0.1075/0.1617	0.0000/0.0000
p_{412}	1.1411/1.1672	−0.0000/−0.0000	0.0000/0.0000
p_{42}	1.2811/1.2331	0.9555/0.9555	0.4782/0.4782
p_{43}	0.4557/0.3641	0.2393/0.2393	0.2004/0.2004
p_{511}	1.1670/0.1770	0.9892/0.9599	1.0000/0.9759
p_{521}	1.5316/1.4698	1.1210/0.9551	1.0000/1.0000
p_{52}	1.0845/1.0760	1.0467/0.9925	1.0000/1.0000
p_{53}	0.2169/0.2772	0.2029/0.2029	0.1765/0.1765

and 17.20 where each entry in Table 17.3 has two values with the one before and the one after "/" indicating the results produced by using three BKG signatures (grass, tree, and road) and four BKG signatures (grass, tree, road, and interferer), respectively.

An interesting finding from Table 17.3 was that the abundance fractions of panel pixels p_{412} unmixed by NCLS and p_{411}, p_{412} unmixed by FCLS were 0. This was caused by the fact that the five panel signatures $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ in Figure 1.16 used for spectral unmixing were not really pure signatures because the panel pixels in the third column that were included for averaging were actually subpixels and not pure signatures. If we repeat the same experiments by using the panel signatures $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ that were obtained by averaging only R panel center pixels in the first and second columns for spectral unmixing, Figures 17.23 and 17.24 show the unmixed results of the 19 R panel pixels via abundance fractional maps produced by three SLSMA methods, LSOSP, NCLS, and FCLS where the results using four BKG signatures (grass, tree, road, and interferer) obtained by Figure 17.20 were significantly improved compared to that by using three BKG signatures (grass, tree, and road), specifically, LSOSP, and NCLS.

If we compare the results in Figures 17.23 and 17.24 to their counterparts, Figures 17.21 and 17.22, it showed that the best SLSMA result was the one produced by NCLS using R panel pixels in the two first columns and four BKG signatures in Figure 17.24(b). Other than that all the results were comparable. For further quantification comparison, Table 17.4 tabulates unmixed abundance fractions of 19 panel pixels in Figures 17.23 and 17.24 produced by LSOSP, NCLS, and FCLS using five panel signatures obtained by averaging all R pixels in the first two columns in Figure 1.15(b) and three BKG signatures and four BKG signatures, respectively, where each entry in Table 17.4 has two values with the one before and the one after "/" indicating the results produced by using three BKG signatures (grass, tree, and road) and four BKG signatures (grass, tree, road, and interferer), respectively.

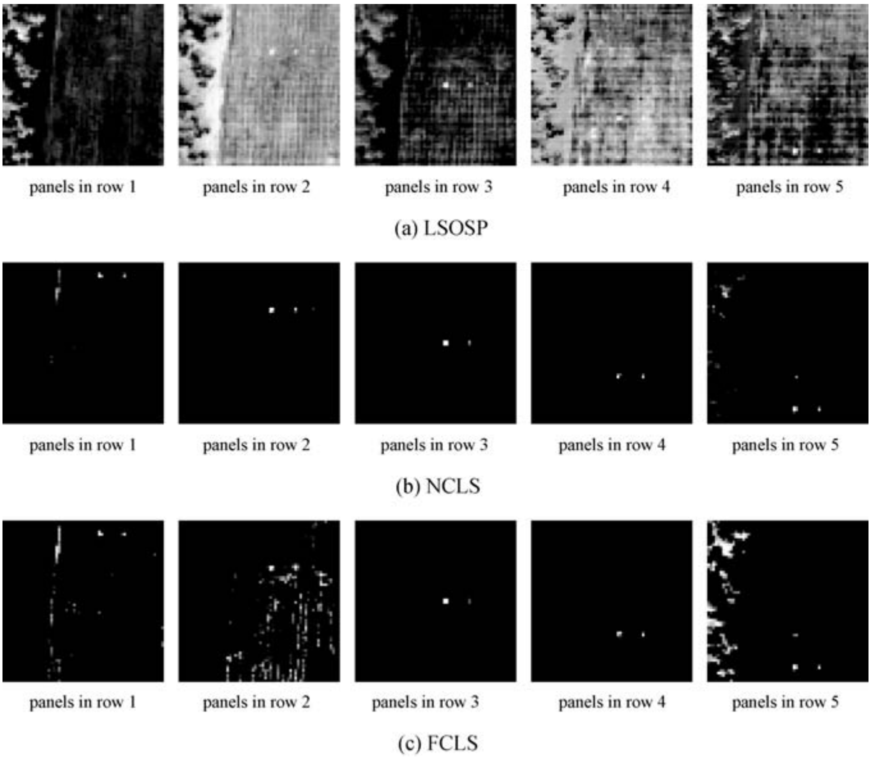
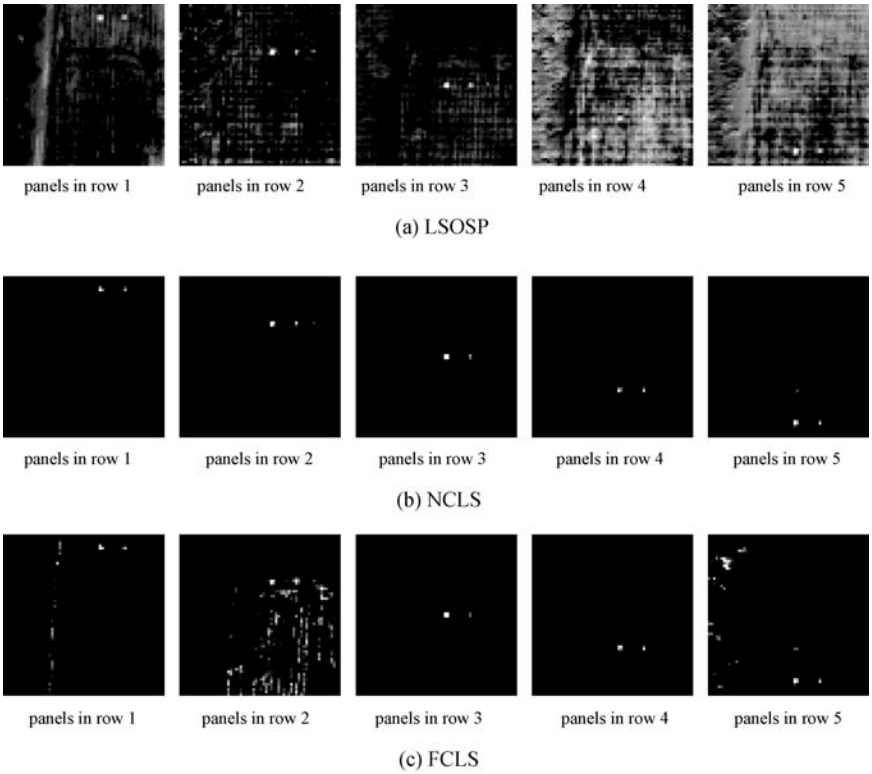


Figure 17.23 Abundance fractional maps of 15 panels with 19 panel pixels produced by LSOSP, NCLS, and FCLS using five panel signatures obtained by averaging R panel pixels in the first two columns in Figure 1.15 (b) and 3 BKG signatures, grass, tree, and road obtained by marked areas inn Figure 17.20.

In comparison with Table 17.3 where the FCLS-unmixed abundance fractions of panel pixels p_{411} and p_{412} were zeros they were now corrected and no longer 0. Despite that the provided ground truth p_{411} and p_{412} were the panel center pixels, from our extensive experience with the HYDICE scene, they were in fact not as pure pixels of 100% abundance purity as we expected. As a result, even though NCLS is partially abundance-constrained, it was very comparable to the fully abundance-constrained FCLS in terms of abundance unmixing. Nevertheless, both performed significantly better than the abundance-unconstrained LSOSP.

Now if we further compare the results in Table 17.4 against that in Table 17.3, it apparently shows that using contaminated or inaccurate prior knowledge may result in significant distortion in quantification of abundance fractions. Furthermore, comparing the results in Tables 17.1 and 17.2 against those in Tables 17.3 and 17.4, USLMA outperformed SLSMA significantly. These experiments further demonstrated two facts. One is that SLSMA was effective only if the prior knowledge was accurate such as the synthetic image experiments conducted for TI2 and TE2 scenarios in Section 17.4. Unfortunately, this may not be true when it comes to real world applications where true target knowledge is generally difficult to obtain, if not impossible. Even in the case that prior target knowledge is available, it may not be reliable due to many unknown signal sources that may contaminate the knowledge. This leads to the second fact that to avoid using unreliable prior knowledge ULSMA certainly provides a better alternative to SLSMA.



(a) LSOSP

(b) NCLS

(c) FCLS

Figure 17.24 Abundance fractional maps of 15 panels with 19 panel pixels produced by LSOSP, NCLS, and FCLS using five panel signatures obtained by averaging R panel pixels in the first two columns in Figure 1.15 (b) and 4 BKG signatures, grass, tree, road, and interferer obtained by marked areas inn Figure 17.20.

17.6 ULSMA Versus Endmember Extraction

Early attempts of performing ULSMA have been focused on simultaneous implementation of endmember selection and LSMA such as combining PCA to determine their purity of selected endmembers, using convex geometry of a simplex to fit data to select endmembers (Boardman, 1993, 1994; Boardman et al. 1995). In addition, some efforts such as multiple endmember spectral mixture analysis in (Roberts et al., 1998; Dennison and Roberts, 2003) and endmember bundles in (Bates et al., 2000) were also proposed to deal with spectral variations of endmembers to be selected. Most recently, the concept of virtual endmembers (VEs) was also introduced in a modified spectral mixture analysis (Tompkins et al., 1997) for endmember selection in such a way that the VEs are those minimizing root-mean-square-error subject to user-specified constraints. The VEs were further explored for endmember selection in an optical real-time adaptive spectral identification system (ORASIS) developed in Bowles and Gilles (2007). One major issue arising in these approaches is unavailability of prior knowledge about how many endmember needed to be selected in the first place. Consequently, they did not perform endmember extraction but rather endmember selection. Specifically, in their approaches endmember selection and linear spectral unmixing must be implemented simultaneously where a prescribed threshold or physical constraints should be imposed to determine when the entire process must be terminated. The LS-based LSMA/CA-based ULSMA

Table 17.4 Estimated abundance fractions of 19 panel pixels in Figures 17.23 and 17.24 produced by LSOSP, NCLS, and FCLS using five panel signatures obtained by averaging all R pixels in the first two columns using (three BKG signatures/four BKG signatures)

	LSOSP	NCLS	FCLS
p_{11}	1.2251/1.2635	0.9554/0.9550	0.7617/0.7617
p_{12}	0.7749/0.7365	0.6508/0.6226	0.6051/0.5369
p_{13}	0.4793/0.4053	0.1536/0.1302	0.1609/0.0075
p_{211}	1.0661/1.0853	0.9929/0.9929	0.9091/0.9091
p_{221}	1.0992/1.1652	0.9530/0.9530	0.7883/0.7883
p_{22}	0.8346/0.7495	0.8130/0.8130	0.8245/0.8245
p_{23}	0.5329/0.4578	0.4018/0.4018	0.4239/0.4239
p_{311}	1.0495/1.0584	0.9204/0.9139	0.9228/0.9136
p_{321}	1.2749/1.2404	0.9560/0.9292	0.9058/0.9025
p_{32}	0.6757/0.7014	0.4699/0.4699	0.4478/0.4478
p_{33}	0.3489/0.3756	0.2072/0.2072	0.2105/0.2105
p_{411}	1.0113/1.0532	0.9198/0.9104	0.5094/0.5094
p_{412}	0.9555/0.9507	0.3053/0.4329	0.3540/0.4378
p_{42}	1.0332/0.9962	0.7862/0.7862	0.7574/0.7574
p_{43}	0.3181/0.2684	0.1896/0.1896	0.1572/0.1572
p_{511}	0.9419/0.9531/	0.8304/0.8304	0.8304/0.8304
p_{521}	1.1901/1.1738	1.0628/1.0295	1.0000/1.0000
p_{52}	0.8680/0.8731	0.9354/0.9354	0.9353/0.9353
p_{53}	0.2022/0.2303	0.1596/0.1596	0.1374/0.1374

presented in this chapter provide rather different approaches by breaking up these two simultaneous processes into two separate processes in sequence. This is because LS-based LSMA/CA-based ULSMA can use VD to first determine how many signatures must be generated to form a linear mixing model for LSMA and then implement LS-UVSFA/CA-UVSFA to extract a desired set of signatures that are further used to unmix data to perform spectral unmixing. This may be the main reason that the term of *endmember selection* instead of *endmember extraction* was used in the references (Boardman et al., 1995; Boardman, 1993, 1994; Roberts et al., 1998; Bates et al., 2000; Dennison and Roberts, 2003; Tompkins et al., 1997; Bowles and Gilles, 2007). Furthermore, in order to make a distinction between the *signature finding* used in our proposed LS-UVSFA/CA-UVSFA and the VEs selected in Tompkins et al. (1997) and Bowles and Gilles (2007), the term of virtual signatures (VSs) is introduced in this chapter for clarity. The VSs found by LS-UVSFA and CA-UVSFA are quite different from (virtual) endmembers produced by endmember selection. Firstly, the VSs are not necessarily pure as opposed to endmembers that are assumed to be pure signatures. Secondly, comparing to criteria used to find endmember such as minimal simplex volume (Craig, 1994) and maximal simplex volume (Winter, 1999) used to find a simplex with minimal/maximal volume, orthogonal projection to compute pixel purity index (PPI) (Boardman, 1995), the criterion used by LS-UVSFA/CA-UVSFA to find VSs is based on IBSI(S) where only the data sample vectors in the sample pool S are used to calculate IBSI(S). One good representative example is the RX detector developed by Reed and Yu (1990) where the sample size S is the entire image data where it uses IBSI(S) to find a whitening matrix that is the inverse of the sample auto-covariance matrix formed by all data sample vectors. As a consequence, the VSs found by the RX detector are actually anomalies that are not necessarily endmembers. So, it is important to realize the difference between LS-UVSFA/CA-UVSFA and endmember selection. Although LS-UVSFA and CA-UVSFA do not necessarily perform endmember selection, they can be considered as endmember extraction algorithms

(EEAs) because LS-UVSFA/CA-UVSFA indeed can also find endmembers in most cases. Thirdly, the VSs found by LS-UVSFA/CA-UVSFA are categorized by two distinct classes of virtual signatures, BKG class and target class, where the VSs in the BKG class are generally mixed as opposed to the VSs in the target class that are usually pure. With this interpretation the VSs found by LS-UVSFA/CA-UVSFA in the target class are essentially those found by EEAs. It should be noted that such a BKG/target class-dichotomy is not new. A similar approach, called hierarchical foreground and background analysis, was also proposed (Pinzon et al. 1998), where a series of weighting vectors can be derived sequentially to simultaneously extract important discriminant features to detect leaf anatomy and chemical concentration at different levels from the spectral information.

It is a common sense that endmember extraction is closely tied with linear spectral unmixing. Nevertheless, they are completely different techniques. Endmember extraction finds all signatures assumed to be present in the data $\{\mathbf{e}_j\}_{j=1}^{\tilde{p}}$. It is generally carried out in an unsupervised manner. Therefore, two issues significantly affect its performance: (1) the number of endmembers and (2) finding these true endmembers. On the other hand, the linear spectral unmixing assumes that data can be best represented by a set of p signatures, $\{\mathbf{m}_j\}_{j=1}^{p}$ in a linear form from which a data sample vector can be unmixed via these signatures. Unlike SLSMA that requires prior knowledge of these signatures $\{\mathbf{m}_j\}_{j=1}^{p}$ ULSMA must rely on an unsupervised means to find these signatures $\{\mathbf{m}_j\}_{j=1}^{p}$. So, the value of p and the set of signatures, $\{\mathbf{m}_j\}_{j=1}^{p}$ have significant impact on unmixing performance. On many occasions in the past, research efforts have made an assumption that $\tilde{p} = p$ and $\{\mathbf{e}_j\}_{j=1}^{\tilde{p}} = \{\mathbf{m}_j\}_{j=1}^{p}$. As a result, a general practice is to use an endmember extraction algorithm to extract a set of potential endmember candidates that can be further verified by linear spectral unmixing and these two processes must be implemented simultaneously or iteratively. The approach of this type is referred to as EEA + LSMA that implements an EEA and LSMA simultaneously to perform ULSMA. One good representative falling in this category is works by Winter (1999a, 1999b, 2004) who developed an algorithm, called N-finder algorithm (N-FINDR), to extract potential endmembers and then used spectral unmixing to determine endmembers. Unfortunately, being able to do so, N-FINDR must assume that the number of endmembers is known *a priori*. But this issue was never addressed in Winter (1999a, 1999b, 2004) and was carried on a trial-and-error basis by simultaneously performing endmember selection and linear spectral unmixing as was done by those proposed in (Boardman et al., 1995; Boardman, 1993, 1994; Roberts et al., 1998; Bates et al., 2000; Dennison and Roberts, 2003; Tompkins et al., 1997; Bowles and Gilles, 2007). This explains that different terms of endmember selection and endmember determination were used (Tompkins et al., 1997; Bowles and Gilles, 2007) and Winter (1999a, 1999b, 2004), respectively, because spectral unmixing was involved to select or determine the endmembers. As a matter of fact, this is generally not true in real-world problems where the true endmembers may not be present in the data in which case EEAs may not be effective in these applications due to the fact that the signatures used by LSMA to unmix data are not necessarily endmembers. This will be demonstrated in the following experiments. LS-ULSMA and CA-ULSMA presented in this chapter are developed exactly for this cause where EEA is replaced with LS-UVSFA and CA-UVSFA.

Finally, in order to further substantiate the utility of LS-ULSMA and CA-ULSMA the HYDICE experiments were also conducted by N-FINDR + LSMA that implemented N-FINDR to extract nine endmembers to form a linear mixing model, which was further used by the LSMA to unmix the data. Since N-FINDR requires dimensionality reduction, three techniques, PCA, maximum noise fraction (MNF), and ICA, were used for this purpose. Figure 17.25 shows nine endmembers extracted by N-FINDR with data dimensionality reduced by PCA, MNF, and ICA where only

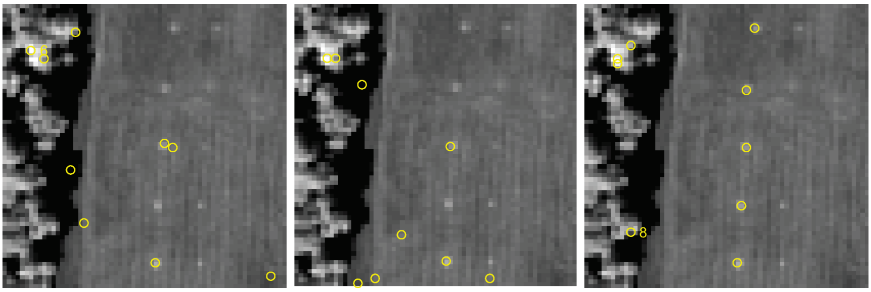

(a) PCA (b) MNF (c) ICA

Figure 17.25 Nine endmembers extracted by N-FINDR using PCA, MNF, and ICA to perform data dimensionality.

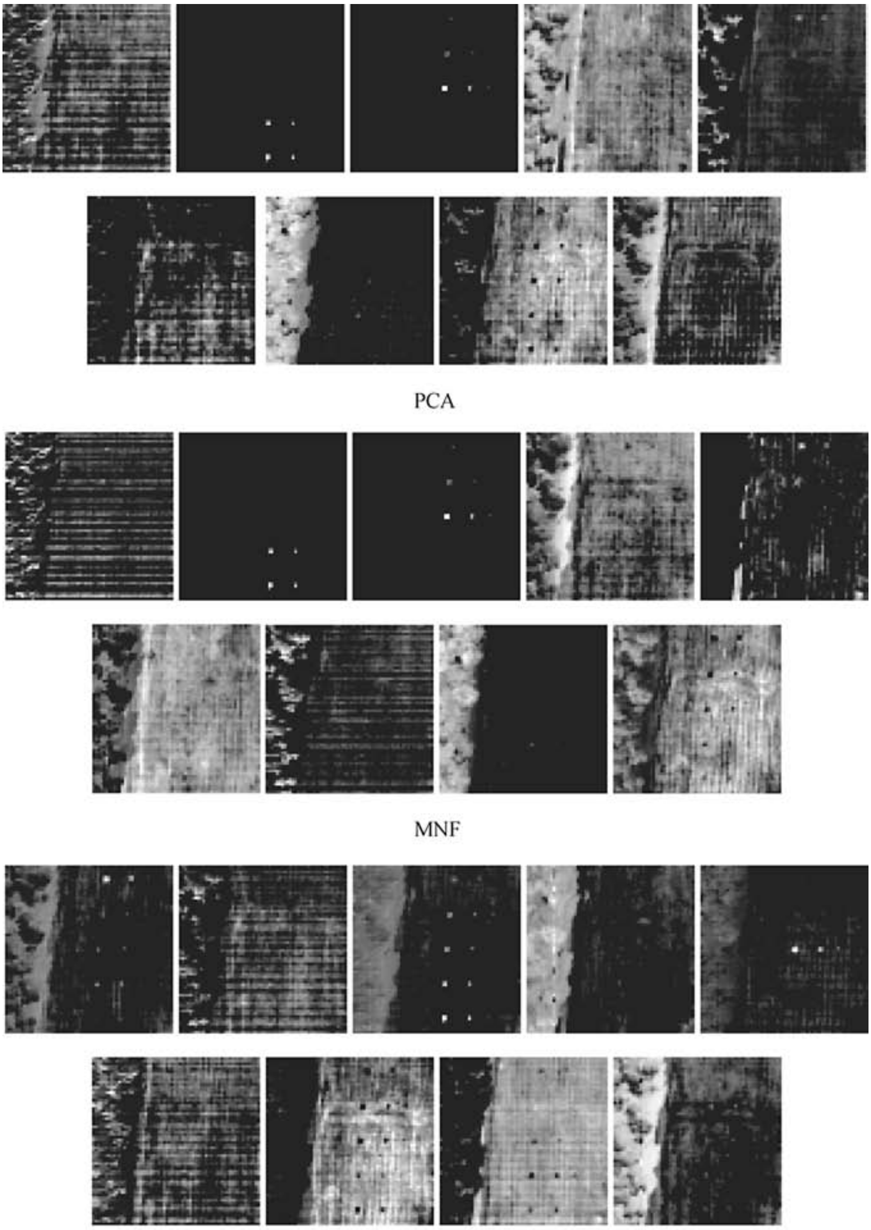
N-FINDR with ICA could successfully extract all the five panel signatures p_{11}, p_{211}, p_{311}, p_{411}, p_{511}, while PCA and MNF could only extract the same two-panel signatures, p_{311}, p_{511}.

Figure 17.26 shows the unmixed results of LSOSP, NCLS, and FCLS using the nine endmembers obtained in Figure 17.25 to form a linear mixing model for spectral unmixing where the orders of nine abundance fractional maps are arranged according to the nine extracted endmembers in a top-to-down and left-to-right manner.

Based on visual assessment of Figure 17.26, PCA and ICA were the worst and best dimensionality reduction techniques, respectively. In addition, the worst and the best unmixed results seemed to be those produced by LSOSP and FCLS, respectively, while NCLS also performed comparably to FCLS. Comparing the results in Figure 17.26 to CA-based ULSMA-unmixed results in Figure 17.19, it was apparent that CA-LSMA performed significantly better than N-FINDR + LSMA. The results in Figure 17.26 only provided qualitative analysis. In order to further quantify the 19 R panel pixels in Figure 17.26, Table 17.5 further tabulates their unmixed abundance fractions where the second column provides the assumed ground truth and the least squares error was calculated based on the sum of squared errors of the19 R panel pixels.

Interestingly, if we compare the results in Table 17.5 to the visual inspection of the results in Figure 17.26, the quantifications of 19 R panel pixels by LSOSP in comparison with NCLS and FCLS were not as bad as they are visualized in Figure 17.26. This is because the results in Table 17.5 did not show background suppression resulting from other signatures. This evidence indicated that both qualitative and quantitative analyses were necessary to provide full assessment of performance.

A final concluding remark is noteworthy. According to extensive experiments the best performance produced by various versions of N-FINDR except the one using ICA to perform dimensionality reduction could only extract three endmembers rather than five endmembers provided by the ground truth in Figure 1.15(b). The reason for this is because the materials made for the panels in second and third rows are the same fabric and the two panel signatures, $\mathbf{p}_2$ and $\mathbf{p}_3$ used to specify panel pixels in these two rows are therefore very similar. In this case, when $\mathbf{p}_3$ was extracted as an endmember and the $\mathbf{p}_2$ would be considered as a signature variation from $\mathbf{p}_3$ in which case $\mathbf{p}_3$ was not a pure signature. As a result, only one endmember in the third row was extracted to represent these two panel signatures. Similarly, an endmember in the fifth row was extracted to represent the two panel signatures, $\mathbf{p}_4$ and $\mathbf{p}_5$ that were used to specify panel pixels in the fourth and fifth rows. As a result, the panel pixels in row 4 in Figure 17.26 were extracted when p_{521} was extracted as an



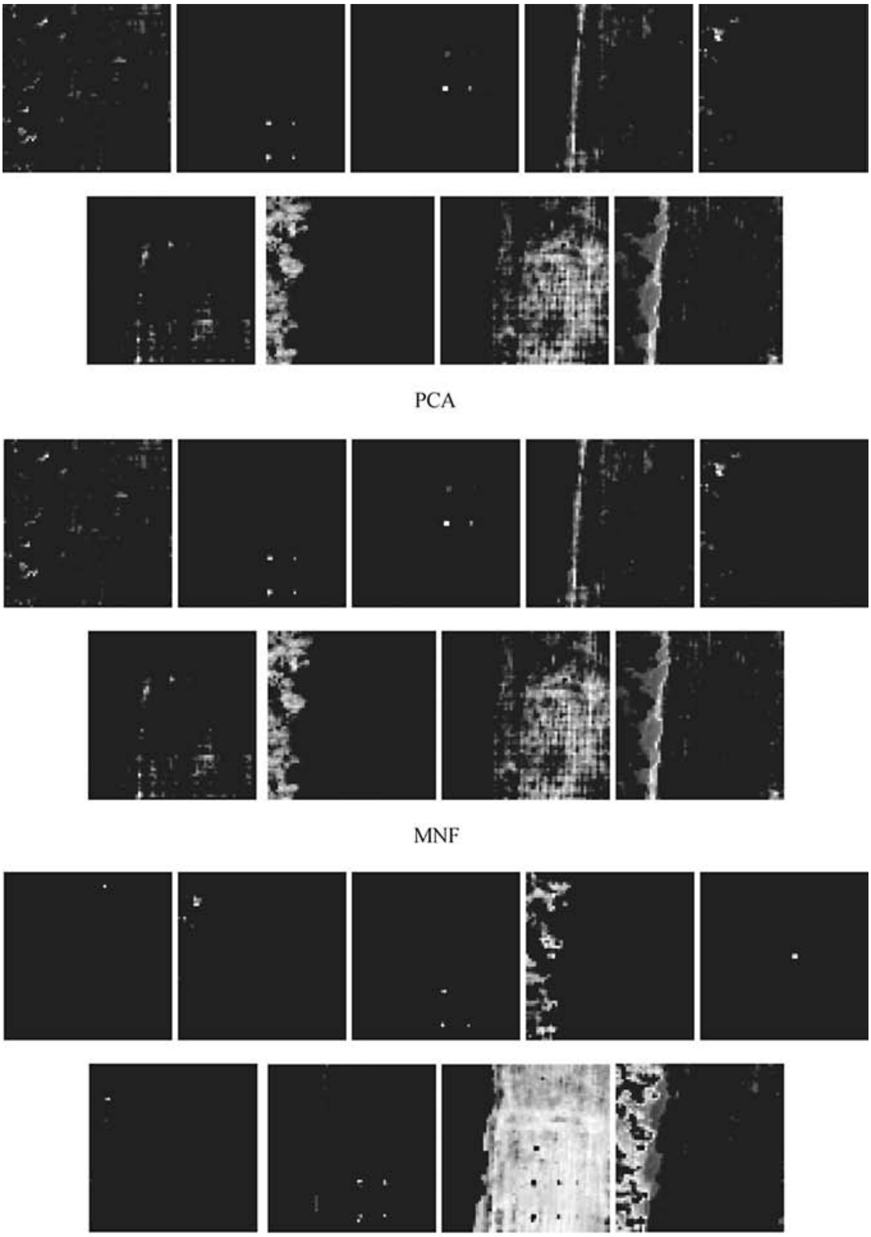
PCA

MNF

ICA

(a) Nine abundance fraction maps generated by LSOSP

Figure 17.26 Unmixed results of 15 panels with 19 panel pixels by LSOSP, NCLS, and FCLS using nine N-FINDR found endmembers for unmixing.



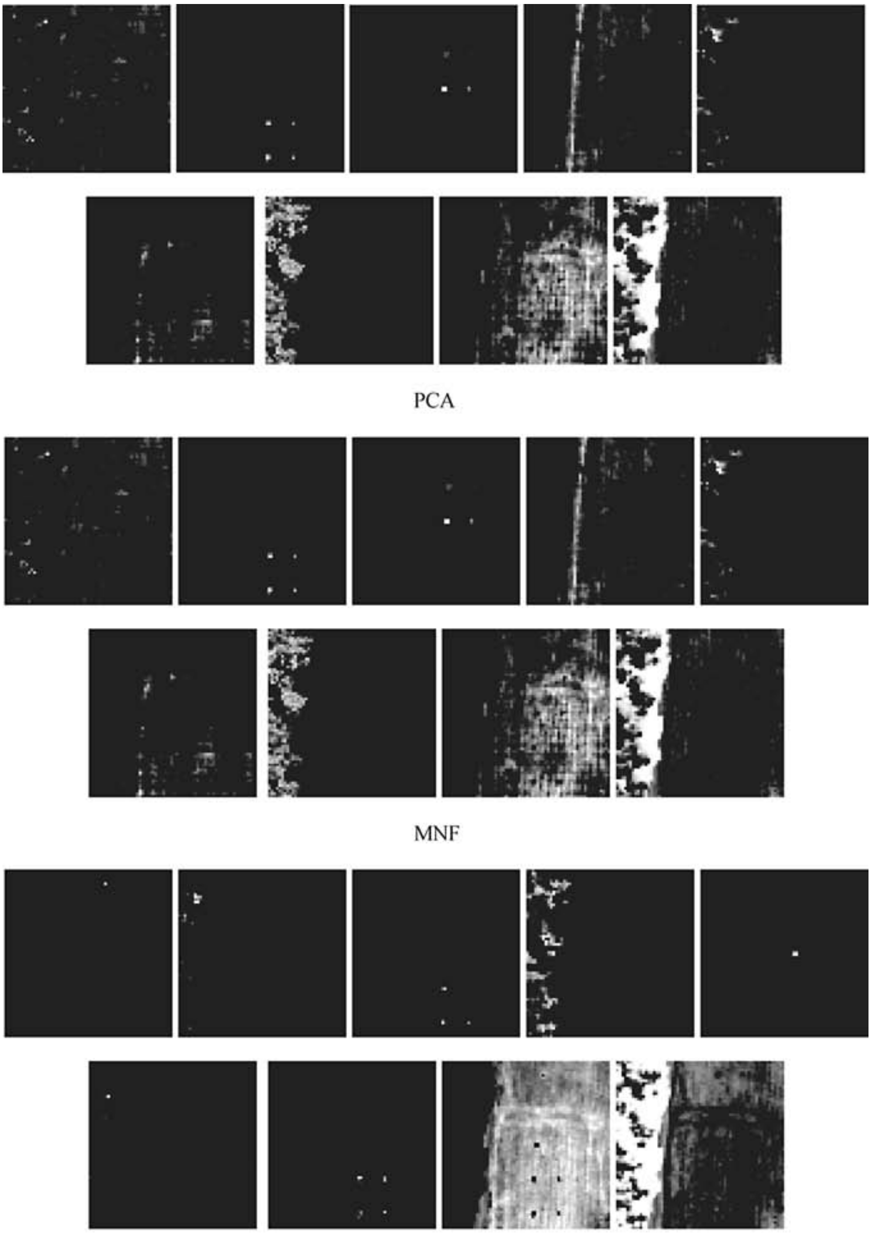
PCA

MNF

ICA

(b) Nine abundance fraction maps generated by NCLS

Figure 17.26 (*Continued*)



PCA

MNF

ICA

(c) Nine abundance fraction maps generated by FCLS

Figure 17.26 (*Continued*)

endmember to be used to unmix panel pixels in row 5, while the panel pixels in row 2 were also extracted when p_{312} was extracted as an endmember to be used to unmix panel pixels in row 3. Interestingly, in order for N-FINDR along with its variants to be able to five endmembers ICA must be used to perform dimensionality reduction due to the fact that the main strength of ICA is

Table 17.5 Quantification results of abundance fractions of 19 panels estimated by LSMA using nine endmembers generated by N-FINDR with three different dimensionality reduction techniques, PCA, MNF, and ICA

Panel pixels	Ground truth	LSOSP			NCLS			FCLS		
		PCA	MNF	ICA	PCA	MNF	ICA	PCA	MNF	ICA
p_{11}	1	0.5246	0.8123	1.0000	0.3129	0.3526	1.0000	0.3118	0.3329	1.0000
p_{12}	1	0.3088	0.3838	0.4529	0.2793	0.3001	0.5285	0.2293	0.3142	0.5273
p_{13}	0.4	0.5217	0.4108	0.7122	0.3386	0.3598	0.7238	0.3679	0.3467	0.7053
p_{211}	1	0.3974	0.4557	0.8465	0.4283	0.4274	0.9472	0.4247	0.4233	0.9384
p_{221}	1	0.4041	0.5010	1.0000	0.4386	0.4376	1.0000	0.4348	0.4285	1.0000
p_{22}	1	0.3308	0.3329	0.6330	0.3604	0.3409	0.7995	0.3519	0.3430	0.7991
p_{23}	0.4	0.3513	0.3316	0.7450	0.2953	0.3762	0.8182	0.2650	0.4231	0.7455
p_{311}	1	0.9727	0.9638	0.8538	0.9180	0.9254	0.8613	0.8991	0.9271	0.8373
p_{321}	1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
p_{32}	1	0.5596	0.5569	0.7250	0.5197	0.5164	0.9622	0.4929	0.5127	0.9308
p_{33}	0.4	0.3691	0.3474	0.7612	0.3837	0.3681	0.8808	0.4037	0.3673	0.7572
p_{411}	1	0.6730	0.6864	0.9610	0.7028	0.7123	0.5755	0.7404	0.7379	0.5690
p_{412}	1	0.7486	0.7669	0.5532	0.7745	0.7819	0.7592	0.7907	0.8065	0.7535
p_{42}	1	0.5723	0.5982	0.8705	0.6043	0.6158	0.6804	0.6396	0.6408	0.6925
p_{43}	0.4	0.2543	0.3715	1.2117	0.2281	0.3792	0.6064	0.2774	0.3285	0.6025
p_{511}	1	0.7303	0.7265	0.6291	0.7108	0.7213	0.5093	0.7141	0.7248	0.5021
p_{521}	1	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
p_{52}	1	0.7897	0.7884	0.6416	0.7765	0.7756	0.6649	0.7707	0.7740	0.6204
p_{53}	0.4	0.2723	0.2872	0.9795	0.2373	0.3181	0.6634	0.2832	0.2698	0.6595
Sum of squared errors		**2.5900**	**2.0577**	**2.3801**	**2.7806**	**2.6434**	**1.6023**	**2.8286**	**2.6374**	**1.4806**

blind source separation. As a consequence, N-FINDR was capable of finding five endmembers as shown in Figure 17.25(c) to specify all the five panel signatures, $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$.

17.7 Conclusions

The performance of LSMA is completely determined by the number of signatures, p and signatures $\{\mathbf{m}_j\}_{j=1}^{p}$ used to form a linear mixing model to unmix data sample vectors. Unfortunately, in real applications none of these two pieces of information is known accurately in advance. So, a key to success in LSMA is to find an appropriate signature matrix $\mathbf{M}$ to form a linear mixing model $\mathbf{r} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n}$ in (2.75) where $\mathbf{r}$ is an image pixel vector and $\mathbf{n}$ is a model correction term. In supervised LSMA (SLSMA), this matrix $\mathbf{M}$ is assumed to be known *a priori*. However, when it comes to ULSMA the knowledge of the signature matrix $\mathbf{M}$ is not available and must be obtained directly from the data. The two unsupervised approaches, LS-UVSFA in Section 17.2 and CA-UVSFA in Section 17.3, provide a means of finding such an unsupervised signature matrix $\mathbf{M}$ for ULSMA. Since the signatures found in the unsupervised signature matrix, $\mathbf{M}$, are real data sample vectors and may not be pure as true endmembers assumed in SLSMA, they are called virtual signatures (VSs) for their distinction from true endmembers. So, the signature matrix $\mathbf{M}$ formed by VSs is also referred to as VS matrix. The performance of ULSMA is completely determined by two factors, the number of VSs, p, and the VSs used to form a linear mixing model, both of which are assumed to be known in SLSMA. Secondly, ULSMA generally outperforms SLSMA in cases that many unknown material signatures that cannot be identified by visual inspection or prior knowledge can now be found by LS-UVSFA/CA-UVSFA. Thirdly, the signatures used to form a linear mixing model for ULSMA are real data sample vectors. Accordingly, they are generally not real endmembers and can be in any form such as subsample vectors and mixed sample vectors. This is the reason that such data sample vectors are referred to as VSs for ULSMA to distinguish from true endmembers used for SLSMA. Last but not least, the performance of LSMA is generally evaluated by unmixed results of signatures that are used to form a linear mixing model either qualitatively or quantitatively. To this end, over the past years the signatures used for LSMA to perform unmixing as SLSMA are usually those in which users are interested. When it is extended to ULSMA the same logic is also applied where endmembers are assumed to be the signatures of interest. However, as demonstrated in the experiments conducted in Section 17.6 this may not be realistic or applicable for real-world applications where mixed BKG signatures that are generally not signatures of interest are also crucial for LSMA to perform background suppression. The ability of LSMA in background suppression has been often overlooked and it can be as important as signatures of interest such as endmembers. Unfortunately, finding appropriate BKG signatures is not a trivial matter. The LS-UVSFA and CA-UVSFA presented in this chapter are primarily designed for this purpose, both of which find a desired set of VSs that includes target and BKG signatures. Whether these VSs are pure or mixed is not of major concern for ULSMA.

As a final note, despite the fact that many efforts have been made to determine the number of endmembers (Kosaka et al., 2005; Nascimento and Dias, 2005; Eches et al., 2010; Cawse et al., 2010; Zare and Gader, 2007, Broadwater and Banerjee, 2009), there is no specific technique developed for determining and finding the number of true endmembers, both of which are actually two separate issues. The VD developed in Chapter 5 is particularly developed for the purpose of determining the number of spectrally distinct signatures not endmembers. Although, VD generally overestimates the number of endmembers as demonstrated in Chang et al. (2010) VD was shown indeed a good estimate for the number of signatures used to form a linear mixing model used by LSMA. For this reason VD has been used across the board to determine the number of signatures required by ULSMA.

18

Pixel Extraction and Information

Because of very high spectral resolution provided by a hyperspectral imager, a single hyperspectral image pixel vector can now unveil subtle and crucial information for data analysis that a traditional image pure-based pixel cannot. Unfortunately, much of such rich information is presumably unknown and cannot be identified *a priori*. This chapter investigates the issue of how to extract pixel information from an exploitation viewpoint. In order to facilitate pixel information analysis, four types of pixels of interest, pure pixel, mixed pixel, anomalous pixel, and homogeneous pixel, are defined. A pure pixel is a pixel whose spectral signature is completely represented by a single-material substance as opposed to a mixed pixel whose spectral signature is composed of more than one material substance. A homogeneous pixel can be defined as a pixel whose spectral signature remains nearly constant subject to small variations within its surroundings in contrast to an anomalous pixel whose signature is spectrally distinct from the signatures of its neighboring pixels. Therefore, a homogeneous pixel can be considered opposite to an anomalous pixel, while a pure pixel is an opposite of a mixed pixel. On one end, pure and mixed pixels can be dealt with from a single-pixel point of view. On the other end, analysis of homogeneous and anomalous pixels must take into account the surrounding pixels within their neighborhoods, that is, their neighboring pixels. Thus, a homogeneous pixel or an anomalous pixel can be either a pure or a mixed pixel. This chapter investigates and designs various scenarios to explore differences among these four types of pixels for comparative analysis.

18.1 Introduction

In traditional two-dimensional (2D) image processing, an image pixel is specified by its intensity and represented by a single value of the gray scale. In hyperspectral image processing, a hyperspectral image is an image cube formed by stacking 2D spectral images acquired by a range of hundreds of spectral channels where a hyperspectral image pixel is actually a column vector, of which each vector component is an image pixel acquired by a specific wavelength. To simplify our discussion, the term "pixel" will be used instead of "pixel vector." Therefore, one great challenge in hyperspectral data exploitation is analysis of information extracted from a hyperspectral image pixel specified by hundreds of spectral channels. However, how much pixel information can be extracted is also determined by what algorithm to be used for information extraction, such as algorithms in PART II (Chapters 7–11) developed for endmember extraction and algorithms in PART III (Chapters 12–17) developed for target detection and classification. In other words, what we are interested in is, "Does an algorithm really do what it is designed for?" For example, in endmember

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

extraction, do pixel purity index (PPI) in Section 7.2.1 and N-finder algorithm (N-FINDR) in Section 7.2.4 really extract pure signatures as they are designed for? In unsupervised target detection, do algorithms, such as automatic target generation process (ATGP) developed by Ren and Chang (2003) in Section 8.5.1 and unsupervised fully constrained least-squares (UFCLS) algorithm developed by Heinz and Chang (2001) in Section 8.5.3, really perform what they are asked for? In anomaly detection, does the RX algorithm developed by Reed and Yu (1990) really find anomalies or something else? Specifically, "what pixel information does an exploitation algorithm really extract?" and "does it really do what it claims to do?" It turns out that answers to these questions are more complicated than what we had thought, as will be demonstrated by experiments in this chapter. In order to facilitate our discussions, four different types of pixels in accordance with their spatial or spectral properties, pure pixel, mixed pixel, homogenous pixel, and anomalous pixel, are introduced for pixel information analysis.

18.2 Four Types of Pixels

According to the definition given by Schowengerdt (1997), an endmember is an idealized pure signature for a spectral class. Therefore, an endmember is, in general, not a pixel. It is a spectral signature that is completely specified by the spectrum of a single-material substance. Accordingly, a pixel is pure if its spectral signature is an endmember. In other words, a pixel whose spectral signature has 100% of purity formed of a single-material substance is a pure pixel, which can also be referred to as an endmember pixel. In this chapter, the endmember pixel and pure pixel will be used interchangeably, whichever is more appropriate for explanation. To the contrary, a mixed pixel is a pixel whose spectral signature is not an endmember. Instead, it is composed of more than one material substance. With this interpretation, when an endmember extraction algorithm (EEA) is implemented, it is expected that signatures or pixels extracted are supposed to be pure signatures or pure pixels, that is, endmember pixels. But, is this really the case in practical applications? In addition to the pure pixel and mixed pixel described above, a concept of homogenous pixel is further introduced for pixel information analysis. A pixel is called a homogeneous pixel if its spectral signature is similar to the spectral signatures of the pixels in its surroundings subject to small deviations. In other words, all the pixels within a neighborhood of a homogenous pixel should have very close and similar spectral signatures to that of the homogenous pixel. This definition is derived from the concept of homogeneous regions used in image segmentation where a homogenous region is a data set consisting of pixels with very close gray-scale values in a tolerable range. An anomalous pixel is completely opposite to a homogeneous pixel and is defined as a pixel whose signature is spectrally distinct from the spectral signatures of its neighboring pixels. While pure pixels and mixed pixels can be analyzed solely by their spectral properties on a single-pixel basis, homogeneous pixels and anomalous pixels must take into account the neighboring pixels within their surroundings in addition to their spectral properties. It is interesting to note that a pure pixel can be a homogenous pixel or an anomalous pixel, so is a mixed pixel. Conversely, a homogeneous pixel and anomalous pixel can also be a pure pixel or a mixed pixel and vice versa. With these four types of pixels defined above, we can further evaluate various algorithms based on what type of pixel information they extract.

In order to demonstrate the utility of these four types of pixels, seven algorithms will be evaluated and compared, which can be categorized into three categories: EEAs, unsupervised target detection algorithms (UTDAs), and anomaly detection algorithms. As for EEAs, two commonly used algorithms, PPI and N-FINDR, and the automated morphological endmember extraction (AMEE) algorithm are considered. In the category of UTDAs, three algorithms, ATGP, UFCLS algorithm, and IEA algorithm, are included for comparison. The third category contains only one widely used anomaly detection algorithm, the RX algorithm, which is also referred to as RX detector (RXD) or RX filter (RXF) in other chapters. These three terminologies have been used interchangeably in this book. The analysis of these seven algorithms will be conducted via extensive computer simulations, synthetic image and real image experiments. As will be demonstrated, various algorithms perform in very different ways in terms of pixel information extracted from the data.

18.3 Algorithms Selected to Extract Pixel Information

While our selection of algorithms for performance evaluation may be subjective, it is our desire to make this selection as representative as possible. Nevertheless, such a selection by no means claims to be complete.

In the first category of algorithms, we evaluate three EEAs developed in Chapter 7 for endmember extraction: PPI available in the Research Systems ENVI software, N-FINDR that has widely been used for endmember extraction, and AMEE which is the only algorithm in this category that makes use of spatial information for endmember extraction. As noted earlier, the N-FINDR implemented here is actually IN-FINDR for its iterative advantages. The second category of algorithms consists of algorithms developed for unsupervised target detection, which have been used to generate *a posteriori* knowledge for applications in supervised target detection and classification. Interestingly, they can also be used for endmember extraction, as demonstrated in Chapter 8. Three algorithms, such as ATGP in Section 8.5.1, UFCLS in Section 8.5.3, and IEA in Section 8.5.4, are of interest. In the third category of algorithms, we look into algorithms developed for anomaly detection. This is due to the fact that occurrence of pure pixels is considered rare. In this case, endmembers behave like anomalies; thus, they can be extracted by anomaly detection algorithms. Since the RX algorithm has widely been used for this purpose and many anomaly detection algorithms that are currently being used can be considered as its variants, it is selected to represent the class of anomaly detection algorithms.

18.4 Pixel Information Analysis via Synthetic Images

As mentioned above, the three categories of seven algorithms described in Section 18.3 represent different ways of generating target pixels, which can be characterized by the proposed four types of pixels. In this section, a comprehensive synthetic image-based study on pixel information analysis will be conducted to evaluate these seven algorithms on the basis of what types of pixels these algorithms really extract in terms of pure (or purest) pixels, mixed pixels, homogenous pixels, and anomalous pixels. The importance and significance of this study is to allow us to simulate various scenarios to evaluate subtle differences among the four different types of pixels, and to further explore the pixel information extracted by these three categories of algorithms for performance analysis. The reflectance spectra of five mineral spectra, alunite, buddingtonite, calcite, kaolinite, and muscovite, obtained from the USGS and shown in Figure 1.7 are first used for computer simulations.

A uniform background image with size of 20×20 pixels was simulated by 100% of the same mixed signature (50% alunite and 50% kaolinite). Next, three sets of 2×2 panels shown in Figure 18.1(a), that is, $\left\{p_{ij}^k\right\}_{i=1,j=1}^{2,2}$, $\left\{\tilde{p}_{ij}^k\right\}_{i=1,j=1}^{2,2}$, and $\left\{\bar{p}_{ij}^k\right\}_{i=1,j=1}^{2,2}$ for $k = 1, 2, 3$, were simulated by each of three signatures, buddingtonite, calcite, and muscovite, according to Table 18.1. The pixels were then implanted in the uniform background, as shown in Figure 18.1(b).

Table 18.2 lists what types of panel pixels can be found in the nine 2×2 panels in Figure 18.1 (a) based on our definitions of four types of pixels, where both "endmember" and "pure" signatures are used to emphasize the nature of pure signature simulated by the experiments, and P, M, H, and A indicate pure, mixed, homogeneous, and anomalous pixels, respectively.

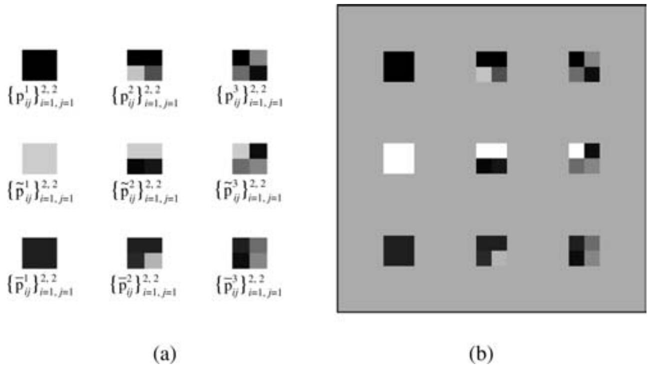

$\{p_{ij}^1\}_{i=1,j=1}^{2,2}$
$\{p_{ij}^2\}_{i=1,j=1}^{2,2}$
$\{p_{ij}^3\}_{i=1,j=1}^{2,2}$
$\{\tilde{p}_{ij}^1\}_{i=1,j=1}^{2,2}$
$\{\tilde{p}_{ij}^2\}_{i=1,j=1}^{2,2}$
$\{\tilde{p}_{ij}^3\}_{i=1,j=1}^{2,2}$
$\{\bar{p}_{ij}^1\}_{i=1,j=1}^{2,2}$
$\{\bar{p}_{ij}^2\}_{i=1,j=1}^{2,2}$
$\{\bar{p}_{ij}^3\}_{i=1,j=1}^{2,2}$
(a)
(b)

Figure 18.1 (a) Nine 2 × 2 panels simulated by buddingtonite, calcite, and muscovite; (b) synthetic image.

There are seven endmember pixels in each of three rows which are made by 100% pure buddingtonite, calcite, and muscovite signatures, $\left\{p_{ij}^1\right\}_{i=1,\,j=1}^{2,2}, p_{11}^2,\ p_{12}^2, p_{11}^3$; $\left\{\tilde{p}_{ij}^1\right\}_{i=1,\,j=1}^{2,2}, \tilde{p}_{11}^2,\ \tilde{p}_{12}^2, \tilde{p}_{11}^3$ and $\left\{\bar{p}_{ij}^1\right\}_{i=1,\,j=1}^{2,2}, \bar{p}_{11}^2,\ \bar{p}_{12}^2, \bar{p}_{11}^3$. All the background pixels are considered to be homogenous pixels as well as mixed pixels. The endmember pixels at the top left of the 2 × 2 panels in the third

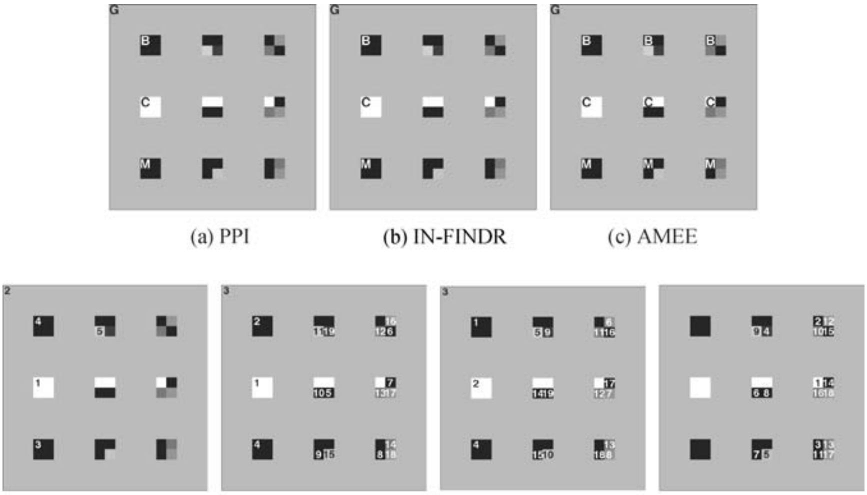
Table 18.1 Abundance fractions simulated for the panel pixels

Panel pixel	Buddingtonite (%)	Calcite (%)	Muscovite (%)
$\left\{p_{ij}^1\right\}_{i=1,\,j=1}^{2,2}, p_{11}^2, p_{12}^2, p_{11}^3$	100	0	0
p_{21}^2	0	75	25
p_{22}^2	0	25	75
p_{12}^3	0	50	50
p_{21}^3	50	50	0
p_{22}^3	50	0	50
$\left\{\tilde{p}_{ij}^1\right\}_{i=1,\,j=1}^{2,2}, \tilde{p}_{11}^2, \tilde{p}_{12}^2, \tilde{p}_{11}^3$	0	100	0
$\tilde{p}_{21}^2$	75	0	25
$\tilde{p}_{22}^2$	25	0	75
$\tilde{p}_{12}^3$	50	0	50
$\tilde{p}_{21}^3$	50	50	0
$\tilde{p}_{22}^3$	0	50	50
$\left\{\bar{p}_{ij}^1\right\}_{i=1,\,j=1}^{2,2}, \bar{p}_{11}^2, \bar{p}_{12}^2, \bar{p}_{11}^3$	0	0	100
$\bar{p}_{21}^2$	75	25	0
$\bar{p}_{22}^2$	25	75	0
$\bar{p}_{12}^3$	50	50	0
$\bar{p}_{21}^3$	50	0	50
$\bar{p}_{22}^3$	0	50	50

Table 18.2 Types of panel pixels based on our definitions of four types of pixels

Panel pixel	Signature		Pixel			
	Pure/endmember	Mixed	P	M	H	A
$\left\{p_{ij}^1\right\}_{i=1,j=1}^{2,2}, p_{11}^2, p_{12}^2$	✓		✓		✓	
$p_{21}^2, p_{22}^2, p_{12}^3, p_{21}^3, p_{22}^3$		✓		✓	✓	
p_{11}^3	✓		✓			✓
$\left\{\tilde{p}_{ij}^1\right\}_{i=1,j=1}^{2,2}, \tilde{p}_{11}^2, \tilde{p}_{12}^2$	✓		✓		✓	
$\tilde{p}_{21}^2, \tilde{p}_{22}^2, \tilde{p}_{12}^3, \tilde{p}_{21}^3, \tilde{p}_{22}^3$		✓		✓	✓	
$\tilde{p}_{11}^3$	✓		✓			✓
$\left\{\bar{p}_{ij}^1\right\}_{i=1,j=1}^{2,2}, \bar{p}_{11}^2, \bar{p}_{12}^2$	✓		✓		✓	
$\bar{p}_{21}^2, \bar{p}_{22}^2, \bar{p}_{12}^3, \bar{p}_{21}^3, \bar{p}_{22}^3$		✓		✓	✓	
p_{11}^3	✓		✓			✓
Background pixels		✓	✓		✓	

column in Figure 18.2(b), p_{11}^3, $\tilde{p}_{11}^3$, $\bar{p}_{11}^3$, are also anomalous pixels because their signatures are spectrally distinct from the spectral signatures of their neighboring pixels. All other panel pixels are mixed pixels with various mixtures of different spectral signatures. According to the simulated synthetic image, there are 19 distinct signatures, of which there are only three 100% pure signatures and 16 mixed signatures.



(a) PPI (b) IN-FINDR (c) AMEE

(d) ATGP (e) UFCLS (f) IEA (g) RX algorithm

Figure 18.2 Results produced by the seven algorithms.

EXAMPLE 18.1

(Analysis of Four Types of Pixels)

Figure 18.2 shows the results produced by the seven algorithms that also revealed many intriguing findings. The numbers in all the figures are labeled according to the order in which the pixels were generated. First, the two popular EEAs, PPI, and IN-FINDR extracted three endmember pixels that represented three distinct pure signatures, buddingtonite (B), calcite (C), muscovite (M), and one background pixel (BKG) that represented the uniform background specified by a signature mixed by 50% of alunite and 50% of kaolinite that can be considered a homogeneous pixel. Both algorithms were terminated after these four pixels were generated. The third EEA, AMEE, extracted 10 pixels before it was terminated.

It is interesting to note that the 10 AMEE-generated pixels represented nine different panels and one background pixel, all of which are considered homogeneous pixels due to the spatial morphological process included in AMEE where homogeneous pixels of the same type were not extracted once the first pixel of its type was extracted. Also, it is interesting to note that the nine AMEE-extracted panel pixels were also endmember pixels. For instance, as shown in Figure 18.2(c), as long as the $\bar{p}^1_{11}$ at the upper left corner of the first panel in the third row was extracted, the three pixels $\bar{p}^1_{12}, \bar{p}^1_{21}$, $\bar{p}^1_{22}$ that are in the same panel as is $\bar{p}^1_{11}$ were not extracted since they are adjacent together and simulated by the same pure signature, muscovite that was used to simulate $\bar{p}^1_{11}$. The same phenomenon was repeated until nine endmember pixels in nine different panels were extracted. Finally, AMEE was terminated after it extracted a background pixel that represented another distinct type of signature mixed by two material substances, alunite and kaolinite. This is also because all the background pixels were homogeneous pixels that were simulated by the same signature. Therefore, as soon as a background pixel was extracted, there were no more pixels that represented distinct signatures.

Despite the fact that the background pixel was not a pure pixel, it extracted as an endmember pixel. This is because the background signature simulated by mixing 50% of alunite and 50% of kaolinite and was considered the pure signature of a mixture of 50% of alunite and 50% of kaolinite. This observation provided a very interesting insight into the definitions of pure pixel and endmember pixel. Since the background signature is a 50–50 even split mixture of two distinct signatures, it cannot be determined by either signature. In this case, the background signature was considered a new signature, which is actually a hybrid signature. If we interpret this hybrid background signature as a new type of pure signature, the background signature then became an endmember. This explains why all EEAs extract a background pixel as an endmember pixel (see Figure 18.2 (a)–(c)).

In contrast to AMEE, the performance of the RX algorithm is completely opposite. As shown in Figure 18.2(g), the RX algorithm extracted all 16 mixed pixels plus three pure pixels, p^3_{11}, $\tilde{p}^3_{11}$, $\bar{p}^3_{11}$, located at the upper-left corner of the third panel in each of the three rows since all the 19 extracted pixels represented spectrally distinct signatures in their surroundings and were considered anomalous pixels. The RX algorithm did not extract the other 18 endmember pixels or any background pixel because they were considered homogenous pixels.

On one hand, UFCLS and IEA, as shown in Figure 18.2(e) and (f), behaved like the RX algorithm, which also extracted all 16 mixed pixels. On the other hand, UFCLS and IEA performed differently from the RX algorithm in terms of extraction of different types of endmember pixels. The endmember pixels generated by the RX algorithm were located at the upper-left corner of the third panel in each of three rows that were considered anomalous pixels. To the contrary, the endmember pixels generated by UFCLS and IEA were located at the upper-left corner of the first panel in each of three rows that were considered homogeneous pixels. More interestingly, UFCLS and IEA generated one more pixel than the RX algorithm did. It was a background pixel that was also considered a homogeneous pixel.

Compared to the six other algorithms, ATGP in Figure 18.3(d) performed a little bit differently. It behaved like PPI and IN-FINDR in the sense that it extracted the three endmember pixels that represented three pure signatures and one mixed background signature. In addition, it also extracted one mixed pixel that both PPI and IN-FINDR did not extract. It was terminated at the fifth pixel due to warning of matrix singularity where the fifth pixel turned out to be a mixed pixel.

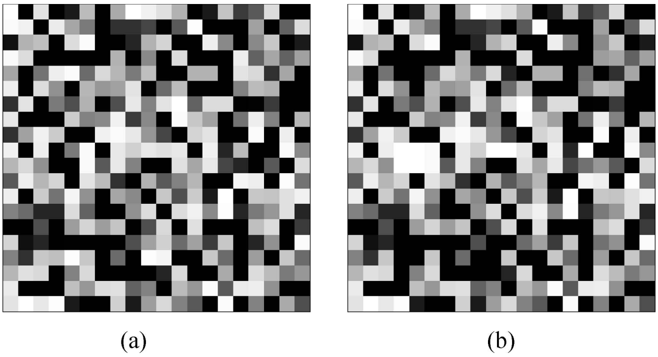


Figure 18.3 (a) Background image corrupted by Gaussian noise with SNR of 30:1; (b) synthetic image.

It should be noted that the EEAs generated all the desired endmember pixels simultaneously compared to other algorithms that generated pixels sequentially one at a time with the numbers indicating the order in which pixels were generated.

EXAMPLE 18.2

(Noise Effect)

The same simulated image used in Example 1 was also used for experiments except that the background image in Figure 18.1(b) was corrupted by Gaussian noise with SNR 30:1, as defined by Harsanyi and Chang (1994). Figure 18.3(a) shows a Gaussian noise corrupted background image, and Figure 18.3(b) shows a synthetic image resulting from implanting the nine panels in Figure 18.1(a) in the noisy background image in Figure 18.3(a).

This example was particularly designed to test how the seven considered algorithms performed in terms of extracting pixel information. Unlike Example 1, the synthetic image in this example was a noise-corrupted image. In this case, we did not have prior knowledge about how many endmembers, *p*, were present in the image. In order to determine this number *p*, we use virtual dimensionality (VD) developed in Chapter 5, where the Harsanyi–Farrand–Chang (HFC) and noise-whitened HFC (NWHFC) methods are used to determine VD. Table 18.3 tabulates the estimated VD based on various false-alarm probabilities.

In Table 18.3, the experiments with SNR = 20:1 and 10:1 are also included for comparison to select an appropriate value for VD. According to the table, it seems that 3 is an appropriate estimate of VD. In this case, only three endmembers or pixels were generated by the seven algorithms, as shown in Figure 18.4(a)–(g). From these figures, PPI and IN-FINDR extracted three endmember pixels that were specified by all the three distinct pure signatures compared to ATGP, UFCLS, and IEA, which extracted only two endmembers plus a background pixel. More interestingly, the three UTDAs performed quite differently in terms of endmember

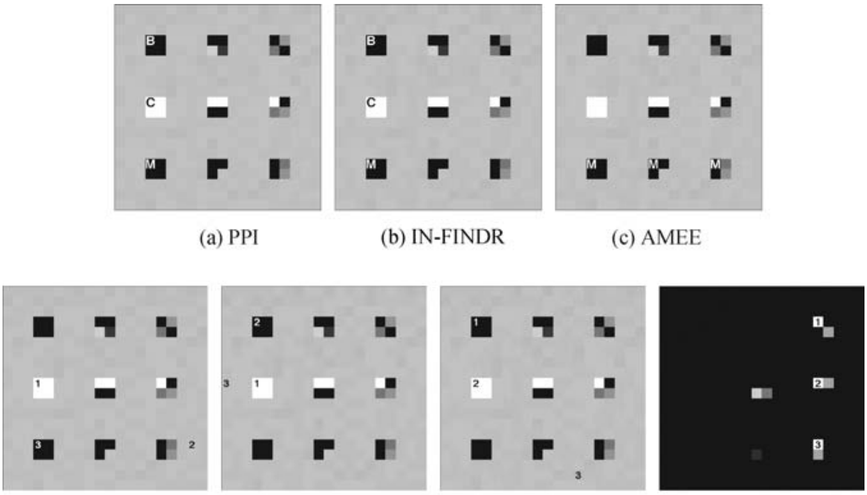
Table 18.3 VD estimated by the HFC and NWHFC methods for the synthetic image in Figure 1.3(c)

P_F	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
HFC (SNR = 30:1)	4	3	2	2	2
NWHFC (SNR = 30:1)	4	4	3	3	2
HFC (SNR = 20:1;10:1)	4	3	2	2	2
NWHFC (SNR = 20:1;10:1)	4	3	2	2	2

extraction and the orders of pixels extracted. While ATGP extracted two endmember pixels in the first panels of the second and third rows, UFCLS and IEA extracted the two endmember pixels in the first panels of the first and second rows instead. Even in the latter case, the order of two endmember pixels extracted by UFCLS and IEA was completely opposite. These experiments demonstrated one of most important and significant differences between an EEA and a UTDA. An EEA did exactly what it was asked to do to extract all the three endmember pixels that represented three distinct pure signatures. In contrast, a UTDA extracted most significant pixels in terms of spectrally distinct signatures, but not necessarily pure signatures. As a result, the three UTDAs extracted a background pixel that was considered a crucial and critical signature since it was used to simulate the entire background. Although the background was not a pure signature, it was indeed the most important signature in the image.

Compared to these five algorithms, AMEE and the RX algorithm also performed very differently. As in Example 1, AMEE extracted three endmember pixels in the three different panels in the same row (i.e., row 3). However, it missed all other two pure signatures. It is also interesting to note that in this example, the RX algorithm performed like an EEA by extracting the three endmember pixels located at the upper-left corner of the third panel in each of the three different rows that represented the three distinct pure signatures. This is because the three extracted endmember pixels were considered anomalous pixels. The same experiments were also conducted for different SNRs, such as 20:1 and 10:1, where the VD was also estimated to be 3, as shown in Table 18.3. The results were very similar and so they are not included here to save space.

To summarize the results demonstrated by the above three examples, it was obvious that PPI and IN-FINDR performed exactly how they were designed for, which was endmember extraction as shown in Figures 18.2(a)–(b) and 18.4(a)–(b). AMEE was also able to extract endmember pixels that were also homogeneous, but not necessarily distinct endmembers as PPI and IN-FINDR did. As shown in Figures 18.2(c) and 18.4(c), AMEE continuously extracted endmember pixels with the same spectral signature until all such endmember pixels were exhausted, and then began to extract endmember pixels with another different type of spectral signature. However, this dilemma can be resolved by including a remove-before-extract strategy in AMEE, in which all endmember pixels with different spectral signatures were eventually able to be extracted. So, with the remove-before-extract strategy implemented, the performance of the three considered EEAs was nearly the same. Unlike EEAs, UTDAs did not necessarily extract endmember pixels. Instead, they extracted pixels with most spectrally distinct and significant signatures which may be mixed pixels such as the 19 pixels in Figure 18.2(d)–(f), and represented all the 19 distinct spectral signatures. Another interesting observation demonstrated by Figure 18.4(d)–(f) is that only three pixels were extracted by the three UTDAs because of the fact that $\mathrm{VD} = 3$. These three pixels represented three distinct spectral signatures, two of which were pure signatures and the third was the background signature. If we let UTDAs continue to extract the fourth pixel, this extracted fourth pixel turned out to be another endmember pixel that represented the third pure signature that was extracted in Figure 18.2(d)–(f) but was missed in Figure 18.4(d)–(f). Finally, according to Figures 18.2(g) and 18.4(g), it was evident that the RX algorithm could only extract anomalous pixels. Comparing the results of the RX algorithm with those produced by AMEE, both algorithms performed completely opposite in the sense that the pixels that were most likely to be extracted by AMEE were homogeneous pixels as opposed to pixels that were most likely anomalous pixels by the RX algorithm. However, if we define a target formed of homogenous pixels and a target formed of anomalous pixels as a homogeneous target and an anomalous target (anomaly), respectively, in which case a target can comprises more than one pixel, then a homogeneous target can become an anomalous target when its size is decreased. For example, the three 2 × 2 panels in the first column in Figure 18.1(b) could be considered homogenous targets. When the size of four pixels was reduced to a 1 × 1 single pixel such as p^3_{11}, $\tilde{p}^3_{11}$, $\bar{p}^3_{11}$, these pixels became single-pixel anomalous targets. Conversely, an anomalous target could be expanded to a homogeneous target by filling more pixels with the same spectral signature such as the case that $p^3_{11}, \tilde{p}^3_{11}, \bar{p}^3_{11}$ could be expanded to the three 2 × 2 panels in the first column in Figure 18.1(b). At the end, an interesting problem arise: “How large is the size of a target to be considered as a homogenous or anomalous target?”. Obviously, the size of an anomalous target cannot be too large. Otherwise, it will not be called an anomaly. In this case, how large can an anomaly be? This issue has recently been investigated by Chang and Hsieh (2007). According to our experiments, AMEE and the RX algorithm seemed to extract pixels on both ends from opposite directions, i.e., endmemner pixels and anomalous pixel, with EEAs and UTDAs right in between.



(a) PPI (b) IN-FINDR (c) AMEE

(d) ATGP (e) UFCLS (f) IEA (g) RX algorithm

Figure 18.4 Results produced by the seven algorithms.

18.5 Real Image Experiments

Two sets of real image data were considered for experiments in this section. The first one is a well-known scene shown in Figures 1.9 and 1.10 collected by the airborne visible infrared imaging spectrometer (AVIRIS) over the Cuprite mining site, Nevada, in 1997. It has 224 spectral bands and image size of 350 × 350 pixels. The second image data were collected by the Digital Airborne Imaging Spectrometer (DAIS 7915) over the city of Pavia, Italy, in 2001, and has size of 400 × 400 pixels. Both image data sets have available ground truth to substantiate our pixel information analysis.

18.5.1 AVIRIS Image Data

The AVIRIS Cuprite scene is available online at http://aviris.jpl.nasa.gov in both radiance and reflectance units. Here, atmospherically corrected data will be used in order to relate to available ground spectra, and also to discuss the impact of atmospheric corrections. Prior to analysis, bands 105–115 and 150–170 were removed due to water absorption and low SNR in those bands. Figure 1.9(b) shows a single band of the AVIRIS data, where the ground truth provides the precise spatial locations of pure pixels that correspond to the five minerals, alunite, buddingtonite, calcite, kaolinite, and muscovite, labeled by "A", "B", "C", "K," and "M." These pixels are carefully verified by using laboratory spectra and a Tetracorder map of ground minerals produced by USGS (available at http://speclab.cr.usgs.gov) (see Figure 1.12). According to this map, the "B" pixel can also be considered an anomaly due to very rare presence of the buddingtonite mineral in the area. In addition to the pure pixels above, the USGS map also revealed the presence of three additional pixels: a mixed pixel composed of alunite and kaolinite, labeled as "X", a homogeneous pixel labeled as "H" and located in an area known as the "montmorillonite playa" in the lowest-rightmost corner of the scene, and, finally, an anomalous pixel labeled as "R" (stands for "rare" pixel), caused by an alteration of the muscovite mineral.

Table 18.4 tabulates the various values of the VD produced by the HFC method using different false-alarm probabilities $P_F = \alpha = 10^{-i}$ with $i = 1, 2, 3, 4, 5$.

Table 18.4 VD estimates with various false-alarm probabilities

P_F	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
p	37	27	23	22	21

Extensive experiments were conducted for each of these values and various image scenes. The results obtained for different values of p are listed in Table 18.4 and were shown to be very similar in the sense that the pure pixels representing the five pure mineral signatures were extracted and most of the extracted pixels were overlapped. According to the table, $p = 22$ seemed a reasonable estimate for the AVIRIS Cuprite image scene. Therefore, only experiments based on $p = 22$ are presented here for illustration and demonstration. Nevertheless, all the arguments made for the case of $p = 22$ can also be applied to other values of p. As shown in Figure 18.5(a) and (b), PPI and N-FINDR produced very close results in this experiment.

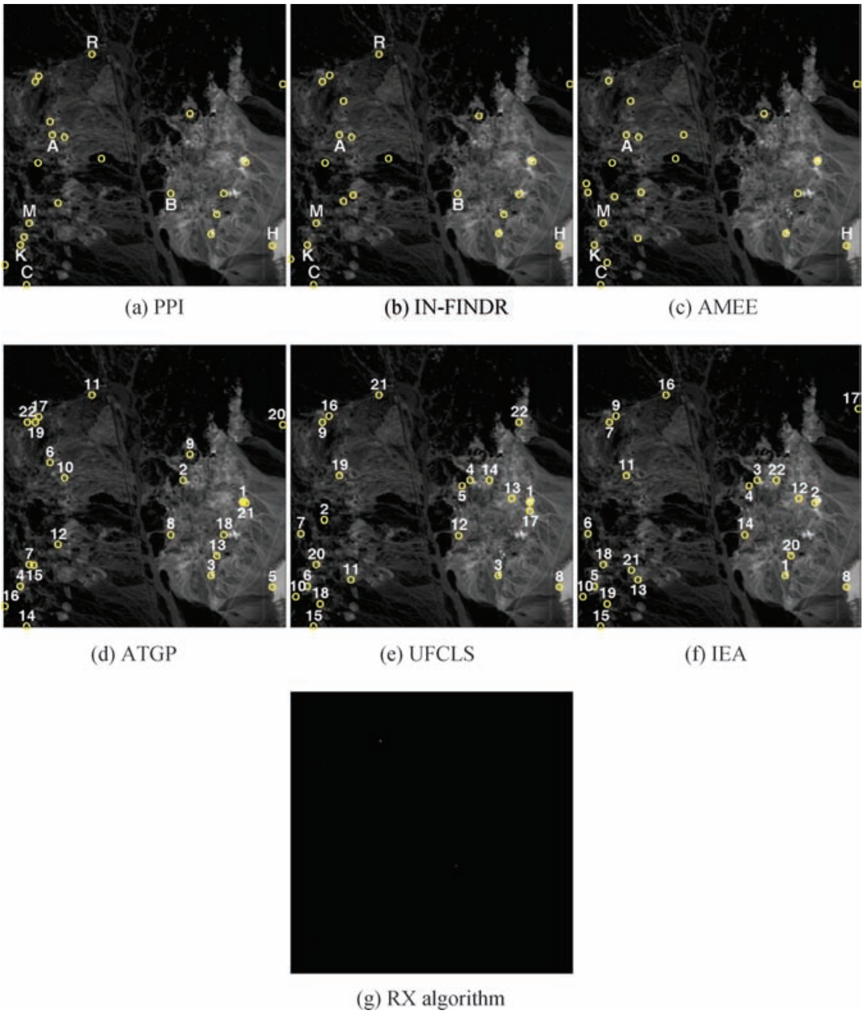
It should be noted that PPI was run by using the ENVI software using 10^4 skewers and parameter τ is set to the mean of the values of the $N_{PPI}(\mathbf{r})$ throughout the data set. (Algorithm parameters are selected in accordance with those used by Plaza et al. (2002)). These two EEAs extracted all the five pure pixels in Figure 18.5, including the anomalies labeled as "B" and "R." The above results indicated that the two EEAs also extracted anomalous pixels provided that they were pure. This observation was confirmed by comparing the detection results of the RX algorithm in Figure 18.5(g) with the output produced by both PPI in Figure 18.5(a) and IN-FINDR in Figure 18.5(b), where the latter extracted pixels in the areas where the RX filter produced a high output. Interestingly, AMEE behaved in a completely opposite way. It extracted neither the pixel labeled as "R" nor the pixel labeled as "B," which is an endmember pixel that could also be considered an anomaly. Instead, AMEE extracted pixels in homogeneous areas, such as the pixel labeled as "H" that was also extracted by PPI and IN-FINDR as an endmember pixel. This indicated that although AMEE is originally designed to extract endmembers, it only performed properly as an EEA provided that the endmember pixels were also homogeneous. Both PPI and IN-FINDR provided better results in extracting endmember pixels that were also anomalous pixels.

Furthermore, results in Figure 18.5(d)–(f) revealed that UTDAs produced results that were similar to those found by EEAs in Figure 18.5(a) and (b). Specifically, all UTDAs detected five pure pixels:

$$\mathbf{r}_{10}^{(\mathrm{ATGP})} = \mathbf{r}_{19}^{(\mathrm{UFCLS})} = \mathbf{r}_{11}^{(\mathrm{IEA})} = \mathrm{A},\ \mathbf{r}_{8}^{(\mathrm{ATGP})} = \mathbf{r}_{12}^{(\mathrm{UFCLS})} = \mathbf{r}_{14}^{(\mathrm{IEA})} = \mathrm{B}, \mathbf{r}_{14}^{(\mathrm{ATGP})} = \mathbf{r}_{15}^{(\mathrm{UFCLS})} = \mathbf{r}_{15}^{(\mathrm{IEA})}$$
$$= \mathrm{C}, \mathbf{r}_{4}^{(\mathrm{ATGP})} = \mathbf{r}_{6}^{(\mathrm{UFCLS})} = \mathbf{r}_{5}^{(\mathrm{IEA})} = \mathrm{K},\ \mathbf{r}_{7}^{(\mathrm{ATGP})} = \mathbf{r}_{20}^{(\mathrm{UFCLS})} = \mathbf{r}_{18}^{(\mathrm{IEA})} = \mathrm{M},$$

where the superscript is included in notations to emphasize which algorithm was used to find the pixels. The most remarkable difference among EEAs in Figure 18.5(a)–(c) and UTDAs in Figure 18.5(d)–(f) was the fact that contrary to all EEAs, the three UTDAs extracted the mixed pixel labeled as "X," where $\mathbf{r}_{2}^{(\mathrm{ATGP})} = \mathbf{r}_{4}^{(\mathrm{UFCLS})} = \mathbf{r}_{3}^{(\mathrm{IEA})} = \mathrm{X}$. The three UTDAs also extracted the anomaly labeled as "R," that is, $\mathbf{r}_{11}^{(\mathrm{ATGP})} = \mathbf{r}_{21}^{(\mathrm{UFCLS})} = \mathbf{r}_{16}^{(\mathrm{IEA})} = \mathrm{R}$, as well as the homogeneous pixel, that is, $\mathbf{r}_{5}^{(\mathrm{ATGP})} = \mathbf{r}_{8}^{(\mathrm{UFCLS})} = \mathbf{r}_{8}^{(\mathrm{IEA})} = \mathrm{H}$.

The real image experiments presented above provided results that complemented synthetic image experiments in previous sections. First, two of the considered EEAs, namely, PPI and IN-FINDR, produced very similar results. In particular, both algorithms extracted endmember pixels that were also anomalous pixels. Since those pixels are located in areas where the RX algorithm produced a high output, this suggested that anomaly detection could be an alternative to



(a) PPI (b) IN-FINDR (c) AMEE

(d) ATGP (e) UFCLS (f) IEA

(g) RX algorithm

Figure 18.5 Pixel information extracted by seven algorithms for the cuprite image.

endmember extraction if the occurrence of pure pixels was low. On the contrary, AMEE could only behave like an EEA provided that the endmember pixels were also homogeneous pixels. Since AMEE uses spatial correlation that plays a more significant role in multispectral images than hyperspectral images, it is our belief that AMEE may perform more effectively for multispectral images in endmember extraction. On the other hand, the three considered UTDAs consistently found all the available pure pixels in the AVIRIS image experiments. However, these algorithms also extracted other types of pixels, including mixed, homogeneous, and anomalous pixels. Therefore, UTDAs could not be used to fully replace EEAs in real data experiments. However, we have experimentally proved in Chang and Plaza (2006) that UTDAs can be used to produce a good initial set of endmembers to significantly speed up performance of an EEA, where many of the initial UTDA-generated pixels eventually turn out to be desired endmembers.

Although experiments with the Cuprite data provided several interesting findings, most of the targets in the scene were given by mineral signatures, and available ground truth is limited to a few pixels only. In order to test the performance of the considered algorithms in a more complex data analysis scenario including different types of targets, a further experiment with additional data sets was conducted in the following section.

18.5.2 DAIS 7915 Image Data

The scene in Figure 18.6 contains different types of targets and several ground-truth regions in a complex urban landscape.

Figure 18.6 shows available ground truth for the scene superimposed on the spectral band at 639 nm collected by the DAIS 7915 imaging spectrometer. This information will be useful to analyze the pixel information extracted from a complex analysis scenario dominated by complex urban features. The scene reveals a dense residential area on one side of the river, as well as open areas and meadows on the other side. Ground-truth information is also available for several areas, with the following land-cover classes: water (blue), trees (dark green), asphalt (orange), parking (cyan), bitumen (red), roofs (magenta), meadows (light green), soil (maroon), and shadows (yellow). By following previous studies on this scene, we take into account only 40 spectral bands of reflective energy and skip thermal infrared bands and middle infrared bands above 1958 nm because of low SNR in those bands. The 40 considered bands are collected by two different spectrometers mounted on DAIS 7915. The first spectrometer uses 32 bands in the range 496–1035 nm (spectral resolution of 17 nm) and the second one uses eight bands in the range 1539–1756 nm (spectral resolution of 27 nm). The scene is atmospherically corrected, and has spatial resolution of 5 m. Table 18.5 tabulates the various values of VD produced by the HFC method with different false-alarm probabilities. According to the table, $p = 14$ seems a reasonable estimate.

Figure 18.7 shows the pixels extracted by different algorithms from the Pavia urban scene. As shown in Figure 18.7(c), AMEE extracted several pixels belonging to classes formed of homogeneous areas, such as river and meadows. On the contrary, the RX algorithm detected an anomaly in Figure 18.7(g), located between the two roads that cross the downtown area from the lower leftmost corner of the scene to the river. Interestingly, this anomaly was not extracted by AMEE but was found by the other two EEAs: PPI and IN-FINDR. In addition, this anomaly was extracted first

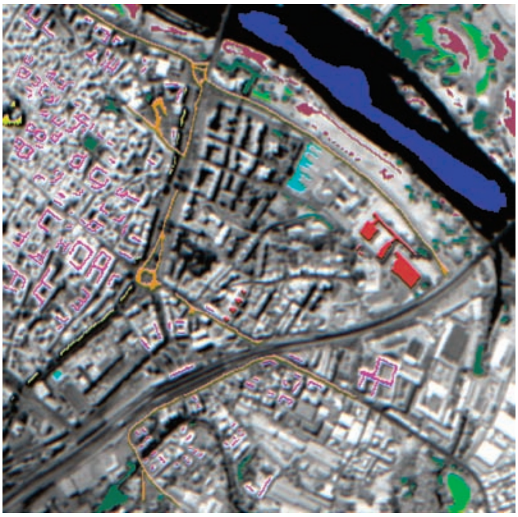
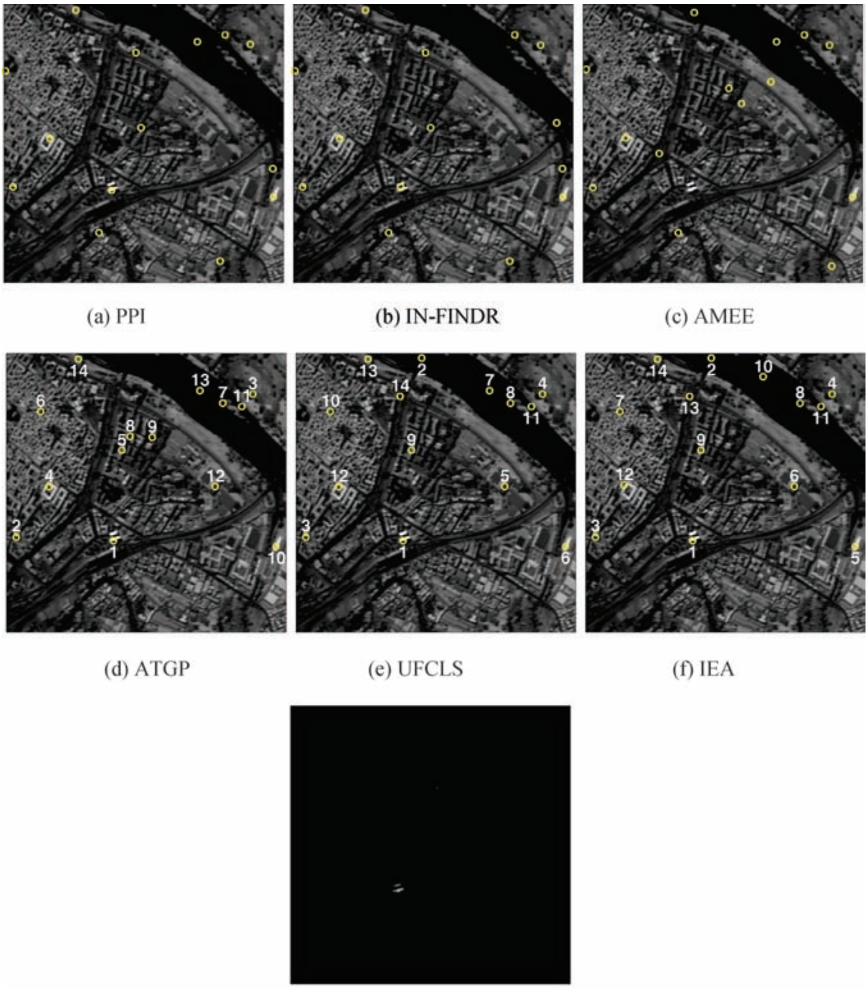


Figure 18.6 Pavia image scene.

Table 18.5 VD estimates with various false-alarm probabilities

P_F	10^{-1}	10^{-2}	10^{-3}	10^{-4}	10^{-5}
p	18	17	16	14	14

by the three UTDAs in Figure 18.7(d)–(f), which indicated that the anomaly also corresponded to the pixel vector with maximum length in the scene. It is also important to emphasize that both PPI and N-FINDR produced very similar results, with most of the pixels overlapped in Figure 18.7(a) and (b). In particular, the pixels extracted from the upper-left corner area of the scene were identical. This area is dominated by the roofs class (in magenta in Figure 18.6) that was of great importance for urban planning studies (Benediktsson et al., 2005). It is also interesting to note that the



(a) PPI (b) IN-FINDR (c) AMEE

(d) ATGP (e) UFCLS (f) IEA

(g) RX algorithm

Figure 18.7 Pixel information extracted by seven algorithms for the DAIS 7915 Pavia image.

Table 18.6 SAM-based spectral similarity values produced by the pixels extracted by different algorithms and mean spectra obtained from ground-truth classes

	EEAs:			UTDAs:		
	PPI	IN-FINDR	AMEE	ATGP	UFCLS	IEA
Bitumen	—	—	—	0.057	0.057	0.057
Meadows	—	—	0.028	—	—	—
Water	0.083	0.091	0.073	0.073	0.073	0.083
Shadows	0.082	0.082	0.082	—	—	—
Parking	0.077	0.077	0.068	0.077	0.077	0.077
Roofs	0.053	0.053	0.048	0.048	0.048	0.048
Soil	0.063	0.063	0.022	0.063	0.063	0.063
Trees	0.047	0.052	0.058	0.058	0.058	0.058
Asphalt	—	—	0.067	0.051	0.051	0.051

three EEAs, PPI, IN-FINDR, and AMEE, extracted a pixel in Figure 18.7(a)–(c) that belonged to the shadow class (in yellow in Figure 18.6) that was not detected by any of the three UTDAs. For illustration purposes, Table 18.6 tabulates the spectral similarity values measured by SAM between the mean spectral signature for each ground-truth class in Figure 18.6 and the most similar pixel extracted by the considered EEAs and UTDAs. In some cases, none of the pixels produced by the considered algorithms could be associated with a certain class.

Interestingly, Table 18.6 confirmed our findings that UTDAs could not detect pixels belonging to the shadow class (such a problem was not observed in EEAs). This might be due to the influence of the pixel vector used as the initial one in the sequential process implemented by UTDAs, that is, the brightest pixel in the scene. In addition, AMEE was the only algorithm able to extract pixels in the meadows class. This was because this class was clearly dominated by homogeneous pixels. Also, the fact that none of the tested EEAs extracted pixels in the bitumen class seemed to indicate that this class was dominated by mixed pixels. Finally, both PPI and IN-FINDR could not extract pixels in the asphalt class, which was subject to spectral variability due to illumination interferers. Interestingly, the spatial information that AMEE took into account helped to characterize this class (it should also be noted that AMEE is based on SAM, which is insensitive to illumination effects). It is our belief that AMEE could perform better in this experiment due to the moderate spectral dimensionality of the urban scene and the importance of spatial information in such an environment.

In summary, the experimental results presented in this section clearly demonstrated a need of combining different types of algorithms for pixel information analysis when the study site involves a complicated scene such as the Pavia urban scene in Figure 18.6.

18.6 Conclusions

Many algorithms have been developed for various applications in hyperspectral data exploitation. Of particular interest is, “What pixel information do these algorithms really extract to achieve the goal that they are designed for?” This chapter investigates this issue via three categories of algorithms, EEAs, UTDAs, and an RX-anomaly detection algorithm, which have received considerable interest in the past. Since these algorithms are unsupervised and performed without prior knowledge, it is important to examine the utility of these algorithms. This chapter explores the insights into these algorithms. In particular, we address an important issue of whether these algorithms really do what they are designed to do. While doing so, four types of pixels, pure pixel, mixed pixel, homogeneous pixel, and anomalous pixel, are introduced for pixel information analysis. In addition, a set of custom-designed experiments are conducted, where the four types of pixels described above are used to evaluate the performance of these algorithms in terms of pixel information extraction. Interestingly, experimental results provide many intriguing findings in these algorithms that may help image analysts in selection of algorithms for specific applications.

V

Hyperspectral Information Compression

Due to enormous data volumes provided by hyperspectral imaging sensors via hundreds of contiguous spectral channels with high spectral interband correlation, it becomes increasingly evident that high compression ratios can be achieved for hyperspectral data without loss of significant information. By realizing tremendous benefits that can be gained by data compression, many efforts have been devoted to hyperspectral data compression. Since a hyperspectral image can be considered as an image cube, many hyperspectral data compression algorithms have been developed by taking advantage of existing two-dimensional (2D) image compression algorithms and extending these 2D techniques to their three-dimensional (3D) counterparts that are generally applied to video images. Most notable are JPEG2000 multicomponent and 3D SPHIT (Set Partition in Hierarchical Tree) that are extended from their counterparts, JPEG2000 and 2D SPHIT. However, there is a danger in taking such an approach without extra caution due to two unique issues encountered in hyperspectral images that never occur in pure pixel-based 3D images. One is the issue of subpixel targets whose size is smaller than pixel size/resolution. Such a target is generally embedded in a single pixel vector and their presence cannot be visualized by its spatial extent. In this case, pure pixel-based 3D image compression techniques may fail to capture their existence. The other is the issue of mixed pixel vectors that may contain two or more substances mixed together within a single pixel vector. Once again, pure pixel-based 3D image compression techniques may not be effective to extract these substances from a mixed pixel vector. In order to address these two issues, spectral compression and spatial compression must be decoupled in such a way that the spectral properties of subpixel targets and mixed pixel vectors can be first preserved and retained by spectral compression so that such spectral characteristics cannot be further suppressed, compromised, or sacrificed by spatial compression or direct 3D compression.

When data compression is performed, two different types of criteria must be specified. One is a design criterion used to develop a compression technique. The other is a performance criterion

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

used to evaluate the effectiveness of a specified compression technique in performance. While these two types of criteria can be considered as separate criteria, they are generally correlated to each other. Specifically, a performance criterion is always a major driving force to determine what design criterion should be selected to design a desired compression technique. So, how effective a data compression technique is in fact determined by a specific application that in turn determines what a best performance criterion is. In other words, when lossy compression is performed, a selected performance criterion must be adapted to retaining fidelity of desired information required for different applications. For example, the Karhunen–Loeve transform (KLT) is the optimal linear transform when both design and performance criteria are the mean squared error (MSE), but it may not be optimal when compression ratio (CR) is used as a performance criterion. This is particularly true for hyperspectral data exploitation where compression performance varies with applications. As another example, in linear spectral mixture analysis (LSMA) for hyperspectral imagery the compression performance should be measured by spectral unmixing error instead of CR or MSE. Similarly, for anomaly detection or endmember extraction the compression performance must be measured by how effectively anomalies or endmembers are extracted rather than CR or MSE. It is a common practice in data compression community that CR, MSE, signal-to-noise ratio (SNR), and peak SNR (PSNR) are most widely used criteria in performance analysis. Unfortunately, these criteria may not be effective performance measures in the above-mentioned applications since the targets of interest such as anomalies and endmembers usually do not have many samples in the data set and, thus, their contribution to MSE, SNR, or PSNR is generally too little to substantiate their existence. Instead, their presence can be only characterized by their spectral properties. On the other hand, MSE or SNR may be appropriate criteria to measure spatial correlation in spatial domain-based applications, but they are certainly not effective in measuring spectral correlation.

Spatial image compression has been studied extensively and well documented in the literature. However, the criteria such as MSE, SNR, or PSNR used for designing and developing compression algorithms are generally not appropriate or ineffective for hyperspectral image data exploitation. Besides, most techniques developed for data compression intend to increase CR, which is calculated by the ratio of the original data size to the compressed data size by maintaining a certain level of image quality measured by a criterion, for example, MSE or SNR. Unfortunately, such data size reduction-based CR approach is generally not practical for hyperspectral image analysis, specifically to address issues of subpixel targets and mixed pixel vectors as mentioned above. More specifically, what really matters for hyperspectral data compression is information not the data size. Therefore, the effectiveness of a hyperspectral data compression technique should not be measured by reduction of data size, but rather by information retained for image analysis after data compression. To clarify this point, the terminology of hyperspectral information compression is introduced here to specifically emphasize that the compression is performed on the basis of data information not data itself. Furthermore, since the information to be preserved by compression is completely determined by a specific application, such hyperspectral information compression can also be referred to as exploitation-based hyperspectral data compression (EHDC) that is performed according to criteria determined by various applications. For example, if an application is LSMA, it is then called LSMA-based hyperspectral data compression. Similarly, if endmember extraction is used for an application, it is called endmember extraction-based hyperspectral data compression. So, an EHDC is generally carried out by a two-stage process with the first stage process performed by a specific exploitation technique, then followed by a 3D compression in the second stage process.

So, in Part V what we are particularly interested is spectral compression specifically designed to perform lossy compression for hyperspectral data at low or very low bit rates where spectral statistics are considered as main performance criteria to develop compression techniques. Accordingly, the commonly used spectral compression technique, principal components analysis (PCA), can be viewed as a second-order spectral statistics-based transform that uses data sample variance as a design criterion to measure the significance of principal components (PCs). Similarly, KLT is also a transformation using a second order of spectral statistics of covariance function as a performance criterion but uses MSE used as a design criterion for distortion measure. In contrast, independent component analysis (ICA) is a transform using an infinite order of spectral statistics as a performance criterion, while the mutual information is used as a design criterion to measure statistical spectral independence among independent components it generates. Such spectral compression has shown great potential in hyperspectral imaging applications. It can be implemented in two perspectives, dimensionality reduction (DR) and band selection (BS), where the former reduces dimensionality via a transform such as PCA or feature extraction-based discriminant analysis, while the latter selects an appropriate set of spectral bands from the set of original data to represent the data. Technically speaking, it is more accurate to consider BS as a data reduction technique rather than data compression technique because it does not really perform compression other than discarding those spectral bands that are not selected. Nevertheless, it does compress data in the sense of data reduction. Accordingly, from a CR point of view BS is indeed a data compression technique.

One serious drawback encountered in hyperspectral compression is the requirement of the prior knowledge about how many dimensions needed to be retained, q, after DR and how many bands needed to be selected, $\tilde{q}$, after BS. Since finding an accurate value of q or $\tilde{q}$ is very challenging, many techniques developed for DR and BS have either assumed this value *a priori* or performed on a trial and error basis. Unfortunately, if this value is not correct, this whole process of DR and BS must be repeated again for a different value of q or $\tilde{q}$. This is a significant setback. Although a recently developed concept, virtual dimensionality (VD) presented in Chapter 5, can be used for this purpose, it only provides a reasonable estimate, but not necessarily an accurate estimate. To alleviate this dilemma two progressive processes to perform spectral compression are developed, progressive spectral dimensionality process (PSDP) in Chapter 20 and progressive band dimensionality process (PBDP) in Chapter 21, where two new concepts, dimensionality prioritization (DP), and band prioritization (BP) are introduced in Chapters 20 and 21, respectively, to allow users to perform spectral compression in a forward and backward manner in the sense of dimensionality expansion and dimensionality reduction. In this case, VD can be used to determine the lower and upper bounds to q or $\tilde{q}$ by a range of $[n_{\mathrm{VD}}, 2n_{\mathrm{VD}}]$, with n_{VD} being the VD-estimated value. Interestingly, similar to design and performance criteria used by data compression both DP and BP also require two types of criteria. One is design criteria that are the same design criteria used to develop DR and BS techniques. The other is prioritization criteria used to rank DR-compressed spectral dimensions compressed by DR as well as to rank spectral bands selected by BS.

Although PSDP via DP and PBDP via BP can process DR and BS in a progressive manner between two bounds n_{VD} and $2n_{\mathrm{VD}}$ without actually knowing the number of spectral dimensions needed to be retained, q, by DR and the number of spectral bands needed to be selected, $\tilde{q}$, by BS, there are still two major issues needed to be resolved in DR and BS. One is fixed-size dimensionality used by traditional DR and BS where the value of q or $\tilde{q}$ must be fixed during the entire process. Unfortunately, this may not be applicable to various applications when the value of q or $\tilde{q}$ cannot be fixed at a constant. For example, the value of q or $\tilde{q}$ cannot be the same when it comes to applications such as endmember extraction, anomaly detection, and spectral unmixing. In other words, q or $\tilde{q}$ must be adaptive. To alleviate this dilemma a new concept of dynamic dimensionality allocation (DDA) derived from variable-length coding in information theory is further developed in Chapter 22 where a hyperspectral data set and its hyperspectral signatures of interest are interpreted as an information source and source alphabets, respectively. With this interpretation the relative spectral discriminatory probabilities (RSDPB) earlier defined in (Chang 2000, 2003a) can be considered as source probabilities to define variable coding lengths for each of hyperspectral signatures to find its own spectral/band dimensionality via, DDA. Finally, a remaining issue is that when DP and BP are used to prioritize spectral dimensions and bands, we may expect that if a spectral dimension/band is highly prioritized dimension/band, so are its neighboring spectral dimensions/bands. While this may be true for BP that prioritizes spectral bands based on information contained in separate and individual spectral bands and does not take into account interband correlation, it may not be necessarily true for DP where the spectral dimensions produced by DR have been taken care of by a DR transform such as PCA, projection pursuit. As a consequence, PBDP cannot be directly used for BS unless a preprocessing called band de-correlation (BD) is included prior to BS. So, such resulting process of implementing PBDP coupled with BD as a preprocessing is called progressive band selection (PBS) that is the main theme in Chapter 23. On the other hand, PSDP can be directly applied to progressive dimensionality reduction and expansion as shown in Safavi (2010) and Chang and Safavi (2011). Therefore, there is no counterpart of Chapter 23 needed for PSDP.

Finally, it should be noted that the progressive process proposed in Part V is very different from a sequential process and an iterative process in that how the process takes place. A sequential process is carried out sequentially where the data sample vectors are fully processed one at a time and the process is completed after it processes the last data sample vector. In this case, a sequential process is a one-shot process. An iterative process generally requires an iterative equation or iterative equations to update and improve results produced by a previous iteration so that better results can be produced for next iteration. It not only fully processes data sample vectors, but also requires repeating the same process over and over again. The proposed progressive process can be considered as embedded process where the previous results are always embedded in the results produced by the subsequent process. So, unlike a sequential or an iterative process a progressive process requires a finite number of passes to complete its process where data sample vectors are only partially processed in each pass and the results obtained in a previous pass will be part of results obtained by a subsequent process. This is a key difference to distinguish a progressive process from an iterative process where the latter process does not need to include previous results as part of results generated at the next iteration, improves its results obtained at previous iterations. More details can be found in Chang (2013).

19

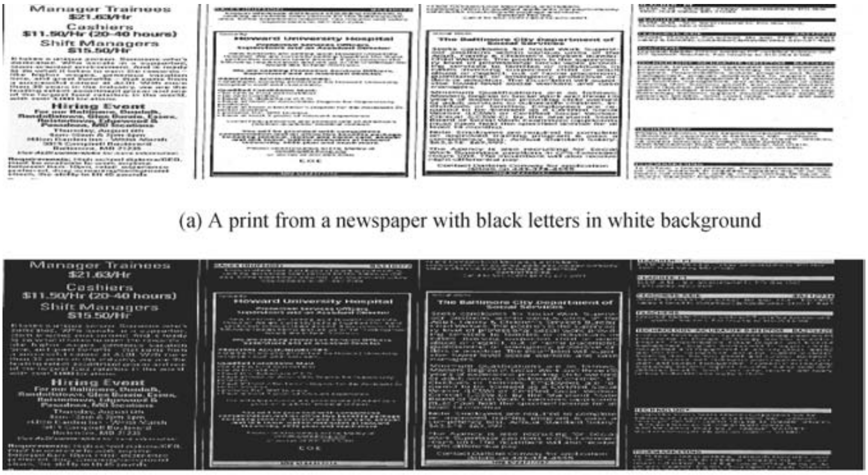
Exploitation-Based Hyperspectral Data Compression

A key to success in data compression is determined by how much information needed to be stored that will be later retrieved for future data processing. Whether or not a compression technique is effective is measured by the degree of loss in future information *retrieval* rather than information *recovery*. Accordingly, data loss does not necessarily mean information loss, which implies that data compression does not necessarily perform information compression. This is particularly evidential in hyperspectral data compression in which many research efforts have been directed to data compression rather than information compression. This chapter introduces a new concept of hyperspectral information compression and explores its utility in hyperspectral data exploitation. In particular, a three-stage process is developed to address issues arising in hyperspectral information compression where the first stage is spectral compression performed by spectral dimensionality reduction via interband de-correlation followed by two stages that implement either exploitation-based information compression in second stage and 3D data compression in the third stage or in a reverse order. Such three-stage hyperspectral information compression is measured by two key factors: information loss and exploitation-based compression criteria. In order to demonstrate the difference between data compression and information compression in hyperspectral image processing, various scenarios are conducted for experiments.

19.1 Introduction

Because of significantly improved spectral and spatial resolution resulting from recent advanced remote sensing instruments many subtle substances such as rare minerals, special species, small objects etc. can now be uncovered and diagnosed by custom-designed data processing techniques such as feature extraction for exploitation. However, this benefit also comes at a price, that is, how to process enormous data volumes without compromising desired information for data processing, specifically, how to compress data while preserving vital information for future information retrieval and data processing. Apparently, this heavily depends on the data to be processed. Different data are acquired for various applications; thus, they require specific processing techniques. This chapter investigates hyperspectral data compression from an information point of view, referred to as hyperspectral information compression.

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.



Manager Trainees
$21.63/Hr
Cashiers
$11.50/Hr (20-40 hours)
Shift Managers
$15.50/Hr

Hiring Event

Howard University Hospital

The Baltimore City Department of Social Services

Manager Trainees
$21.63/Hr
Cashiers
$11.50/Hr (20-40 hours)
Shift Managers
$15.50/Hr

Hiring Event

Howard University Hospital

The Baltimore City Department of Social Services

(a) A print from a newspaper with black letters in white background

(b) A print converted from the print in (a) by changing white background to black background and black letters to white letters

Figure 19.1 An example of a print from a newspaper.

Before proceeding we need to make a distinction between information compression and data compression. Let us consider the following example. Assume that a document such as a newspaper is represented by a binary image with 0 corresponding to letters and 1 being assigned to background so that the document can be read by black letters in white background as shown in Figure 19.1(a).

Now, if we perform a lossless data compression on the image in Figure 19.1(a) by converting all 1s to 0s and 0s to 1s, the resulting document shown in Figure 19.1(b) is also a binary image with 1 and 0 assigned to letters and background, respectively, in which case the original black letter-white background document is converted to a binary image with white letters in black background. In light of data compression, there is no data loss in these two images, but the form of the information presented by these two images is quite different in terms of contrast. Similarly, this is also true for photo development from negative films. These two simple examples demonstrate that lossless data compression can actually enhance or degrade information and does not necessarily imply information compression. In other words, lossless data compression may result in changes of different forms to represent information such as image enhancement or degradation, contrast improvement or reduction for visual assessment as demonstrated in Figure 19.1. For instance, an image after JPEG-2000 lossless compression may appear more visually pleasant than its original image in terms of information representation. Bearing this in mind a key to success in data compression is not determined by how much data volumes have been compressed, but rather determined by how much information is retained to be used for image interpretation. Therefore, whether or not a compression technique is effective should be measured by the degree of loss in information "*retrieval*" for data processing rather than information "*recovery*" in a data form. Accordingly, data loss does not necessarily mean information loss. This further implies that data compression does not necessarily perform information compression. It is particularly evident in hyperspectral data compression in which many research efforts have been focused on data compression in the past rather than information compression that should be the case in hyperspectral compression.

On the other hand, the criterion used to measure compression performance is also crucial in hyperspectral data compression. The following simple example should suffice to justify this salient difference. Consider a combat vehicle such as a tank running around in a very large battle field, for example a desert field. Due to the size of a tank, usually 8 m × 4 m, its spatial presence in the entire image scene may only occupy a few pixels. Consequently, the energy of a tank contributed to the second-order statistics such as mean squared error (MSE) or signal-to-noise ratio (SNR) is very little and limited. So, when compression techniques based on criteria of second-order statistics are performed, the tank considered as the only main target in the field will be very much likely sacrificed if the criterion to used is data compression ratio, which is high. This is because MSE caused by the tank is almost zero compared to the entire image and SNR nearly remains unchanged even if the tank is removed from the field. However, from a viewpoint of intelligent gathering, the tank is the only vital information that needs to be preserved. The loss of the tank information is devastating. Under this circumstance, an effective criterion must be one that can capture the tank information during compression. Since the tank can be considered as an anomaly in the image scene, it generally cannot be characterized by second-order statistics, but rather by high-order statistics such as skewness, kurtosis, etc. In this case, data compression must be performed by information compression via a high-order statistics-based criterion that can retain the tank information to accomplish its goal in which case anomaly detection should be used as an exploitation criterion for compression.

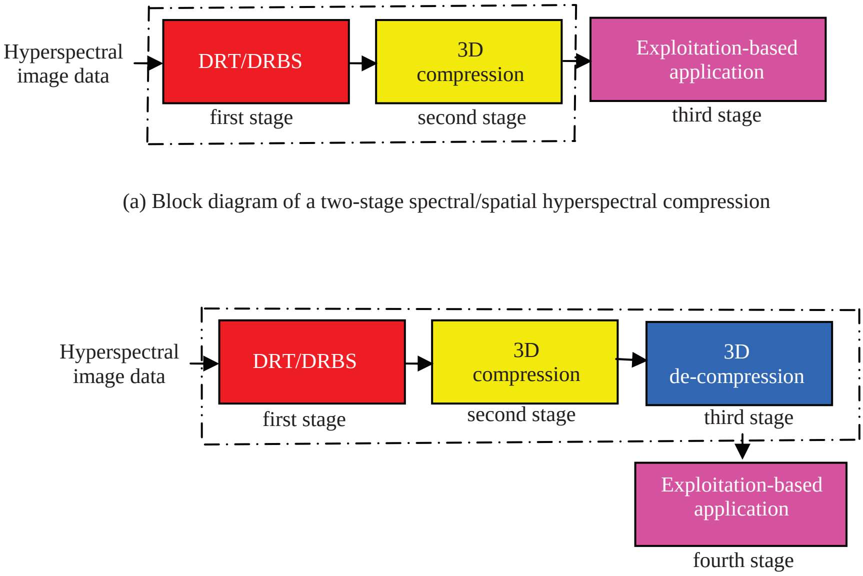
The reason why the hyperspectral imagery is called "hyperspectral" arises in its wealthy spectral information provided by hundreds of contiguous spectral channels. It is the spectral information that matters and is more crucial and critical than spatial information such as the above tank example. Therefore, an effective and efficient hyperspectral information compression technique should compress data in a hierarchical fashion rather than one-shot compression such as 3D compression. Since different data sets provide a different levels of information, a compression technique that works for one case does not necessarily work for another. So, a desired hyperspectral information compression technique should be able to adapt to the data to be processed where the compression must be performed based on information extraction and retrieval for subsequent data processing. This type of information compression is determined by applications, not classical data compression that mainly deals with "*data*" loss and reconstruction. Unfortunately, most data compression techniques available in the literature are developed to achieve "*data*" compression, but not "*information*" compression.

19.2 Hyperspectral Information Compression Systems

In this section, we investigate hyperspectral information compression systems, which implement exploitation-based criteria to compress hyperspectral imagery while preserving information for data processing. In particular, we develop two types of hyperspectral exploitation-based compression systems depicted in Figure 19.2(a) and (b) and Figure 19.3 where the compression takes place in different stages sequentially.

While both systems perform dimensionality reduction by transforms (DRT) and dimensionality reduction by band selection (DRBS) in their first stage to reduce spectral dimensionality, exploitation-based application compression and 3D compression implemented in the second and subsequent stages of both systems are actually reversed. In the systems in Figure 19.2(a) and (b) a 3D compression is carried out in middle stage (second stage in Figure 19.2(a) and second and third stages in Figure 19.2(b)) prior to an exploitation-based application in the last stage (third stage in Figure 19.2(a) and fourth stage in Figure 19.2(b)) compared to the system in Figure 19.3 that performs exploitation-based applications in the second stage prior to 3D compression in the third stage.

According to Figure 19.2(a) and (b) and Figure 19.3, three key components are involved in the designed information compression systems, each of which has tremendous impact on the compression performance. The first component must be performed and executed by either DRT or DRBS developed in Chapter 6. This step performs spectral dimensionality reduction that is a crucial step



(a) Block diagram of a two-stage spectral/spatial hyperspectral compression

(b) Block diagram of a three-stage spectral/spatial hyperspectral compression

Figure 19.2 Spectral/spatial hyperspectral compression systems with 3D compression followed by exploitation-based application.

to preserve spectral information that provides all the necessary and desired information for its follow-up exploitation-based application. The second component is an exploitation-based application that measures effectiveness of an information compression system. Such an exploitation-based application includes linear spectral mixture analysis, anomaly detection, mixed pixel classification, and quantification in Chang (2003b) and endmember extraction in Chapters 7–11 and so on. Since DRT/DRBS is implemented in the first component prior to the second component, it has tremendous impact on its follow-up exploitation-based application implemented in the second component. Consequently, these two components are actually coherent and should be considered jointly together when they are implemented. On some occasions, they can be combined and implemented as a single process for spectral compression. Nevertheless, depending on different applications a certain stage of compression may be skipped when it is not necessary. The third component is 3D compression that performs spectral/spatial compression on the entire image as a cube. Many spectral/spatial compression techniques developed in the past for hyperspectral imagery take two approaches. One is to perform 3D compression directly on hyperspectral data as depicted in Figure 19.4.

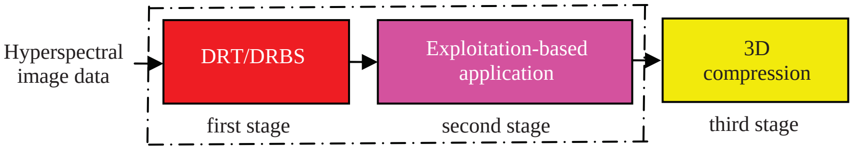


Figure 19.3 Spectral/spatial hyperspectral compression systems with exploitation-based application followed by 3D compression.

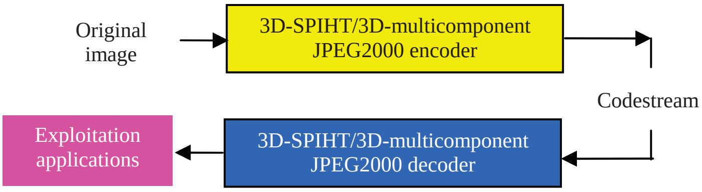

Original image
3D-SPIHT/3D-multicomponent JPEG2000 encoder
Codestream
3D-SPIHT/3D-multicomponent JPEG2000 decoder
Exploitation applications

Figure 19.4 Three-dimensional compression system.

Or alternatively, a second approach is to perform spectral compression to remove interband spectral redundancy followed by 2D spatial compression on spectrally de-correlated images. The assumption made here is that there is little spectral information remaining after spectral compression is performed. This is generally not true. For example, a components analysis transform such as PCA can only spectrally de-correlate second-order statistics, while the correlation characterized by high-order statistics stilled remains in the image data. In this case, compressing 2D spectral images alone essentially discards high-order statistics of correlation among spectral images. Interestingly, this observation has been overlooked as witnessed in many publications in the literature. On the other hand, if the BS is used to perform DR, 3D compression techniques should be used to achieve both spectral and spatial compression since there is still spectral correlation among selected bands and 2D compression techniques can only compress spatial correlation not spectral correlation.

19.3 Spectral/Spatial Compression

Since band-to-band correlation is usually very high in hyperspectral imagery, removing such redundant information can achieve a significant compression ratio. Two major approaches are generally used for hyperspectral compression, which are dimensionality reduction by transforms (DRT) and DRBS in Chapter 6. The DR is often accomplished by DRT that compacts information into a small number of components, while the DRBS selects a small number of bands in some sense of optimality to represent data. However, a key issue is how to find an optimal component transform to perform DR for best possible hyperspectral compression or how to effectively select *significant* bands that can preserve desired information for hyperspectral compression to optimize performance of a designated exploitation application. In other words, the success of DRT and DRBS in hyperspectral compression is determined by how much information is extracted and preserved for the follow-up exploitation data processing after DRT and DRBS. Therefore, DRT and DRBS must be performed by custom-designed criteria for information extraction. This issue can be addressed by hyperspectral compression via dimensionality prioritization in Chapter 20, and by hyperspectral compression via band prioritization in Chapter 21. While the techniques developed in Chapters 20 and 21 can be directly applied to hyperspectral information compression, various versions of DRT and DRBS developed in Chapter 6 are not immediately ready for compression since they are generally developed for DR and not particularly designed for information compression. Therefore, in what follows, we reinvent the wheel by redeveloping the techniques in Chapter 6 for the purpose of hyperspectral information compression.

There are key differences between the spectral/spatial compression presented in this section and 3D-cube compression. One is that our proposed spectral/spatial compression de-couples spectral compression from spatial compression to perform spectral dimensionality reduction prior to 3D-cube compression. Another is that after spectral dimensionality reduction our spectral/spatial compression still performs 3D-cube compression on the spectral dimensionality reduced 3D-cube data compared to only 2D spatial compression being performed on those spectral compressed data by 1D spectral compression. A third difference is that there are indeed two types of spectral compression carried out in our proposed spectral/spatial compression: one is spectral dimensionality reduction and the other is spectral redundancy in 3D-cube compression. As a result, two compression criteria, spectral compression criterion and exploitation-based compression criterion, are needed and must be designed from an exploitation point of view to best fit applications. Finally, a fourth difference is that that according to our experience spectral information is better preserved using dimensionality reduction than using 1D wavelet compression since it offers better de-correlation. Accordingly, on many occasions even spectral dimensionality reduction implemented in conjunction with only 2D spatial compression may outperform 3D-cube compression techniques. One such example is the linear spectral mixture analysis (LSMA)-based hyperspectral image compression (Du and Chang, 2004a) where LSMA is first used to perform spectral compression by transforming an original hyperspectral image cube to a small number of abundance fractional images that are further processed by a follow-up spatial compression. It is interesting to note that many transform coding methods developed in the literature for hyperpspectral image compression generally perform 1D-spectral/2D-spatial compression where a 2D spatial compression technique is applied to individual spectral decorrelated components. However, it has been shown in Ramakrishna (2004), Ramakrishna et al. (2005a, 2005b) that 1D-spectral/3D-cube compression performed slightly better than 1D-spectral/2D-spatial compression. This is because the former performs two types of spectral compression, spectral dimensionality reduction by 1D spectral compression followed by removing spectral redundancy via 1D discrete wavelet in 3D-cube compression as opposed to the latter that is only benefited from 1D spectral dimensionality reduction. As a result, 2D spatial compression is generally not as effective as 3D-cube compression.

19.3.1 Dimensionality Reduction by Transform-Based Spectral Compression

Despite the fact that a hyperspectral image can be viewed as a 3D image cube, there are several major unique features that a hyperspectral image distinguishes itself from being viewed as a 3D image cube. The first and foremost is spectral features provided by hundreds of contiguous spectral channels. Unlike pure voxels in a 3D image, a hyperspectral image pixel vector is specified by a wide range of wavelengths in a third dimension that characterizes the spectral properties of a single pixel vector. Using the spectral profile captured in the spectral domain a single pixel vector in a 3D hyperspectral image cube can be solely analyzed by its spectral characterization. Another important unique feature provided by hyperspectral imagery is that many material substances of interest can be only explored by their spectral properties, not spatial properties such as small wastes in environmental pollution, chemical/biological agent detection in bioterrorism, camouflaged combat vehicles, and decoys in surveillance applications. In addition, certain targets such as chemical plumes, biological agents, which are considered to be relatively small with no rigid shapes but yet provide significant information, generally cannot be processed by rigid object-based image processing or identified by prior knowledge. Instead, these targets can be only captured and characterized by their spectral properties. Therefore, when a compression ratio is high, whether or not a hyperspectral image compression technique is effective may not be necessarily determined by its spatial compression as do most compression techniques in image processing. This is because small and subtle targets such as subpixel and mixed pixel targets may be very likely sacrificed by low-bit rate compression due to their limited spatial presence. Under such a circumstance, we need rely on spectral compression to retain these targets. Accordingly, separating spectral compression from 3D compression may be more desirable and effective than 3D-cube compression performing spectral and spatial information all together simultaneously in the sense that both JPEG2000 Part II and 3D-SPIHT codec perform spectral and spatial compression using separable transformations (i.e., 1D linear transform or 1D wavelet packet transform in the spectral dimension and 2D discrete wavelet transform (DWT) in the spatial dimensions) as a one-shot operation. In what follows, several transform-based spectral compression-based approaches to dimensionality reduction are developed for this purpose.

19.3.1.1 Determination of Number of PCs/ICs to be Retained

One of primary obstacles to implement PCA/ICA is to determine how many principal components (PCs) or independent components (ICs) are significant for information preservation. In the past, the number of PCs/ICs is determined by the amount of signal energy calculated from data variances that correspond to eigenvalues. Unfortunately, it was shown (Chang, 2003a; Chang and Du, 2004) that using accumulated sums of eigenvalues as a criterion to determine the number of PCs/ICs was not reliable and also not accurate in most cases in hyperspectral imagery. This is because subtle objects such as small targets, anomalies generally contribute little energies to eigenvalues that may not be retained in the first few PCs/ICs. In order to mitigate this dilemma, the concept of virtual dimensionality (VD) developed by Chang (2003b) and Chang and Du (2004) and also detailed in Chapter 5 can serve as a purpose to meet this need. If we assume that each spectrally distinct signature is accommodated by a single PC/IC, then the total number of PCs/ICs required to accommodate all the spectrally distinct signatures will be VD.

19.3.1.2 PCA (ICA)/2D Compression

PCA/2D compression is probably the most popular and commonly used in hyperspectral compression. It first uses PCA to spectrally de-correlate information of second-order statistics among all spectral bands and then followed up by a 2D compression technique to perform spatial compression on each of spectrally de-correlated bands so as to achieve hyperspectral data compression. The number of PCs, denoted by q to be retained can be determined by VD. These q PCs are then compressed by a 2D compression technique as code streams for data transmission. Then the corresponding 2D decompression technique is further applied to de-compress the received transmitted code streams for final exploitation applications. A similar approach using ICA/2D compression can be also implemented in exactly the same fashion that PCA/2D compression does. Figure 19.5 shows a block diagram that describes a process including a new component introduced by VD as a preprocessing prior to PCA/ICA for the purpose of estimating the number of PCs/ICs, q to be retained after spectral dimensionality reduction. The system in Figure 19.5 is referred to as PCA (ICA)/2D-SPHIT and PCA(ICA)/2D-JPEG2000, respectively.

An algorithm to implement the system in Figure 19.5 is described a follows:

PCA(ICA)/2D compression algorithm

1. Determine n_{VD} of an L-band hyperspectral image, q.
2. Apply PCA/ICA to reduce the original data dimensionality to q PCs (ICs).

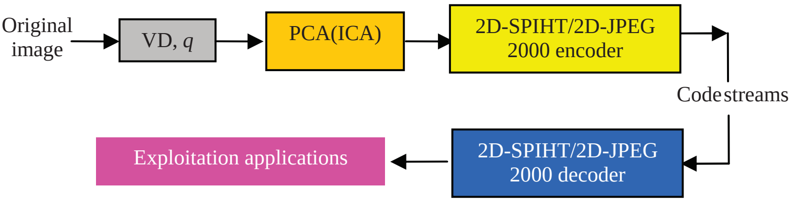

Original image
VD, q
PCA(ICA)
2D-SPIHT/2D-JPEG 2000 encoder
Codestreams
2D-SPIHT/2D-JPEG 2000 decoder
Exploitation applications

Figure 19.5 PCA(ICA)/2D compression system.

3. Use a 2D compression technique such as JPEG2000 or 2D SPIHT to each of q PCs (ICs) and encode them into a set of code streams for data transmission.
4. Implement 2D de-compression corresponding to the one used in step 3 to de-compress the received transmitted code streams to reconstruct the original q-PC (IC) image cube as a q PCA(ICA)-decompressed image cube.
5. Exploit the resulting compressed 3D image cube obtained in step 4 for various applications.

It should be noted that if there is no signal source transmission required, steps 2 and 3 can be skipped and step 4 described in step 5 should be replaced by step 2. Since JPEG2000 and SPHIT have been shown to be most promising and effective compression techniques in the literature and produce nearly the same results, either of these two compression techniques specified in Figure 19.5 can be used for compression.

19.3.1.3 PCA (ICA)/3D Compression

When 2D compression is implemented in conjunction with PCA/ICA in Figure 19.5, a naive assumption is made on the fact that all spectral information can be nearly de-correlated by PCA/-ICA so that the loss of spectral information caused by such a transform can be ignored without significant impact on compression performance. Unfortunately, this is generally not true. First of all, PCA/ICA is usually used to perform spectral dimensionality reduction for de-correlation not necessarily for spectral redundancy removal. As a consequence, using 2D compression techniques can only compress 2D spatial information not spectral information. To address this issue, a 3D compression is needed. Figure 19.6 modifies the block diagram in Figure 19.5 by replacing 2D JPEG and 2D-SPHIT with 3D-Multicomponent JPEG 2000 and 3D-SPHIT, which are referred to as PCA(ICA)/3D-SPIHT, PCA(ICA)/3D-multicomponent JPEG 2000, respectively.

PCA(ICA)/3D compression algorithm

1. Determine n_{VD} of an L-band hyperspectral image, q.
2. Apply PCA/ICA to reduce the original data dimensionality to q PCs (ICs).
3. Use a 3D compression technique such as 3D-multicomponent JPEG2000 or 3D SPIHT to q-PCs (ICs) formed image cube and encode them into a set of code streams for data transmission.
4. Implement 3D de-compression corresponding to the one used in step 3 to de-compress the received transmitted code streams to reconstruct the original q-PC (IC) image cube as a q PCA(ICA)-decompressed image cube.
5. Exploit the resulting compressed 3D image cube obtained in step 4 for various applications.

It should be noted that if there is no signal source transmission required, steps 2 and 3 can be skipped and step 4 described in step 5 should be replaced by step 2.

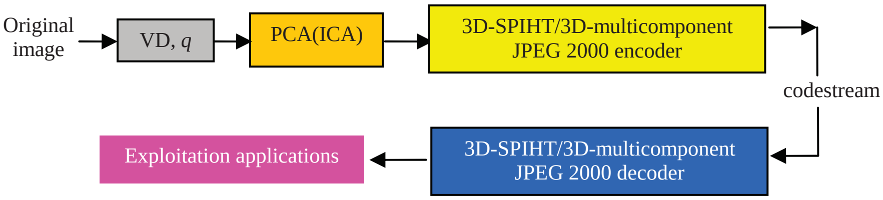

Original image
VD, q
PCA(ICA)
3D-SPIHT/3D-multicomponent JPEG 2000 encoder
codestream
3D-SPIHT/3D-multicomponent JPEG 2000 decoder
Exploitation applications

Figure 19.6 PCA(ICA)/3D compression system.

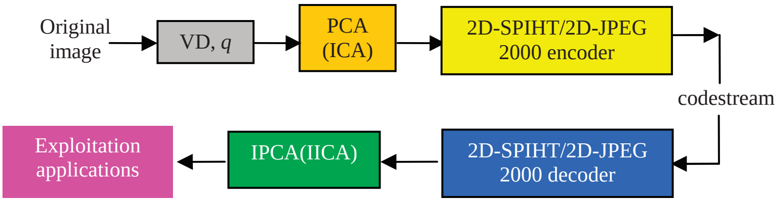

Original image
VD, q
PCA (ICA)
2D-SPIHT/2D-JPEG 2000 encoder
codestream
2D-SPIHT/2D-JPEG 2000 decoder
IPCA(IICA)
Exploitation applications

Figure 19.7 IPCA/2D compression system.

19.3.1.4 Inverse PCA (Inverse ICA)/2D Compression

In Figures 19.5 and 19.6, an exploitation application is directly applied to compressed hyperspectral image data, which is a reduced dimensional image cube formed by q PCs/ICs. As an alternative, an exploitation application can be also applied to a reconstructed image data of the original L-dimensional data space by q PCs/ICs via an inverse transformation. In this case, an inverse transform of PCA (IPCA) or an inverse transform of ICA (IICA) is further applied to a q-PCs/ICs image cube to reconstruct a 3D image that has the same number of spectral bands as the original image has, L. Such an approach is referred to as IPCA (IICA)/2D compression depicted in Figure 19.7 where JPEG 2000 and 2D-SPIHT can be used as 2D compression techniques.

A detailed implementation of IPCA(IICA)/2D compression is summarized as follows.

IPCA(IICA)/2D compression algorithm

1. Determine n_{VD} of an L-band hyperspectral image, q.
2. Apply PCA/ICA to reduce the original data dimensionality to q PCs (ICs).
3. Use a 2D compression technique such as JPEG2000 or 2D SPIHT to each of q PCs (ICs) and encode them into a set of code streams for data transmission.
4. Implement 2D de-compression corresponding to the one used in step 3 to de-compress the received transmitted code streams to reconstruct the original q-PC (IC) image cube as a q PCA(ICA)-decompressed image cube.
5. Apply IPCA(IICA) to the 3D reconstructed image cube obtained in step 4 to reconstruct a 3D L-band image cube.
6. Exploit the resulting compressed 3D image cube obtained in step 5 for various applications.

It should be noted that if there is no signal source transmission required, steps 4 and 5 can be skipped and step 5 described in step 6 should be replaced by step 2 without encoding.

19.3.1.5 Inverse PCA (Inverse PCA)/3D Compression

In analogy with IPCA (IICA)/2D compression, two IPCA (IICA)/3D compression systems can be also implemented, referred to as IPCA (IICA)/3D-SPIHT and IPCA (IICA)/3D-Multicomponent JPEG2000. In other words, PCA/ICA is first used to de-correlate a hyperspectral image for spectral compression. Then VD determines the number of PCs/ICs, denoted by q, which must be retained for compression. Then a 3D compression technique is applied to an image cube formed by q PCs/ICs for further compression. Finally, an inverse transform of PCA/ICA is applied to de-compressed q PCs/ICs-formed image cube to reconstruct a 3D image with the same number of spectral bands as the original image has, L, for exploitation applications. A block diagram of IPCA (IICA)/3D compression is depicted in Figure 19.8.

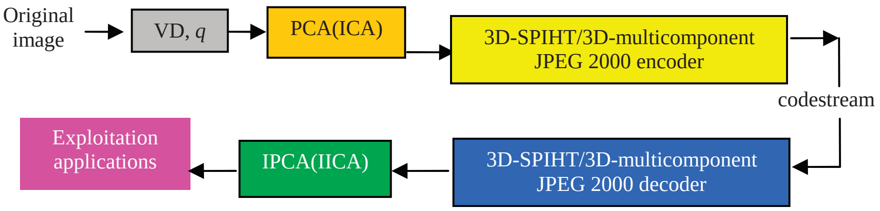

Original image
VD, q
PCA(ICA)
3D-SPIHT/3D-multicomponent JPEG 2000 encoder
codestream
3D-SPIHT/3D-multicomponent JPEG 2000 decoder
IPCA(IICA)
Exploitation applications

Figure 19.8 IPCA/3D compression system.

The details of implementing IPCA (IICA)/3D compression are briefly described as follows.

IPCA (IICA)/3D compression algorithm

1. Determine n_{VD} of an L-band hyperspectral image, q.
2. Apply PCA/ICA to reduce the original data dimensionality to q PCs (ICs).
3. Use a 3D compression technique such as 3D-multicomponent JPEG2000 or 3D SPIHT to q-PCs (ICs) formed image cube and encode them into a set of code streams for data transmission.
4. Implement 2D de-compression corresponding to the one used in step 3 to de-compress the received transmitted code streams to reconstruct the original q-PC (IC) image cube as a q PCA(ICA)-decompressed image cube.
5. Apply IPCA(IICA) to the 3D reconstructed image cube obtained in step 4 to reconstruct a 3D L-band image cube.
6. Exploit the resulting compressed 3D image cube obtained in step 5 for various applications.

Furthermore, if there is no signal source transmission required, steps 4 and 5 can be skipped and step 5 described in step 6 should be replaced by step 2 without encoding. In addition, two 3D compression techniques, 3D-SPIHT and 3D-multicomponent JPEG2000, can be used in the above algorithm. However, since 3D SPIHT requires dimensions to be multiples of 2^{n+1} with n being the number of levels in wavelet decomposition, IPCA/3D-SPIHT may not be applicable when the number of spectral bands does not meet this constraint.

19.3.1.6 Mixed Component Transforms for Hyperspectral Compression

As noted, the sample covariance matrix used by PCA is of second-order statistics. PCA is considered as a second-order statistics transform that can only preserve information characterized by second-order statistics through transformation. In many applications preserving information of second-order statistics is generally not sufficient in substance characterization such as small objects, rare targets, etc, which cannot be generally captured by second-order statistics, but rather by statistics of order higher than 2. Under such a circumstance, PCA may not be effective. By contrast, ICA is developed to capture information characterized by statistical independence that may help to resolve this dilemma. For ICA to be effective two assumptions must be satisfied. One crucial assumption is that all the signal sources must be random sources. Since a linear sum of a finite number of Gaussian signal sources is still Gaussian, a second critical assumption is that at most one signal source can be Gaussian. However, due to this particular assumption, ICA is able to capture information that is characterized by non-Gaussianity whose statistics goes beyond second-order statistics. Because of that, PCA and ICA actually perform mutual disjoint transformations and complement each other. This leads to a brief that combining and mixing both PCA and ICA to form a single transform may yield better compression and desired performance.

In order to investigate such a mixed (m,n)-PCA/ICA transform that combines the first m PCs with n ICs, three issues need to be addressed (Chai et al. 2007). One is how many components are required for such a mixed (m,n)-PCA/ICA transform. The second issue is what n ICs should be selected for the (m,n)-PCA/ICA transform. Unlike PCA, which ranks PCs according to eigenvalues with decreasing order, ICA does not prioritize the ICs it generates. Selecting appropriate n ICs is crucial for the (m,n)-PCA/ICA transform. The third issue is how to combine two different sets of projection vectors, PCA-generated eigenvectors and ICA-generated projection vectors, that are not necessarily same vectors. Each of these three issues will be resolved as follows.

The first issue can be addressed by VD that has been used in PCA and ICA discussed in previous sections. With the help of VD, we can assume that VD-estimated $n_{\mathrm{VD}} = q$ is the total number of components needed in the (m,n)-PCA/ICA transform with $q = m + n$.

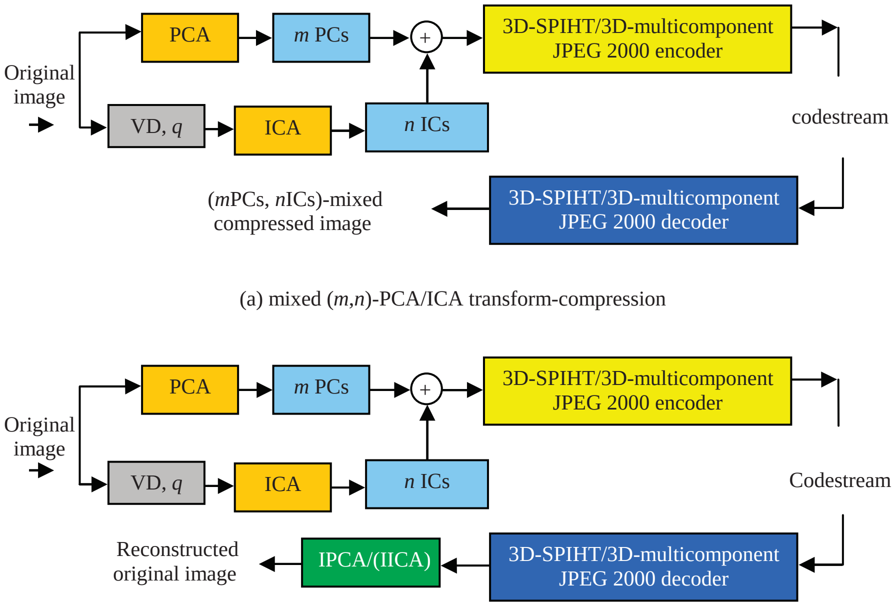
The second issue is to rank ICs by different criteria in a similar manner that PCA does for its PCs using variance as a criterion. This issue is addressed in Chapter 6. Of particular interest is the automatic target generation process (ATGP) that will be used in the proposed (m,n)-PCA/ICA transform.

To address the third issue, it first determines how many PCs resulting from PCA will be selected, denoted by m, and let $\{\mathbf{v}_i\}_{i=1}^{m}$ be the eigenvectors that generate the first m PCs. Then all data samples are then projected to a space orthogonal to the space spanned by the m eigenvectors $\{\mathbf{v}_i\}_{i=1}^{m}$. Assume that this resulting orthogonal subspace is denoted by $\left\langle \{\mathbf{v}_i\}_{i=1}^{m} \right\rangle^{\perp}$. Then the ATGP-based ICA is applied to the space $\left\langle \{\mathbf{v}_i\}_{i=1}^{m} \right\rangle^{\perp}$ to find the first n ICs with their corresponding projection vectors, denoted by $\{\mathbf{z}_i\}_{i=1}^{n}$. Combining the m eigenvectors $\{\mathbf{v}_i\}_{i=1}^{m}$ with the n projection vectors $\{\mathbf{z}_i\}_{i=1}^{n}$ yields a new set of basis vectors $\{\mathbf{w}_i\}_{i=1}^{p}$ for our desired (m,n)-PCA/ICA transform where $\mathbf{w}_i = \mathbf{v}_i$ for $1 \le i \le m$, $\mathbf{w}_{i+m} = \mathbf{z}_i$ for $1 \le i \le n$ and $q = m + n$.

Once all the three issues are resolved, a mixed (m,n)-PCA/ICA transform can be developed for spectral/spatial compression as follows.

Mixed (m,n)-PCA/ICA compression algorithm

1. Use n_{VD} to estimate the number of components needed to be retained for spectral compression, denoted by q.
2. Perform PCA to find eigenvalues $\{\lambda_l\}_{l=1}^{L}$ and their corresponding eigenvectors $\{\mathbf{v}_l\}_{l=1}^{L}$ and retain m PCs with the m largest eigenvalues.
3. Define an $(L-m) \times (L-m)$ eigenvalue diagonal matrix by $\mathbf{D} = \begin{bmatrix} \lambda_{m+1} & 0 & \mathbf{0} \\ 0 & \ddots & 0 \\ \mathbf{0} & 0 & \lambda_L \end{bmatrix}$ and an $L \times (L-m)$ eigenvector matrix by $\mathbf{E} = [\mathbf{v}_{m+1}\mathbf{v}_{m+2}\cdots\mathbf{v}_L]$. A whitening matrix $\mathbf{D}^{-1/2}\mathbf{E}^T$ is then used to sphere the mean-removed data matrix X. Let the resulting matrix $\hat{\mathbf{X}} = \mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{X}$ be denoted by $\hat{\mathbf{X}}$.
4. Apply FastICA using the $n = p - m$ ATGP-generated pixel vectors as initial projection vectors to the sphered data $\hat{\mathbf{X}}$ to generate n projection vectors denoted by $\left\{\mathbf{z}_j\right\}_{j=1}^{n}$ as well as $p-1$ ICs. Let Z denote the projection matrix formed by $\mathbf{Z} = [\mathbf{z}_1\mathbf{z}_2\cdots\mathbf{z}_n]$ with dimensionality of $(L-m) \times n$ and define $\hat{\mathbf{Z}} = \mathbf{Z}^T\mathbf{D}^{-1/2}\mathbf{E}^T$.
5. Form a new image cube Y by the m PCs and the n ICs and let $\mathbf{W} = \left[\mathbf{v}_1\hat{\mathbf{Z}}\right]$.
6. Apply IPCA, $\mathbf{X} = \mathbf{W}^{-1}\mathbf{Y}$and add the mean back to reconstruct X. Since q is generally much smaller than L, $n < L$, in which case the inverse of W is always taken as its pseudo inverse.


PCA
m PCs
+
3D-SPIHT/3D-multicomponent JPEG 2000 encoder
Original image
VD, q
ICA
n ICs
codestream
(mPCs, nICs)-mixed compressed image
3D-SPIHT/3D-multicomponent JPEG 2000 decoder
PCA
m PCs
+
3D-SPIHT/3D-multicomponent JPEG 2000 encoder
Original image
VD, q
ICA
n ICs
Codestream
Reconstructed original image
IPCA/(IICA)
3D-SPIHT/3D-multicomponent JPEG 2000 decoder

(a) mixed (m,n)-PCA/ICA transform-compression

(b) mixed (m,n)-PCA/ICA transform-decompression

Figure 19.9 Structure of mixed (m,n)-PCA/ICA transform.

Figure 19.9(a) depicts a block diagram of mixed (m,n)-PCA/ICA transform for data compression where the I(PCA/ICA) in Figure 19.9(b) denotes the inverse of mixed (m,n)-PCA/ICA transform via the projection vector $\{\mathbf{w}_i\}_{i=1}^{p}$. It should be noted that if there is no signal source transmission required, the process of encoding-decoding and reconstruction image described in the lower part of the diagram can be skipped.

19.3.2 Dimensionality Reduction by Band Selection-Based Spectral Compression

Another approach to spectral dimensionality reduction is band selection, called DRBS. Since the same treatment carried out for DRT can be applied to DRBS, only the implementations of algorithms are described as follows. However, it should be noted that since there are no images that can be reconstructed in the original data space from BS-compressed q-band images, no similar algorithms corresponding to Figures 19.7, 19.8 and 19.9(b) can be derived for BS/2D or 3D compression.

BS/2D compression algorithm

1. Determine n_{VD} of an L-band hyperspectral image, p.
2. Apply a BS technique to select p bands.
3. Use a 2D compression technique such as JPEG2000 or 2D SPIHT to each of p band images and encode them into a set of code streams for data transmission.
4. Implement 2D de-compression corresponding to the one used in step 3 to de-compress the received transmitted code streams to decompress the original p-band image cube as a new reconstructed p-band image cube.
5. Exploit the resulting compressed 3D image cube obtained in step 4 for various applications.

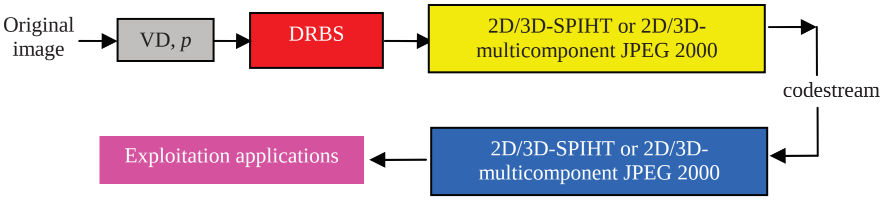

Original image
VD, p
DRBS
2D/3D-SPIHT or 2D/3D-multicomponent JPEG 2000
codestream
2D/3D-SPIHT or 2D/3D-multicomponent JPEG 2000
Exploitation applications

Figure 19.10 DRBS/3D compression process.

BS/3D-cube compression algorithm

1. Determine n_{VD} of an L-band hyperspectral image, p.
2. Apply a BS technique to select p bands.
3. Form the p band images obtained in step 2 as a 3D image cube, referred to as 3D p-band image cube.
4. Use a 3D compression technique such as 3D-multicomponent JPEG2000 or 3D SPIHT to the 3D p-band image cube and encode them into a set of code streams for data transmission.
5. Implement 3D de-compression corresponding to the one used in step 4 is de-compress the received transmitted code streams to reconstruct the original p-band image cube as a new reconstructed p-band image cube.
6. Exploit the resulting compressed 3D image cube obtained in step 5 for various applications.

Figure 19.10 depicts a block diagram of the BS/2D or 3D spectral/spatial compression. It should be noted that if there is no signal source transmission required, the process of encoding-decoding described in the diagram can be skipped.

19.4 Progressive Spectral/Spatial Compression

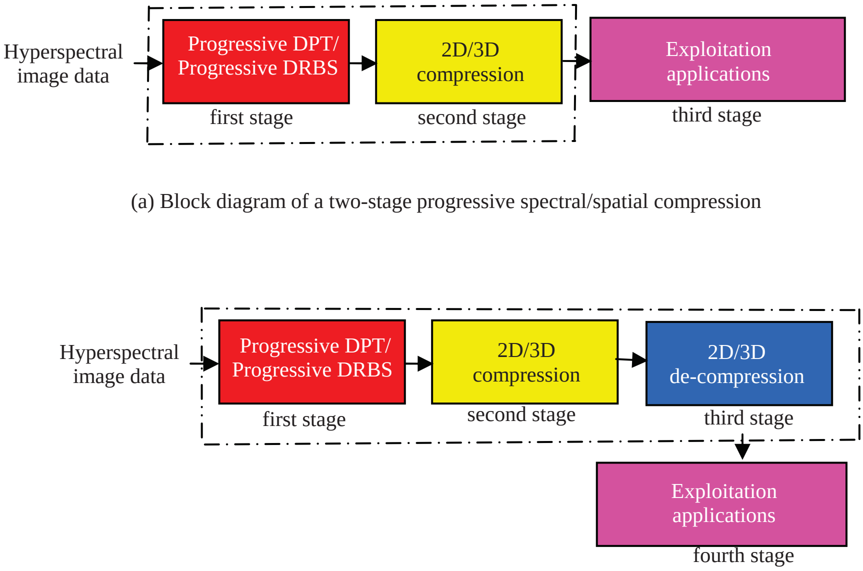
The spectral/spatial compression presented in Section 19.3 requires the knowledge of q that is the number of dimensions needed to be retained for DR and p that is the number of bands needed to be selected for BS to achieve spectral compression. Despite that VD can be used to estimate the values of the q and p, both the q and p in fact vary with applications as discussed in Chapter 5. In order to avoid dealing with the issue of determining the q and p, an alternative approach called progressive spectral compression for DRT and DRBS will be developed in Chapters 20–21 to perform DRT and BS progressively via dimensionality prioritization and band prioritization, respectively. Its idea can be depicted in Figure 19.10(a) and (b), which is very similar to Figures 19.5–19.9 except DRT/DRBS replaced by progressive DRT/progressive DRBS (Figure 19.11).

19.5 3D Compression

Many 3D-cube image compression techniques are generally extended directly from their 2-D counterparts. Two 3D-cube compression techniques of particular interest that will be used in this chapter are JPEG2000 Multicomponent (ISO, 2000b) that is an extension of wavelet-based 2D-JPEG2000 (Taubman and Marcellin, 2000) and 3D-SPIHT, which is extended by 2D-SPIHT developed by Said and Pearlman (1996).

19.5.1 3D-Multicomponent JPEG

JPEG2000 (Taubman and Marcellin, 2000; Rucker et al., 2005) is a new still image compression standard that has replaced the commonly used DCT-based JPEG. It is a wavelet-based compression



(a) Block diagram of a two-stage progressive spectral/spatial compression

(b) Block diagram of a three-stage progressive spectral/spatial compression

Figure 19.11 Progressive spectral/spatial compression.

technique that adds/improves features such as coding of regions of interest, progressive coding, scalability, etc. The entire coding can be divided into four stages: tiling, discrete wavelets transform (DWT), scalar quantization, and block coding. The image is divided into rectangular regions called tiles; each tile gets encoded separately. The purpose of dividing images into tiles is that the decoder needs to decode only certain parts of the image on demand, instead of decoding the entire image and also less memory will be needed by the decoder to decode the image. After dividing the image into tiles, a wavelet transform is applied to each tile. The wavelet transform is followed by scalar quantization to quantize the sub-bands. The scalar quantized sub-bands representing different scales are coded using embedded block coding with block truncation (EBCOT) (Taubman and Marcellin, 2000; Rucker et al., 2005; ISO, 2000a; ISO, 2000b; Taubman, 2000). For the case of hyperspectral imagery the Part II of JPEG2000 (ISO, 2000b) is implemented to allow multi-component image compression that involves grouping of arbitrary subsets of components into component collections and applying point transforms along the spectral direction like wavelet transform. The postcompression rate-distortion optimizer of EBCOT is simultaneously applied to all code blocks across all the components.

19.5.2 3D-SPIHT Compression

Recently, an approach developed by Said and Pearlman (1996), called set partitioning in hierarchical trees (SPIHT) has become popular. Two main features introduced by Shapiro (1993) are used in the SPIHT algorithm. First, it utilizes a partial ordering of coefficients by magnitude and transmits the most significant bits first. Second, the ordering data are not explicitly transmitted. The decoder running the same algorithm can trace the ordering information from the transmitted information. Kim et al. (2000) later extended 2D-SPIHT to 3D-SPIHT for video compression in a relatively straightforward manner. There is no constraint imposed on the SPIHT algorithm regarding the dimensionality of the data. If all pixels are lined up in decreasing order of magnitude, 3D-SPIHT performs exactly the same as 2D-SPIHT. In the case of 3D sub-band structure, one can use a wavelet packet transform to allow a different number of decompositions between the spatial and spectral dimensions.

19.6 Exploration-Based Applications

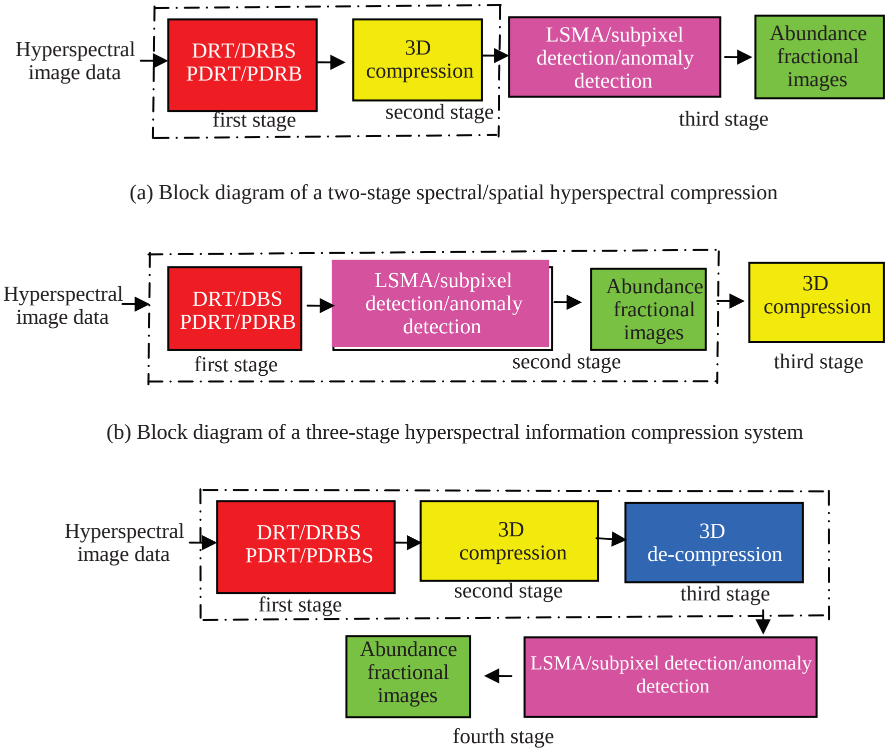
The exploitation-based applications are a key component in hyperspectral information compression proposed in Figures 19.2(a), (b), and 19.3 because the other two components, DR/BS and 3D compression, are developed to support and enhance its functionalities. It is this component to make a hyperpscetral information system versatile and adaptive. Its success is determined by the exploitation criterion used for hyperspectral information compression that represents the information extracted from data for future information retrieval and data processing. While considering all possible exploitation applications is impossible, in this section we describe only four exploitation applications of particular interest in hyperspectral image analysis, anomaly detection, subpixel target detection, spectral unmixing, and endmember extraction, each of which requires a different level of target information to be compressed. For example, anomaly detection and endmember extraction requires no target information at all compared to the spectral unmixing that needs complete target knowledge of image endmembers in the data. The subpixel target detection is somewhere in between and needs only the target information of interest while discarding all other target knowledge.

19.6.1 Linear Spectral Mixture Analysis

Linear spectral mixture analysis (LSMA) has been widely used to perform spectral unmixing. For LSMA to work well, we need to find an appropriate set of image endmembers present in the data that form a base of linear mixture model for spectral unmixing. There are two ways to acquire information of these image endmembers. One is provided by *a priori* knowledge and another is obtained directly from the data in an unsupervised manner (see Chapter 17). Using LSMA to compress hyperspectral imagery can achieve significant compression ratios because the crucial information has been preserved in the unmixed images, referred to as fractional abundance images that represent abundance fractions of these image endmembers for future fast information processing. In this case, directly dealing with these abundance fractions may be more effective than working on the entire data. Two of major advantages resulting from such an LSMA-compression are that (1) the image background, which is generally of no interest, has been significantly suppressed and (2) the interpixel spatial correlation in these abundance fractional images has been substantially reduced, which makes the follow-up 3D compression work more effectively to compress only the abundance fractional images (Figure 19.12).

19.6.2 Subpixel Target Detection

Subpixel target detection plays a crucial role in applications of reconnaissance, search and rescue, and identification of targets of interest. Unlike LSMA, subpixel target detection does not require complete signature knowledge to be used to form a linear mixing model. It has specific targets to be detected, which must be provided *a priori*. It searches targets of interest in an unknown environment that generally has very complicated background which may hinder the searching process. So, when compression is performed, such unwanted background is certainly not desired. Instead, the information of targets must be preserved by full knowledge. Obviously, conventional measures



(a) Block diagram of a two-stage spectral/spatial hyperspectral compression

(b) Block diagram of a three-stage hyperspectral information compression system

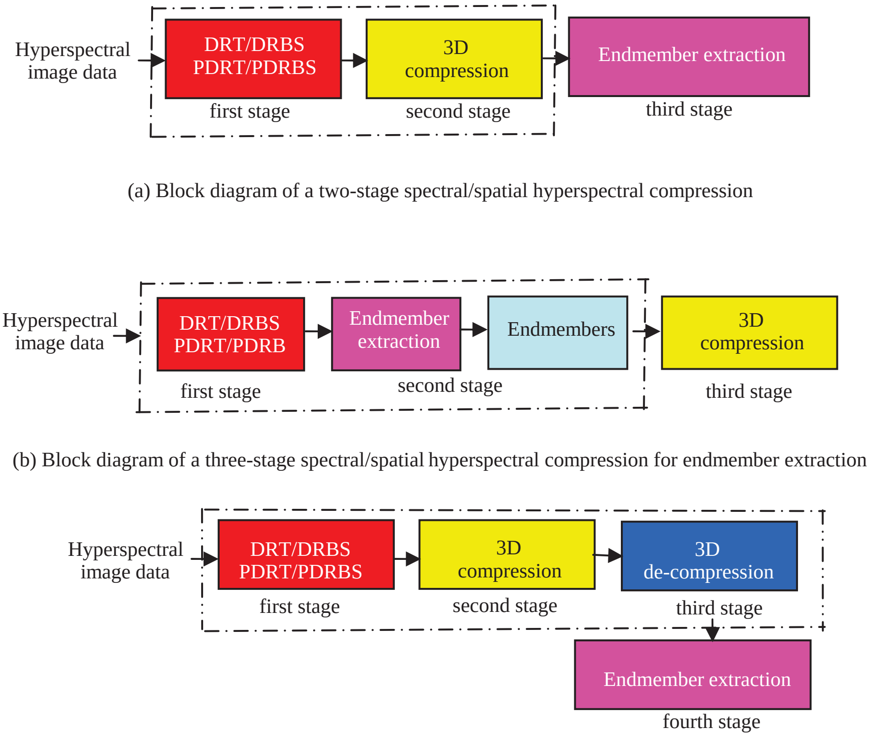
(c) Block diagram of a four-stage spectral/spatial hyperspectral information compression system

Figure 19.12 Block diagrams for hyperspectral information compression systems for LSMA.

used for data compression such as mean squared error (MSE) and signal-to-noise ratio (SNR) are not appropriate for subpxiel detection since the targets of interest in subpxiel detection generally contribute very little to MSE and SNR. When it comes to compression, extra care and caution must be taken to ensure that the target information should not be sacrificed by any spatial compression technique.

19.6.3 Anomaly Detection

Anomaly detection detects unknown targets without prior knowledge. It has been discussed in Chang (2003b). It neither performs mixture analysis as does LSMA in spectral unmixing nor performs detection of specific targets as does the subpxiel detection in reconnaissance applications. Instead, it searches for unknown targets of interest that cannot be identified by prior knowledge or visual assessment *a priori*. Its major functionality is in surveillance applications where the targets of interest are generally unknown and insignificant in terms of its spatial extent in presence that cannot be visually inspected. Examples include special spices in agriculture and ecology, rare minerals in geology, toxic waste in environmental monitoring, combat vehicles in battlefield, landmine detection in combat zone, drugging trafficking in law enforcement, chemical/biological agent detection in bioterrorism, terrorist activities in intelligent gathering, tumor/cancer detection in medical diagnosis, and so on. These targets usually appear unknowingly with very low probabilities. When they do occur, their size is relatively small and their sample pool is also very limited.



(a) Block diagram of a two-stage spectral/spatial hyperspectral compression

(b) Block diagram of a three-stage spectral/spatial hyperspectral compression for endmember extraction

(c) Block diagram of a four-stage spectral/spatial hyperspectral compression for endmember extraction

Figure 19.13 Block diagrams for hyperspectral image compression for endmember extraction.

However, these are the targets of major interest to hyperspectral image analysts and cannot be compromised by any compression means. On the other hand, when compression is performed, only the information of these targets needs to be preserved while other information can be suppressed. Analogous to subpixel detection, such targets of interest generally do not contribute much to MSE and SNR. As a consequence, a direct compression without accounting for anomaly detection may result in significant loss of the desired target information. This justifies the need of exploitation-based compression.

19.6.4 Endmember Extraction

Endmember extraction is one of fundamental preprocesses in hyperspectral data exploitation and provides basic understanding of hyperspectral imagery directly from the data itself without assumed prior knowledge (Figure 19.13). A great deal of discussions on endmember extraction is already presented in Part II, Chapters 7–11 and will not be discussion here to avoid replication.

19.7 Experiments

All the exploitation-based spectral/spatial compression techniques presented in previous sections are carried out in two stages, that is, VD-determined spectral compression in the first stage followed by either JPEG2000 Multicomponent or 3D-SPIHT spatial compression in the second stage. For the spatial compression, a variable bit-rate lossy compression technique is used in both JPEG2000 Multicomponent and 3D-SPIHT. Since PCA and ICA transforms generate real numbers, we have rounded these numbers to 16 bits in the implementation of spectral/spatial compression techniques. Also, PCA and ICA are data dependent transforms; thus, their component projection vectors need to be stored and/or transmitted in order to perform reconstruction of the data. This factor is considered as an overhead and further included in calculation of compression ratio.

The compression ratios are chosen to be 20, 40, 60, and 100 because little difference is noted in the detection/quantification performance for compression ratios lower than 20. This implies that for very low compression ratios (<10) 3D-cube compression alone and spectral/spatial based compression can successfully preserve the subpixel and mixed pixel information. Such subtle difference can be only observed when the data are compressed with high compression ratios (>40). In order to address the issues of subpixels and mixed pixels, two examples are custom-designed to illustrate and demonstrate the superiority of spectral/spatial compression over 3D-cube compression in terms of preserving subpixel and mixed pixel spectral information.

19.7.1 Synthetic Image Experiments

The first example was designed to investigate the issue of subpixel quantification of subtle targets (weak signals) embedded in a single background, in which case both PCA- and ICA-based compression techniques worked well compared to 3D-cube compression alone. As a matter of fact, it was demonstrated in Ramakrishna et al. (2006) that for subpixel detection of subtle targets (weak signals) over a single background ICA-based compression techniques worked the best. The second example was designed to investigate the issue of subpixel and mixed pixel quantification of strong targets (strong signals) embedded in multiple backgrounds, in which case ICA did not work as expected and the best results were obtained by using PCA-based compression.

EXAMPLE 19.1

(Single Background)

The synthetic image to be used for our experiments in this example is shown in Figure 19.14 that is similar to the real HYDICE scene in Figure 1.15(a) and has the same size of 64 × 64 pixel vectors.

The background in the synthetic image is simulated by $\mathbf{p}_1$ panel signature from the image scene in Figure 1.17 with an added Gaussian noise to achieve signal-to-noise ratio (SNR) 30:1. There are 16 panels located at the center and arranged in four rows with four panels in each row. The four panels in the *i*th row and the first column composed of single pure pixels, denoted by $p_{i,11}p_{i,12}$, $p_{i,21}$, and $p_{i,22}$ pixel vectors

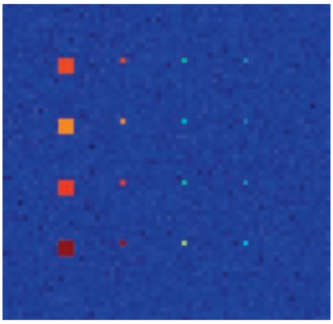


Figure 19.14 Sixteen panels implanted in a single background.

Table 19.1 Abundance fractions (%) of the panels in five rows

	$(p_{i,11}, p_{i,12}, p_{i,21}, p_{i,22})$				(p_{i2})	(p_{i3})	(p_{i4})
Panel pixels	$p_{1,11}$	$p_{1,21}$	$p_{1,21}$	$p_{1,22}$	p_{12}	p_{13}	p_{14}
$\mathbf{p}_2$	100	100	100	100	75	50	25
Panel pixels	$p_{2,11}$	$p_{2,21}$	$p_{2,21}$	$p_{2,22}$	p_{22}	p_{32}	p_{24}
$\mathbf{p}_3$	100	100	100	100	75	50	25
Panel pixels	$p_{3,11}$	$p_{3,21}$	$p_{3,21}$	$p_{3,22}$	p_{32}	p_{33}	p_{34}
$\mathbf{p}_3$	100	100	100	100	75	50	25
Panel pixels	$p_{4,11}$	$p_{4,21}$	$p_{4,21}$	$p_{4,22}$	p_{42}	p_{43}	p_{44}
$\mathbf{p}_4$	100	100	100	100	75	50	25

simulated by $\mathbf{p}_i$ from Figure 1.16 for $i = 1, 2, 3, 4$, respectively. The single panel in the *i*th row and the second column is a single-pixel panel, denoted by p_{i2} simulated by $\mathbf{p}_i$ with abundance 75%. The single panel in the *i*th row and the third column are single-pixel panel, denoted by p_{i3} simulated by $\mathbf{p}_i$ with abundance 50%. The single panel in the *i*th row and the fourth column is a single-pixel panel, denoted by p_{i4} simulated by $\mathbf{p}_i$ with abundance 25%. Figure 19.14 shows a synthetic image obtained by implanting the 16 simulated panels in the background image where their corresponding background pixels are removed to accommodate the panel pixels. It should be noted that the noise background in Figure 19.14 has been visually suppressed because of high-intensity gray level values of panel pixels. Table 19.1 shows the subpixel abundance fractions of the panels in five rows for the different compression techniques over different compression ratios.

Clearly, the synthetic image in Figure 19.14 is composed of five different classes including four panel signatures $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, $\mathbf{p}_5$ and one background class $\mathbf{p}_1$. To verify this number, the virtual dimensionality (VD) in Chapter 5 was used to determine the number of spectrally distinct signatures present in this synthetic image by the Harsanyi–Farrand–Chang (HFC) method developed in Harsanyi et al. (1994). It was 5 across all false alarm probabilities, which was exactly the same number of spectrally distinct signatures according to the ground truth. The VD-estimated value, $n_{\text{VD}} = 5$ provided the necessary knowledge about how many components needed to be retained when dimensionality reduction was performed, provided that each component can be used to accommodate one distinct signature. In the case of Figure 19.14, the number of components required after dimensionality reduction was 5.

In order to demonstrate the issues of subpixels and mixed pixels caused by lossy data compression, the unsupervised fully constrained least squares (UFCLS) developed by Heinz and Chang (2001) in Chapter 8 was used to unmix the abundance fractions of all subpixels and mixed pixels in Figure 19.14. Since the results obtained for panels in each of five rows were very similar, Table 19.2 only tabulates the abundance fractions of the second, third, and the fourth single pixel panels in the first row of the synthetic image obtained by 3D-cube compression and spectral/3D compression techniques. From Table 19.2 the performance of 3D SPIHT seemed acceptable for CR = 20, 40, but its performance for CR = 60 and 100 was poor. The performance of JPEG 2000 Multicomponent was very poor in all the cases. On the other hand, the spectral/3D compression techniques except ICA/JPEG2000 Multicomponent performed well under all compression ratios where the crucial subpixel and mixed pixel information was well preserved through spectral compression with the spatial compression only causing very little or no deterioration of subpixel and mixed pixel information. In fact, the compression ratios did not seem to have impact on their compression performance. However, despite the fact that ICA/JPEG2000 Multicomponent did not perform as well as the other three spectral/3D compression techniques did, it is interesting to note that its inverse counterpart, IICA/JPEG2000 Multicomponent, did perform very well. This implies that ICA/JPEG2000 Multicomponent tended to over-unmix the abundance fractions by over-suppressing the background. In addition, the use of inverse ICA-reconstructed images seemed to be able to correct the issue of over-unmixed abundance fractions resulting from compression.

Table 19.3 also calculates SNR and MSE for the two spectral/3D compression techniques, IPCA/JPEG2000 Multicomponent and IICA spectral/ JPEG2000 Multicomponent along with 3D-SPIHT, JPEG2000 Multicomponent for CR = 100, 60, 40, and 20.

Table 19.2 Abundance fractions (%) for the second, third, and the fourth single pixel panels in the first row for different compression ratios

Compression ratio (CR)	100			60			40			20		
Panels	p_{12}	p_{13}	p_{14}	p_{12}	p_{13}	p_{14}	p_{12}	p_{13}	p_{14}	p_{12}	p_{13}	p_{14}
original image	75	50	25	75	50	25	75	50	25	75	50	25
3D-SPIHT	66	39	13	67	47	20	70	50	24	73	50	24
JPEG2000 Multicomponent	61	25	9	67	36	17	70	42	20	71	45	21
PCA/JPEG2000 Multicomponent	75	49	24	75	49	24	75	49	24	75	49	24
ICA/JPEG2000 Multicomponent	75	59	35	75	58	34	75	58	34	75	58	34
IPCA/JPEG2000 Multicomponent	75	49	24	75	49	24	75	49	24	75	49	24
IICA/JPEG2000 Multicomponent	75	49	24	75	49	24	75	49	24	75	49	24

It should be noted that in order to make fair compression, no MSE and SNR were calculated for PCA/JPEG2000 Multicomponent and ICA/JPEG2000 Multicomponent because they were performed in reduced dimensions. As shown in Table 19.3, IICA yielded the worst MSE and SNR, but its performance was among the best in terms of subpixel quantification according to Table 19.2. The reason for this was because the image background was largely suppressed by ICA and much of background information was lost in the image reconstruction by IICA.

EXAMPLE 19.2

(Multiple Backgrounds)

Example 19.1 demonstrated that JPEG2000 Multicomponent did not perform well for all compression ratios, but 3D-SPIHT did reasonably well in some cases such as CR = 20, 40. In the following example, we will further show that both 3D-SPIHT and JPEG2000 Multicomponent fail to address the issues of subpixels and mixed pixels. In doing so, the synthetic image shown in Figure 19.15 was custom designed to address the inability of 3D compression such as 3D-SPIHT and JPEG2000 Multicomponent in preserving quantitative information provided by subpixels and mixed pixels.

Table 19.3 SNR and MSE values for different compression techniques

Compression ratio (CR)	100		60		40		20	
	SNR	MSE	SNR	MSE	SNR	MSE	SNR	MSE
3D - SPIHT	37.65	1.44E + 05	37.94	1.35E + 05	38.35	1.23E + 05	39.53	9.32E + 04
JPEG2000 Multicomponent	37.54	1.48E + 05	37.80	1.40E + 05	38.13	1.29E + 05	39.28	9.86E + 04
IPCA/JPEG2000 Multicomponent	37.74	1.42E + 05	37.74	1.42E + 05	37.74	1.42E + 05	37.74	1.42E + 05
IICA/JPEG2000 Multicomponent	22.41	4.98E + 06	22.41	4.98E + 06	22.41	4.98E + 06	22.41	4.98E + 06

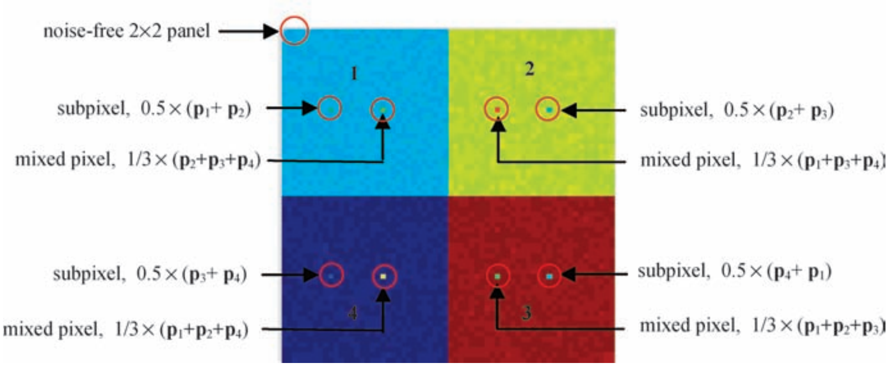

noise-free 2×2 panel
1
2
subpixel, $0.5\times(\mathbf{p}_1+\mathbf{p}_2)$
mixed pixel, $1/3\times(\mathbf{p}_2+\mathbf{p}_3+\mathbf{p}_4)$
subpixel, $0.5\times(\mathbf{p}_2+\mathbf{p}_3)$
mixed pixel, $1/3\times(\mathbf{p}_1+\mathbf{p}_3+\mathbf{p}_4)$
subpixel, $0.5\times(\mathbf{p}_3+\mathbf{p}_4)$
mixed pixel, $1/3\times(\mathbf{p}_1+\mathbf{p}_2+\mathbf{p}_4)$
4
3
subpixel, $0.5\times(\mathbf{p}_4+\mathbf{p}_1)$
mixed pixel, $1/3\times(\mathbf{p}_1+\mathbf{p}_2+\mathbf{p}_3)$

Figure 19.15 Synthetic imagery for Example 2 showing the four quadrants and the subpixel and mixed pixel implanted in each.

The image in Figure 19.15 was similar to the one used in Figure 19.14 in Example 19.1 with the same image background the same size of 64 × 64 pixel vectors, but had different inserted panel pixels indicated in Figure 19.15. Specifically, the image is divided into four quarters, each of which had its own background composed of a different panel signature where the background in each of the four quarters was simulated by one of $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, respectively, with an added Gaussian noise to achieve signal-to-noise ratio (SNR) 30:1. The background pixels in a 2 × 2 square panel marked by a circle at the upper left corner had no noise added. In each of these quarters a subpixel and a mixed pixel with specific abundance fractions were implanted by replacing the original pixels. There were two single one-pixel panels implanted in each background marked by circles in Figure 19.15. The difference between images in Figures 19.14 and 19.15 is that the image background in Figure 19.15 was made up of four different signatures instead of a single image background signature in Figure 19.14. Table 19.4 tabulates the abundance fractions of subpixels and mixed pixels in the four quadrant simulated by four panel spectral signatures $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$ from Figure 1.16.

It should be noted that a subpixel target simulated was defined as a target with fractional abundance embedded in a background (B). For example, in the first quadrant the first panel contained a subpixel with target signature $\mathbf{p}_2$ of fractional abundance 50% with the background signature $\mathbf{p}_1$ of 50% and the second panel is a mixed pixel with the three signatures $\mathbf{p}_2$, $\mathbf{p}_3$, and $\mathbf{p}_4$ sharing the same fractional abundance 1/3.

Table 19.4 Abundance fractions (%) of the implanted panels in each of the four quadrants

Quadrant	Pixels	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$
1	Panel 1–subpixel	50 (B)	50 (target)	0	0
	Panel 2–mixed pixel	0	1/3	1/3	1/3
2	Panel 1–subpixel	0	50 (B)	50 (target)	0
	Panel 2– mixed pixel	1/3	0	1/3	1/3
3	Panel 1– subpixel	0	0	50 (B)	50 (target)
	Panel 2– ixed pixel	1/3	1/3	0	1/3
4	Panel 1– subpixel	50 (target)	0	0	50 (B)
	Panel 2– mixed pixel	1/3	1/3	1/3	0

Table 19.5 Abundances fractions (%) for the subpixel and the mixed pixel panels in the first quadrant using UFCLS unmixing for different compression ratios

Panels		Subpixel				Mixed			
CR		100	60	40	20	100	60	40	20
Original	p_1	50	50	50	50	0	0	0	0
	p_2	50	50	50	50	33	33	33	33
	p_3	0	0	0	0	33	33	33	33
	p_4	0	0	0	0	33	33	33	33
3D-SPIHT	p_1	70	61	61	55	35	25	25	11
	p_2	25	36	36	43	11	17	17	24
	p_3	5	3	3	2	29	31	31	33
	p_4	0	0	0	0	25	27	27	31
JPEG2000	p_1	78	68	68	61	46	20	17	10
Multicomponent	p_2	8	23	23	33	13	30	31	35
	p_3	12	7	8	6	23	24	25	26
	p_4	2	2	1	0	18	26	28	29
PCA/JPEG2000	p_1	49	48	48	48	1	1	1	1
Multicomponent	p_2	45	47	47	47	32	32	32	32
	p_3	4	4	4	4	34	34	34	34
	p_4	1	1	1	1	33	33	33	33
IPCA/JPEG2000	p_1	49	50	50	50	0	0	0	0
Multicomponent	p_2	51	50	50	50	33	33	33	33
	p_3	0	0	0	0	33	33	33	33
	p_4	0	0	0	0	33	33	33	33

It is worth noting that this example was particularly designed so that ICA would not be applicable because the image backgrounds in the four quadrants were simulated by different Gaussian noises, in which case ICA could not unmix the simulated four Gaussian noises.

Once again, VD was used to estimate the number of spectrally distinct signatures for this scene, which was 4 across all false alarm probabilities $P_F = 10^{-1}$, 10^{-2}, 10^{-3}, and 10^{-4}. This was different from the value of 5 estimated by VD for the scene in Figure 19.14. The value of 4 estimated by VD was exactly the same number of panel signatures. $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$ used to simulate the image in Figure 19.15 and the original dimensionality was reduced to 4, i.e., $q = 4$.

Now, UFCLS was used to unmix the abundance fractions of all subpixels and mixed pixels. Table 19.5 tabulates UFCLS-unmixed abundance fractions for the subpixel and mixed pixel panels.

Apparently, the compression performance of 3D SPIHT and JPEG2000 Multicomponent was very poor for CR = 100 and 60 and was only improved slightly even for CR = 40 and 20 in quantification of subpixels and mixed pixels, particularly, their performance in mixed pixel quantification with all compression ratios. By contrast, IPCA-JPEG2000 Multicomponent and PCA-JPEG2000 with all compression ratios performed consistently better for both subpixel and mixed pixel quantification. The experimental results in this example were also very similar to those in Example 19.1.

Table 19.6 also calculates SNR and MSE for the three compression techniques, 3D-SPIHT, JPEG2000 Multicomponent and IPCA/JPEG2000 Multicomponent with CR = 100, 60, 40, and 20 where all the three produced similar SNRs and MSEs.

According to Tables 19.5 and 19.6 IPCA/JPEG2000 Multicomponent performed very well in both quantification of subpixels and mixed pixels with very close SNR/MSE. Similar conclusions drawn from Examples 19.1 and 19.2 were also observed in Ramakishna (2004), Ramakishna et al. (2005), and Ramakishna et al. (2006). All these experiments conducted above as well as those in in Ramakishna (2004), Ramakishna et al. (2005), and Ramakishna et al. (2006) demonstrated an important fact that using SNR and MSE as compression measures was inappropriate to address the issues of subpixels and mixed pixels when compression ratio was high, that is, greater than 20. This implies that SNR and MSE could not be blindly applied to hyperspectral data compression without precaution.

Table 19.6 SNR and MSE values for different compression techniques

CR	100		60		40		20	
	SNR	MSE	SNR	MSE	SNR	MSE	SNR	MSE
3D-SPIHT	48.68	1.22E + 04	48.98	1.14E + 04	49.35	1.05E + 04	50.56	7.90E + 03
JPEG2000 Multicomponent	48.45	1.28E + 04	48.81	1.18E + 04	49.13	1.10E + 04	50.17	8.64E + 03
IPCA/JPEG2000 Multicomponent	48.94	1.15E + 04	48.94	1.15E + 04	48.94	1.15E + 04	48.94	1.15E + 04

19.7.2 Real Image Experiments

In this section, similar experiments conducted for the synthetic images in Section 19.6.1 were also performed for the real image scene in Figure 1.16. The CEM discussed in Chapter 2 and UFCLS in Chapter 8 were also used to investigate the issues of subpixel detection and mixed pixel classification/quantification respectively for the same six compression techniques used in Section IV, 3D-SPIHT, JPEG2000Multicomponent, PCA/JPEG2000 Multicomponent, ICA/JPEG2000 Multicomponent, IPCA/JPEG2000 Multicomponent, and IICA/JPEG2000 Multicomponent for performance evaluation. The VD estimated for this scene was 9 according to experiments conducted in Chapter 5.

EXAMPLE 19.3

(Subpixel Panel Detection)

The CEM was implemented on the 15 panels in Figure 1.15(b), particularly the five subpixel panels, p_{13}, p_{23}, p_{33}, p_{43}, p_{53} in the third column to show the effect of lossy compression on target detection where the $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, $\mathbf{p}_5$ in Figure 1.16 were used as desired target signatures. The results of the CEM implemented on the original uncompressed image are shown in Figure 19.16.

Since similar results were obtained for panels in all the five rows, Figures 19.17–19.22 only show results obtained for detection of panels for the fifth row by applying CEM to six compressed images resulting from 3D-SPIHT, JPEG2000 Multicomponent, PCA/JPEG2000 Multicomponent, ICA/JPEG2000 Multicomponent, IPCA/JPEG2000 Multicomponent, IICA/JPEG2000 Multicomponent. This is because the detection of the panels in the fifth row is challenging due to the panels in rows 4 and 5 made by the same panel materials with two different paints. As a result, CEM detected panels in both rows 4 and 5 if either $\mathbf{p}_4$ or $\mathbf{p}_5$ was used as a desired signature. This was also true for CEM in detection of panels in the second and third rows.

Comparing all the detection results in Figures 19.17–19.22 against that in Figure 19.16, the best results were those produced by ICA/JPEG2000 Multicomponent and IICA/JPEG2000 Multicomponent where PCA-based/3D compression was among the worst and 3D compression came in between but did not perform well either.

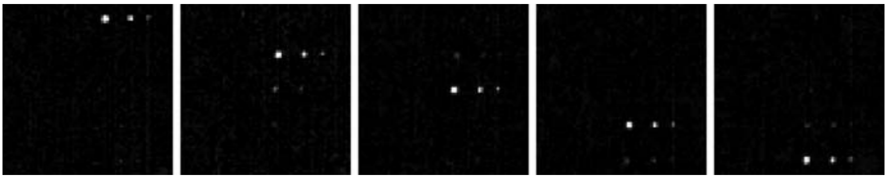


Figure 19.16 15-panel detection results by applying CEM on the original image scene in Figure 1.16(a).

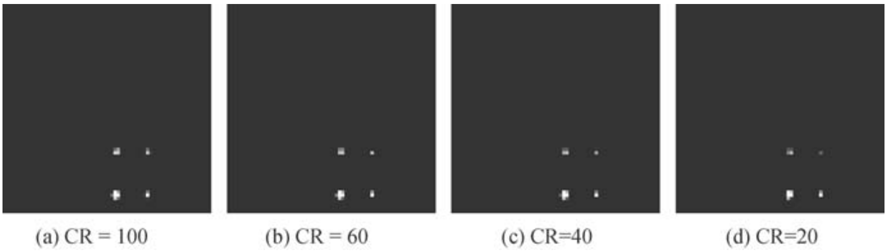


Figure 19.17 Detection of panels in the fifth row by produced by applying CEM to 3D-SPIHT compressed images.

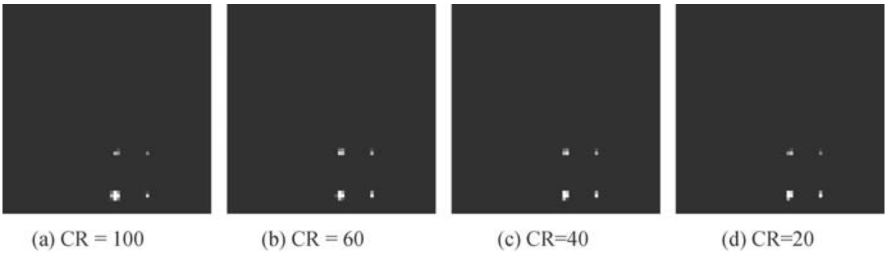


Figure 19.18 Detection of panels in the fifth row by produced by applying CEM to JPEG2000 Multi-component-compressed images.

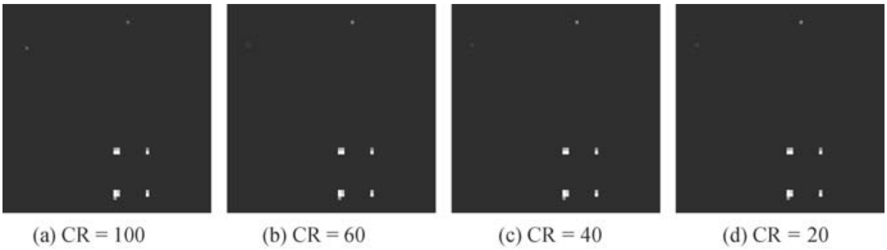


Figure 19.19 Detection of panels in the fifth row by produced by applying CEM to PCA/JPEG2000 Multi-component compressed images.

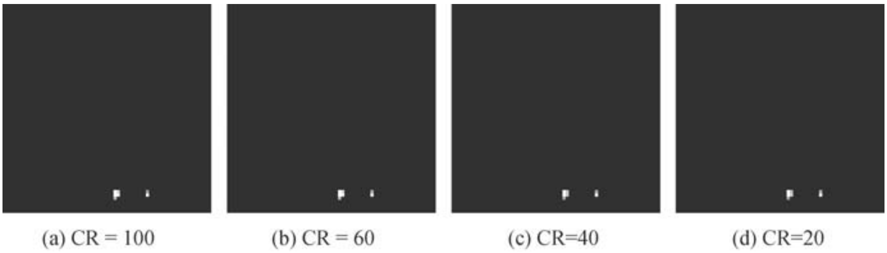


Figure 19.20 Detection of panels in fifth row by produced by applying CEM to ICA/JPEG2000 Multi-component compressed images.

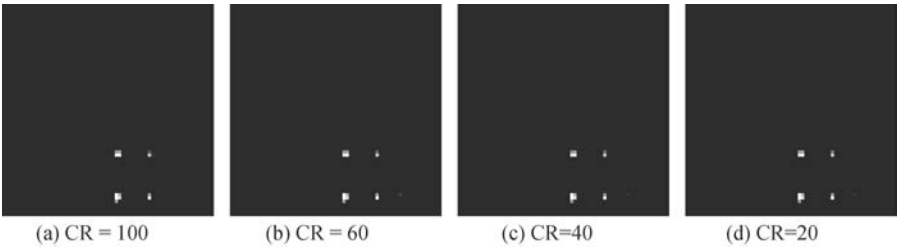


Figure 19.21 Detection of panels in the fifth row by produced by applying CEM to IPCA/JPEG2000 Multicomponent compressed images.

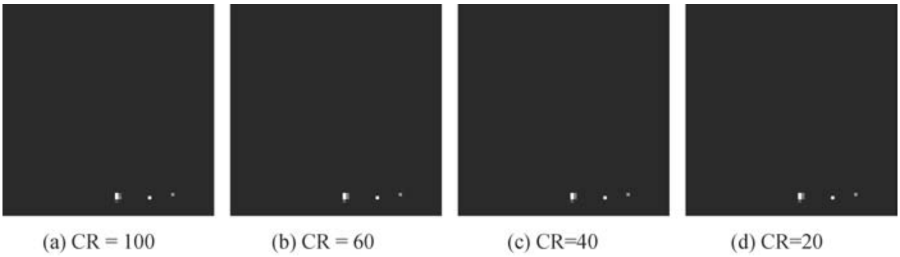


Figure 19.22 Detection of panels in the fifth row by produced by applying CEM to IICA/JPEG2000 Multicomponent compressed images.

EXAMPLE 19.4

(Mixed Pixel Panel Quantification)

The experiments conducted in Example 19.3 were designed to investigate the issue of subpixel target detection where only desired target knowledge was required. The following example was conducted to demonstrate the ineffectiveness of 3D lossy compression on mixed pixel panel classification and quantification with UFCLS used to perform unmixing for quantification of the 15 panels. In this case, the target knowledge must be known prior to UFCLS. It was demonstrated in Heinz and Chang (2001) and Chang (2003a) that 34 target pixels could be generated in an unsupervised manner and provided sufficient target information for UFCLS to perform well. Once again, due to similar results that could be obtained for all the 15 panel pixels as Example 19.3 did, Table 19.7 only tabulates UFCLS-unmixed abundance fractions (%) of three panels in the fifth row where the panel in the first column is a two-pixel panel, denoted by p_{511} and p_{512}, the panel pixel p_{52} in the second column and the subpanel pixel by p_{53} in the third column. The six lossy compression techniques 3D-SPIHT, JPEG2000 Multicomponent, PCA/JPEG2000 Multicomponent and ICA /JPEG2000 Multicomponent, IPCA/JPEG2000 Multicomponent, IICA-JPEG2000 Multicomponent were evaluated for CR = 100, 60, 40, and 20.

From this table we can see that ICA-based spectral/spatial compression techniques clearly outperformed the other compression techniques. More interestingly, the compression ratios had little effect on the unmixed abundance fractions of PCA/spatial and ICA/spatial compression techniques, while the accuracy of the unmixed abundance fractions of 3D-SPIHT and 3D-multicomponent JPEG2000 was gradually increased with compression ratios. Finally, Table 19.8 also tabulates SNR and MSE for the four compression techniques, 3D-SPIHT, JPEG2000 Multicomponent, IPCA/JPEG2000 Multicomponent, and IICA/JPEG2000 Multicomponent with CR = 100, 60, 40, and 20 where IICA yielded the worst MSE and SNR, but produced the best detection performance in Figure 19.22 among the four techniques.

Table 19.7 Abundance fractions (%) of the mixed pixel panels of the fifth row

CR	100				60				40				20			
	p_{511}	p_{512}	p_{52}	p_{53}	p_{511}	p_{512}	p_{52}	p_{53}	p_{511}	p_{512}	p_{52}	p_{53}	p_{511}	p_{512}	p_{52}	p_{53}
Original image	72	100	78	15	72	100	78	15	72	100	78	15	72	100	78	15
3D-SPIHT	58	100	35	1	55	100	56	8	60	100	62	13	69	100	74	13
JPEG 2000 Multicomponent	57	100	18	0	67	100	48	9	66	100	61	6	62	100	57	11
PCA/JPEG2000 Multicomponent	*n*	*n*	*n*	*n*	*n*	*n*	*n*	*n*	*n*	*n*	*n*	*n*	*n*	*n*	*n*	*n*
ICA/JPEG2000 Multicomponent	71	100	70	0	69	100	72	7	69	100	72	12	69	100	72	12
IPCA/JPEG2000 Multicomponent	71	100	77	10	73	100	78	15	72	100	78	18	73	100	78	16
IICA/JPEG2000 Multicomponent	77	100	74	6	67	100	74	8	68	100	75	11	68	100	75	11

The above experiments provided further evidence that MSE and SNR were indeed not appropriate criteria to be used for compression to address issues of subpixels and mixed pixels for hyperspectral image compression.

One particular comment is noteworthy. According to all the conducted experiments, the improvement on compression performance of 3D compression techniques was closely related and proportional to an increase in SNR, decrease of MSE as well as decrease of compression ratio. That is, the better performance the 3D compression; the higher the SNR, the smaller the MSE and the lower the compression ratio. This observation explains the reason why researchers in data compression community have focused their attention on criteria of SNR and MSE. Unfortunately, this common sense is no longer true for hyperspectral data compression when compression ratio is high as just demonstrated in the above four examples. This is because SNR and MSE have very little impact on subpixel and mixed pixel analyses such as subpixel detection and mixed pixel quantification in these cases. In other words, in order for a hyperspectral data compression to be effective, exploitation applications are the key to its success. Blinding using SNR and MSE as compression criteria may mislead results in hyperspectral data interpretation and analysis.

EXAMPLE 19.5

(Mixed Component Analysis for Spectral/Spatial Compression)

The HYDICE 15-panel scene was once again used for experiments to demonstrate the utility of mixed-component analysis for spectral/spatial compression systems in Figure 19.9. Five scenarios of mixed

Table 19.8 SNR and MSE for different compression techniques

CR	100		60		40		20	
	SNR	MSE	SNR	MSE	SNR	MSE	SNR	MSE
3D-SPIHT	28.38	9.01E + 05	31.76	4.13E + 05	34.93	1.99E + 05	41.10	4.80E + 04
JPEG2000 Multicomponent	26.93	1.25E + 06	30.02	6.16E + 05	32.79	3.26E + 05	38.03	9.74E + 04
IPCA/JPEG2000 Multicomponent	40.34	5.73E + 04	42.52	3.46E + 04	42.79	3.26E + 04	42.79	3.26E + 04
IICA/JPEG2000 Multicomponent	11.66	4.22E + 07	11.66	4.23E + 07	11.66	4.23E + 07	11.66	4.23E + 07

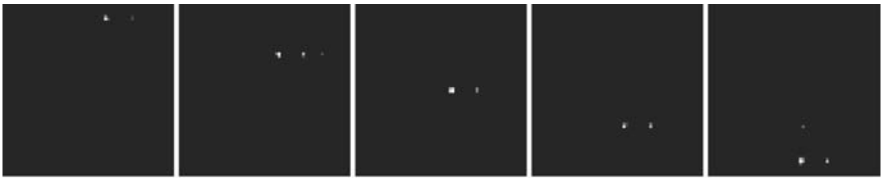


Figure 19.23 15-panel classification by UFCLS on the original image cube.

PCA/ICA transform with ($m = 9$, $n = 0$) (i.e., PCA), ($m = 0$, $n = 9$) (i.e., ICA), ($m = 1$, $n = 8$), and ($m = 2$, $n = 7$) were investigated for applications in mixed pixel classification/quantification with UFCLS used to evaluate their performance on compressed and decompressed domains. These experiments were considered in Chai et al. (2007) to investigate mixed PCA/ICA component analysis for hyperspectral imagery. In this case, the estimated value of VD was $q = 9$ and the number of classes to be classified was also set to $p = q = 9$. Like Experiment 19.6.4, 34 target pixels were used to provide prior knowledge for mixed pixel classification/quantification. Therefore, UFCLS was applied to the original HYDICE image with p set to 34 to perform unmixing. Figure 19.23 shows only five images that are used to unmix the 15 panels. These mixed pixel classification results were used as a benchmark for studying the following spectral/spatial compression experiments conducted by mixed component analysis.

It should be noted that since the panels in rows 2 and 3 are made the same material with different paints, the detection of panels in row 2 might also detect panels in row 3 and vice versa. Similarly, it is also true for detection for the panels in rows 4 and 5.

EXPERIMENT 19.1

(Scenario 1: PCA, ($m = 9, n = 0$))

In this experiment, mixed PCA/ICA transform was reduced to the standard PCA transform. Figure 19.24 shows the nine PCA-generated PCs. Figures 19.25 and 19.26 show the results unmixed by UFCLS for the 15 panels using nine PCs compressed and decompressed image cubes, respectively.

As demonstrated in Figures 19.25 and 19.26, UFCLS performed very poorly in unmixing the 15 panels in the scene. Comparing the results in Figures 19.25 and 19.26 against that in Figure 19.23 PCA-based spectral/-spatial compression failed to capture subtle details of the 15 panels. This was primarily due to the fact that PCA-based spectral compression was a second-order statistics-based transform that largely characterizes background information in Figures 19.25 and 19.26, but not panel information that is generally preserved by

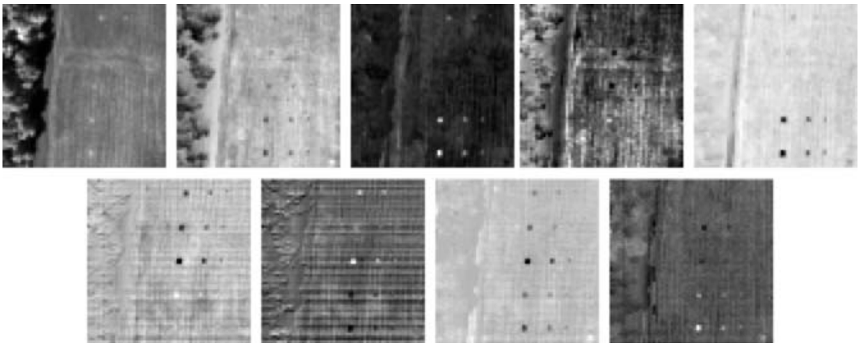


Figure 19.24 Nine principal components.

Figure 19.25 Classification by UFCLS using nine PCs in Figure 19.24.

high-order statistics as demonstrated in the following example. As also demonstrated in Figures 19.25 and 19.26, the results produced by using the nine PC-decompressed image cubes only show little improvement on panel detection and classification. This implied that the decompressed image cube reconstructed from the nine PCs seemed provided no significant advantage in target detection and mixed pixel classification.

EXPERIMENT 19.2

(Scenario 2: ICA, ($m = 0, n = 9$))

As a complete opposite of Scenario 1 in Experiment 19.1 of Example 19.5, this experiment considered another scenario, called Scenario 2 where all nine components used for mixed pixel classification were nine ICs shown in Figure 19.27 obtained from the FastICA using the ATGP-generated initial projection vector.

Evidently, the nine ICs in Figure 19.27 already extracted all the 15 panels, while the background information preserved in Figure 19.24 seemed not shown in these nine ICs. Figures 19.28 and 19.29 show the results produced by UFCLS in unmixing of the 15 panels using nine ICs compressed and decompressed image cubes respectively where UFCLS performed using ICA better than using PCA in Figures 19.25 and 19.26 in terms of panel target detection.

Figure 19.26 15-panel classification by UFCLS on PCA-decompressed image cube obtained by using nine PCs in Figure 19.24.

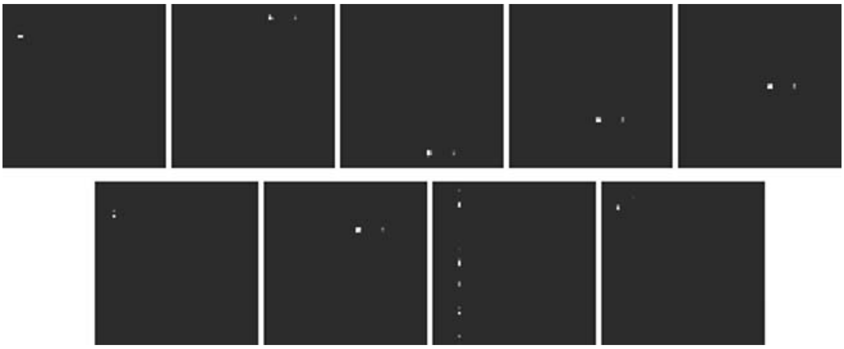


Figure 19.27 Nine independent components.

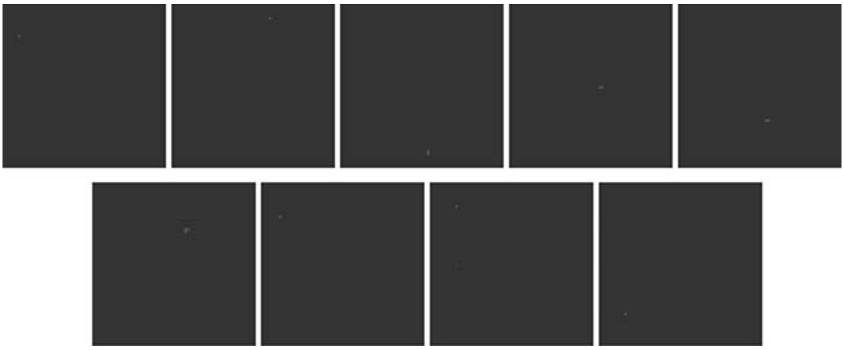


Figure 19.28 Classification by UFCLS using nine ICs in Figure 19.27.

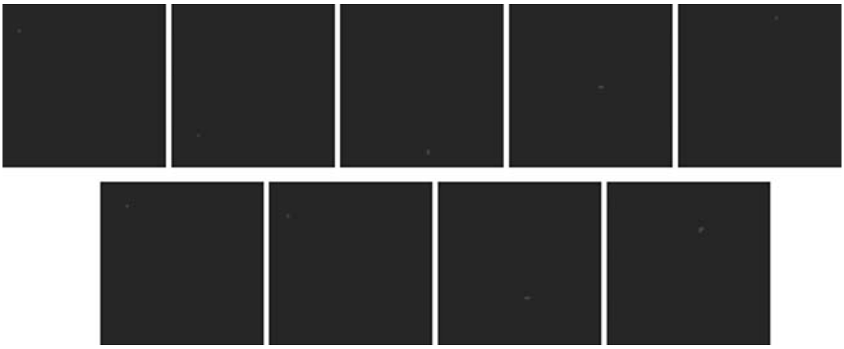


Figure 19.29 Classification by UFCLS using ICA-decompressed image cube obtained by using nine ICs in Figure 19.27.

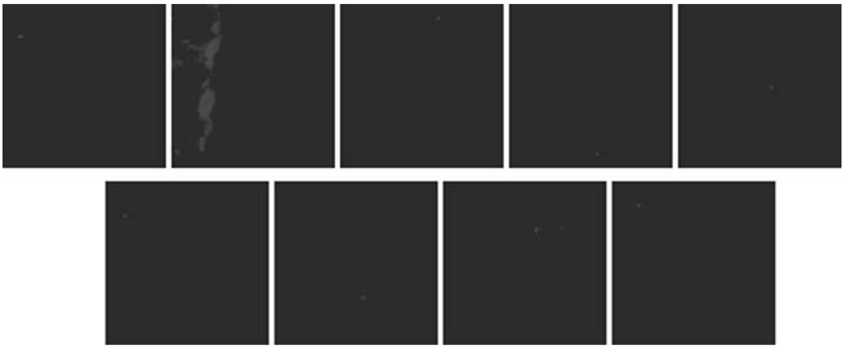


Figure 19.30 Classification by UFCLS using nine mixed components.

EXPERIMENT 19.3

(Scenario 3: ($m = 1, n = 8$)-PCA/ICA transform)

Experiments 19.1 and 19.2 of Example 19.5 demonstrated advantages and disadvantages of PCA or ICA transform used for compression. This experiment investigated a scenario called Scenario 3 to see how much gain can be obtained by the proposed mixed (m,n) PCA/ICA transform.

In order to see the performance of mixed (1,8)-PCA/ICA transform in mixed pixel classification, Figures 19.30 and 19.31 show the 15-panel unmixed results by UFCLS using the (1,8) mixed component-compressed and decompressed image cubes, respectively.

The results obtained in Figure 19.30 by using the (1,8) mixed component-compressed image cube seemed slightly better than those obtained in Figure 19.31 by using the (1,8) mixed component-decompressed image cube in terms of classification of 15 panels, particularly, panels in rows 2 and 3. But results shown in Figure 19.31 were closer to UFCLS results obtained based on the original image cube shown in Figure 19.23. In addition, they also extracted more background classes than that in Figure 19.30.

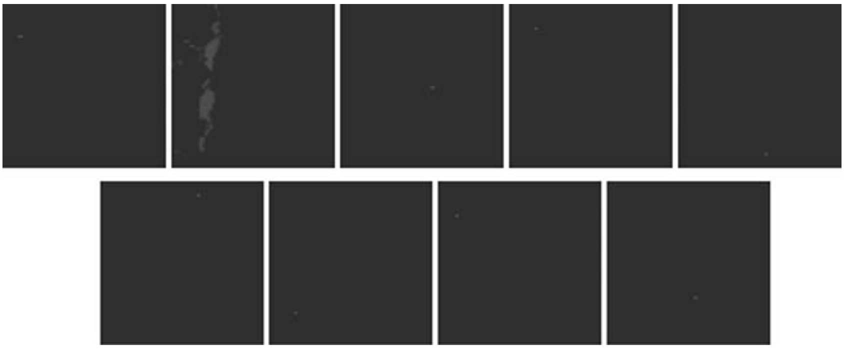


Figure 19.31 Classification by UFCLS on the (1,8)-PCA/ICA-decompressed image cube obtained by using nine mixed components.

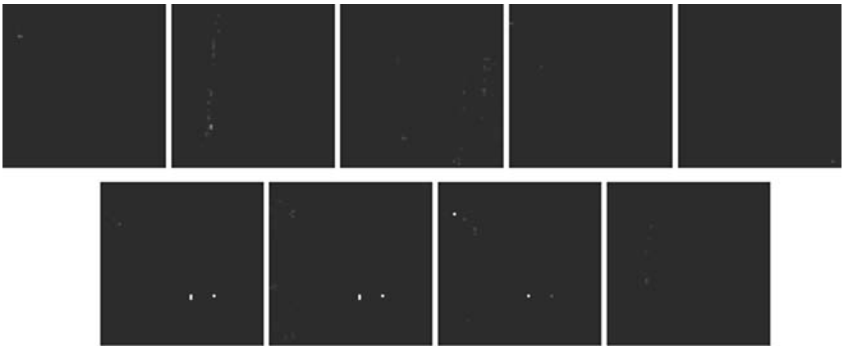


Figure 19.32 Classification by UFCLS using nine mixed components.

EXPERIMENT 19.4

(Scenario 4: ($m = 2, n = 7$)-PCA/ICA transform)

This experiment further investigated another scenario, called Scenario 4 to see if a further improvement can be gained by increasing one PC to 2 PCs while reducing eight ICs to seven ICs operated by a mixed (2,7) PCA/ICA transform. Figures 19.32 and 19.33 show the 15-panel unmixed results by UFCLS using the (2,7) mixed component-compressed and (2,7) mixed component-decompressed image cubes, respectively, where UFCLS performed poorly in unmixing the 15 panels in both cases.

This experiment further demonstrated that adding one more PC could only do more harm than good for UFCLS-mixed pixel classification. This is because the first PC was sufficiently enough to capture second-order statistics and adding a second PC did not provide useful information in terms of panel detection and classification, but rather obscured the panel information.

Finally, the LSEs were calculated for four scenarios conducted in the above four experiments using mixed (m,n) PCA/ICA transforms with (m,n) = (9,0), (0,9), (1,8), and (2,7). Their LSE results are tabulated in Table 19.9.

As clearly shown in Table 19.8, LSE was increased as more PCs were replaced by ICs where the smallest and largest LSEs were produced by (9,0)-PCA and (0.9)-ICA, respectively. However, the worst performance in mixed pixel classification came from the (9,0)-PCA and (2,7)-PCA/ICA, which yielded the smaller LSEs compared to (0,9)-ICA and (1,8)-PCA/ICA that performed significantly better in the 15-panel classification but produced larger LSEs. This evidence demonstrated that a smaller LSE did not necessarily produce better

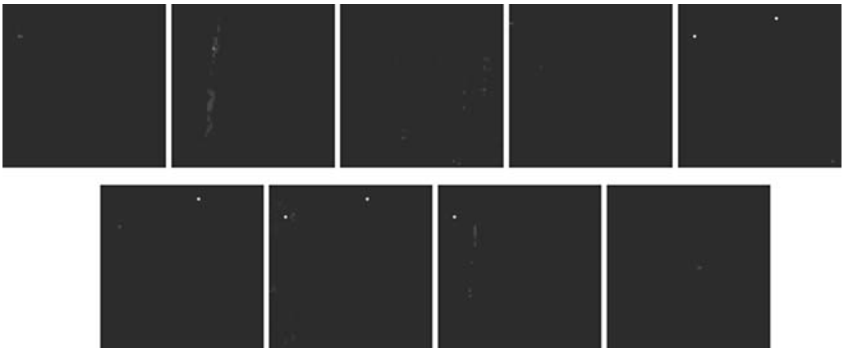


Figure 19.33 Classification by UFCLS on PCA/ICA-decompressed image cube obtained by using nine mixed components.

Table 19.9 LSEs calculated for Experiments 19.1–19.4

	(9,0)	(0,9)	(1,8)	(2,7)
LSE	3.25×10^4	4.24×10^7	8.07×10^6	3.83×10^5

performance in mixed pixel classification. This further suggested that the LSE might not be an appropriate measure for hyperspectral data compression.

EXPERIMENT 19.5

(Mixed Pixel Quantification)

The above four experiments were evaluated by visual assessment qualitatively. It is very difficult to see how a mixed (*m*,*n*)-PCA/ICA transform performed quantitatively. In doing so, we conducted this experiment to evaluate the performance of a mixed (*m*,*n*)-PCA/ICA transform in mixed pixel quantification for the 15 panels in Figure 2. According to the results obtained by the four scenarios only Scenarios 2 and 3 successfully unmixed 15 panels in terms of classification, so only the results produced by (0,9)-PCA/ICA transform and (1,8)-PCA/ICA transform were used for mixed pixel quantification analysis. Table 19.10 tabulates their abundance quantification of the 15 panels obtained in Figures 19.28 and 19.29 and Figures 19.30 and 19.31, respectively, where the results for abundance quantification of the 15 panels obtained by applying UFCLS to the original image cube in Figure 19.23 are also included for benchmark comparison.

As noted in Table 19.10, the subpixel panels, p_{13}, p_{23}, p_{33}, p_{43}, and p_{53} in Figure 1.16(b) have the size of 1 m × 1 m smaller than the 1.56 m × 1.56 m-spatial resolution. It indicates that the size of these five subpixel panels is approximately $1/(1.56\,\mathrm{m})^2 = 0.41,091\ \mathrm{m}^2$ that is equivalent to saying that the abundance fractions

Table 19.10 Abundance quantification by UFCLS using mixed (*m*, *n*)-PCA/ICA spectral/spatial compression

	($m=1$, $n=8$)-PCA/ICA		($m=0$, $n=9$)-PCA/ICA		UFCLS ($p=34$)
	Compressed	Decompressed	Compressed	Decompressed	
p_{11}	1	1	1	1	1
p_{12}	0.4748	0.4419	0.4708	0.4283	0.4098
p_{13}	0.2175	0.1542	0.2166	0.1721	0.0499
p_{211}	0.9748	0.3438	1	1	0.5255
p_{221}	1	0.3424	0.9853	0.9191	0.3141
p_{22}	0.9031	0.2890	0.9181	0.9051	0.6917
p_{23}	0.3652	0.1112	0.3751	0.248	0.4221
p_{311}	0.8372	0.9326	0.844	0.7569	0.8647
p_{312}	1	1	1	1	1
p_{32}	0.5615	0.5439	0.5647	0.4946	0.5343
p_{33}	0.3859	0.3406	0.3929	0.3122	0.3285
p_{41}	1	1	1	0.466	1
p_{412}	0.8928	0.9019	0.8991	0.5501	0.3821
p_{42}	0.8052	0.7936	0.7974	0.8361	0.7034
p_{43}	0.2622	0.1986	0.2489	0.2596	0.2242
p_{511}	0.7265	0.6956	0.7252	0.5846	0.7203
p_{521}	1	1	1	1	1
p_{52}	0.7386	0.7294	0.7349	0.724	0.7789
p_{53}	0.1982	0.1368	0.1812	0.1213	0.1466

of these three subpixels present in a single pixel are close to 0.41,091. With this interpretation, Table 19.10 shows that both mixed (1,8)-PCA/ICA transform performed slightly better than mixed (0,9)-PCA/ICA transform in the 15-panel abundance quantification with UFCLS results obtained on the reconstruction image based on (1,8)-PCA/ICA to have the closest results to UFCLS results produced using the original image cube.

One remark is noteworthy. VD = 9 used in the above experiments was only an estimate and not necessarily to be exact. According to our extensive experiments on the HYDICE 15-panel scene in Figure 1.16 for dimensionality reduction, VD = 9 was appropriate. However, since the value estimated by VD was fixed, VD = 9 used for dimensionality reduction may not be appropriate for other applications such as band selection (Chang and Wang, 2006). This is because different applications require different false alarm probabilities. So, we can always make the estimate of VD variable by varying false alarm probability at different thresholding levels. In order to make VD versatile, varying the false alarm probability can adapt various applications. Keeping this in mind, VD = 9 may be effective for dimensionality reduction, but it may not imply that it is appropriate for data compression because using ICA alone may not preserve background information and using PCA alone may miss information of small objects. Of course, if the value of *p* is sufficiently large, PCA and ICA can perform well. However, for a smaller value of *p*, neither could work well as demonstrated by our experiments. In this case, our proposed mixed PCA/ICA transform could still work well for the same value of *p*. This provides the evidence that mixed PCA/ICA transform has advantages over PCA and ICA by retaining strengths of PCA and ICA.

EXAMPLE 19.6

(Mixed Pixel Panel Quantification)

Since all similar experiments conducted for HYDICE data can be also applied to this AVIRIS Cuprite data in Figure 1.12, only mixed pixel quantification was studied for quantitative analysis for illustration to demonstrate that the proposed mixed PCA/ICA was a general approach that also worked for any hyperspectral data. In order to simplify our experiment, only a quarter size of the image at right bottom corner in Figure 19.34(a) was selected for study and is shown in Figure 19.34(a) and (b) with four 3 × 3 simulated panels implanted at the left bottom corner in Figure 19.34(b) in such a way that the implanted panel pixels replaced their corresponding real image pixels. In other words, the image scenes in Figures 1.12(b) and 19.34(b) are identical except that there are 36 simulated (nine pixels for each panel) pixels implanted in Figure 19.34(b) to replace the original 36 real image pixels in Figure 1.12(b). These four panels, denoted by p_{ij} for $i = 1,2$ and $j = 1,2$, were simulated in accordance with composition of signatures specified in Table 19.11 where row and column indices are indicated by *i* and *j*, and the BKG was specified by the signatures of the real image pixels in the

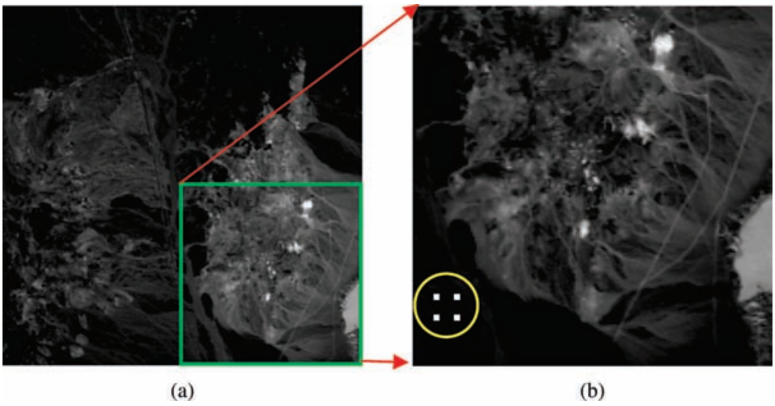


Figure 19.34 Spatial locations of four implanted panels.

Table 19.11 Compositions of four simulated panels

Pixel	Signatures	Composition
Nine pixels in p_{11}	Calcite	100% Calcite + 0% BKG
Nine pixels in p_{12}	Calcite, BKG	50% Calcite + 50% BKG
Nine pixels in p_{21}	Kaolinite	100% Kaolinite + 0% BKG
Nine pixels in p_{22}	Kaolinite, BKG	50% Kaolinite + 50% BKG

scene that were replaced by their corresponding implanted pixels. For example, all nine pixels in panel p_{12} were simulated by 50% Calcite and 50% background (BKG) signature specified by their replaced real image pixels. So, all of these nine simulated panel pixels contained 50% Calcite, but 50% different BKG signatures.

Two benefits can be gained from our designed experiments. One is that the data are available on website so that whoever is interested in our proposed method can repeat what we did in the experiments to compare their new algorithms. The other benefit is that all parameters used to simulate the panels were fully controlled so that performance analysis was conducted impartially and objectively. VD estimated for the image scene in Figure 15(b) was 10 with the false alarm probability set by $P_F = 10^{-3}$. By virtue of VD = 10 and the 36 panel pixels simulated in Table 19.11 a detailed analysis for their quantifications can be conducted by UFCLS using PCA only, ICA only, and Mixed PCA/ICA transforms. Interestingly, the quantitative unmixed results obtained by UFCLS for panels in row 1 and row 2 were quite different and were analyzed as follows.

1. For panel pixels in the first panel p_{11} in row 1 simulated by the 100% mineral signature Calcite, UFCLS only using PCA (i.e., (10,0)-PCA/ICA.) could not detect any of nine pure panel pixels in the panel p_{11}, while UFCLS was able to detect and quantify all the nine panels pixels with correct 100% abundance of Calcite if the only ICA (i.e., (0,10)-PCA/ICA.) and Mixed PCA/-ICA (1,9) transform were used.
2. For panel pixels in the first panel p_{21} in row 2 simulated by the 100% mineral signature kaolinite, UFCLS using all the three different transforms successfully detected and quantified all the nine panel pixels with correct 100% abundance of kaolinite. According to the ground truth provided by the USGS, the major background for this scene is made up by the mixture of alunite and kaolinite. As a result, PCA was able to pull out some information of kaolinite from the panel p_{21}. This is different from the results obtained above from the panel p_{11} where all the nine pixels are made up by 100% calcite that is not part of the image background. In this case, the calcite could not be extracted by the second-order statistics transforms; instead it could be only extracted by the high-order statistics-based transforms.
3. For the panel pixels in the second panel p_{12} in row 1, Table 19.12 tabulates UFCLS-unmixed abundance fractions of the Calcite contained in the nine panel pixels in the second panel p_{12} in row 1 where there were no results for PCA since it missed all the nine panel pixels. Figure 19.35 shows the graphical representations plotted by abundances in Table 19.12 for visual assessment where it clearly shows that the (1,9)-PCA/ICA performed significantly better than the (0,10)-PCA/ICA transform.
4. For the panel pixels in the second panel p_{22} in row 2, Table 19.13 tabulates UFCLS-unmixed abundance fractions of the kaolinite contained in the nine panel pixels in the second panel p_{22}. Figure 19.36 shows the graphical representations plotted by abundances in Table 19.5 for visual assessment.
5. According to Figure 19.36, Mixed PCA/ICA(1,9) transform was best among all the three transforms and the (0,10)-PCA/ICA transform came to the next with the (10,0)-PCA/ICA transform was the worst.

Table 19.12 UFCLS estimated abundance fractions of calcite contained in nine panel pixels in p_{12} in row 1

50% Calcite	PCA only ($m = 10$, $n = 0$)	ICA only ($m = 0$, $n = 10$) (%)	Mixed PCA/ICA ($m = 1$, $n = 9$) (%)
Pixel 1	N/A	58.87	51.29
Pixel 2		57.48	49.92
Pixel 3		58.70	50.94
Pixel 4		55.65	47.52
Pixel 5		59.30	51.26
Pixel 6		58.65	50.81
Pixel 7		56.87	48.93
Pixel 8		61.08	53.34
Pixel 9		54.93	46.85

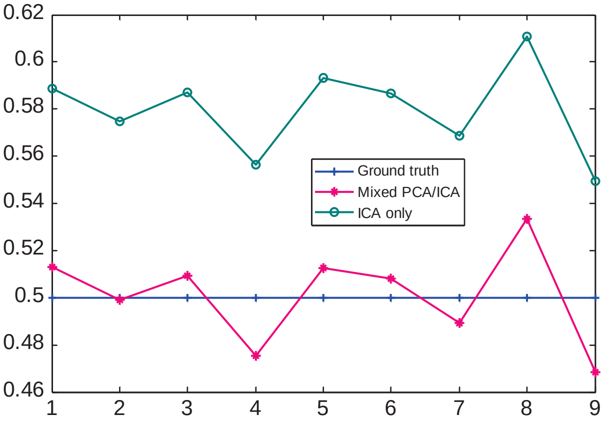


Figure 19.35 Graphical representation of Table 19.12 for visual assessment.

Table 19.13 UFCLS-estimated abundance fractions of kaolinite contained in the nine panel pixels in p_{22} in row 2

50% Kaolinite	PCA only ($m = 10$, $n = 0$) (%)	ICA only ($m = 0$, $n = 10$) (%)	Mixed PCA/ICA ($m = 1$, $n = 9$) (%)
Pixel 1	53.94	52.54	50.29
Pixel 2	50.79	53.57	51.11
Pixel 3	55.75	50.89	48.61
Pixel 4	51.80	51.50	49.00
Pixel 5	54.00	52.81	50.55
Pixel 6	52.06	51.18	48.98
Pixel 7	52.04	52.02	49.92
Pixel 8	52.86	53.53	50.96
Pixel 9	51.56	51.55	49.38

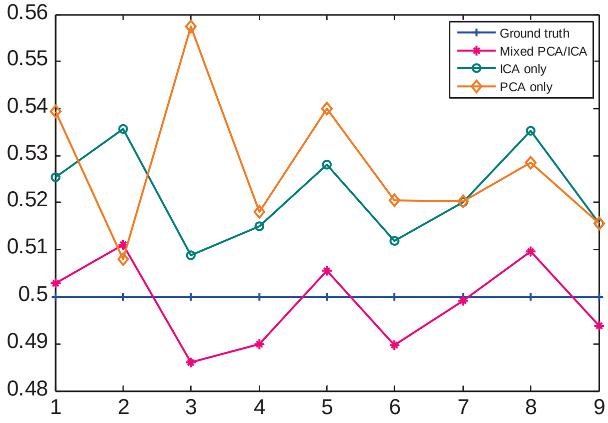


Figure 19.36 Graphical representation of Table 19.13 for visual assessment.

19.8 Conclusions

Hyperspectral data compression has been considered as a crucial step in preprocessing of hyperspectral data. Instead of focusing on design and development of 3D compression algorithms as most of current efforts are devoted to hyperspectral data compression, this chapter takes a rather different approach by addressing and investigating two important and crucial issues arising in hyperspectral data compression, subpixels and mixed pixels analysis. In particular, it shows and demonstrates via experiments several important overlooked issues that have been proven crucial in hyperspectral data compression and need to be addressed. For a hyperspectral data compression to be effective, hyperspectral data compression must be conducted on an exploitation basis and a blind use of data compression technique generally results in inappropriate interpretation. A direct application of 3D lossy compression techniques to hyperspectral imagery may cause significant loss of crucial information provided by subpixels and/or mixed pixels. Secondly, SNR and MSE have been shown inappropriate to be used as compression criteria for subpixel and mixed pixel analysis when the compression ratio is high. In other words, higher SNR or lower MSE does not guarantee better compression performance in terms of information extraction and vice versa when compression rate at low bits. Thirdly, to address the issues of subpixels and mixed pixels, spectral/spatial compression techniques are shown to be always better and more effective than 3D compression techniques. Fourthly, in order to spectral/spatial compression techniques, a newly developed concept of VD is proposed for hyperspectral image compression to estimate number of principal components needed to be retained for dimensionality reduction. Over the past years, this number has been assumed *a priori* on a trial and error basis. Using spectral/spatial compression in conjunction with VD for hyperspectral data compression is a new approach. Lastly, despite that we did not include spectral/2D compression techniques in this chapter, the results in Ramakishna (2004), Ramakishna et al. (2005), and Ramakishna et al. (2006) have shown that the spectral/3D compression always performed better than the spectral/2D spatial compression techniques where the latter have been extensively used in spectral/spatial compression in the literature, while the former has not received attention in the past. The reason that we believe is that many researchers may have thought that since spectral compression has done its task to decorrelate spectral information among spectral bands, there is no need of using 3D compression and instead, 2D spatial compression may be sufficient. Unfortunately, this generally is not true as demonstrated by experiments in this chapter. As noted, since the goal of this chapter is not to develop new compression algorithms, two well-known 3D compression techniques, 3D SPIHT and JPEG2000 Multicomponent, are used for benchmark compression.

20

Progressive Spectral Dimensionality Process

Hyperspectral compression is considered the first step to preserve crucial and vital spectral information in the two-stage spectral/spatial compression proposed in Chapter 19, where dimensionality reduction by transform (DRT) and dimensionality reduction by band selection (DRBS) discussed in Chapter 6 play a vital role in dealing with the so-called curse of dimensionality in spectral compression. One key issue in implementing dimensionality reduction (DR) and DRBS for spectral compression is that the number of dimensions q to be retained after DRT and the number of bands $\tilde{q}$ to be retained after DRBS must be known *a priori*. Despite the fact that these two values, q and $\tilde{q}$, can be estimated by virtual dimensionality (VD) developed in Chapter 5, VD is not a one-size-fit-all universal criterion for various applications. In order to mitigate the dependence on VD, this chapter develops a new DRT, to be called progressive spectral dimensionality process (PSDP), which introduces a new concept of dimensionality prioritization (DP) that revolutionizes how the commonly used DR is implemented. The transform used to perform DR (i.e., DRT) is a linear transformation that converts the original data dimensions into spectral transformed components, each of which represents a new dimension, referred to as spectral dimension. The idea of PSDP is to first specify a particular DRT to transform the original data into a new spectral data space, where the original data dimensions are represented by spectral dimensions. It is then to select a DP criterion to calculate the information contained in each spectral dimension as a priority score to rank the significance of this particular spectral dimension. By means of these DP-ranked priority scores, two dual processes can be derived to perform PSDP. One is referred to as progressive spectral dimensionality reduction (PSDR) via DP, which removes spectral dimensions progressively according to their corresponding priority scores in ascending order. In contrast, a complete reverse process, referred to as progressive spectral dimensionality expansion (PSDE) via DP, adds new spectral dimensions progressively according to their corresponding priority scores in descending order to expand spectral dimensions. These two dual processes, PSDR and PSDE, carry out PSDP in a complete reverse manner in terms of decreasing and increasing data information, respectively. By taking advantage of PSDE and PSDR, hyperspectral data can be processed progressively in data communication, transmission, and compression.

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

20.1 Introduction

Dimensionality reduction (DR) is probably the most common and popular technique that has widely been used in multivariate data analysis to resolve so-called curse of dimensionality (Fukunaga, 1990; Bischop, 1995). Chapter 6 provides a comprehensive description of DRT techniques. Since hyperspectral imaging sensors utilize hundreds of contiguous spectral bands for data acquisition, processing such enormous data volumes becomes a formidable and challenging task for image analysts. To mitigate this dilemma, DR is a feasible solution to reduce the original data space to a relatively low data space to meet practical needs, such as removal of data redundancy, reduction of the expensive computational cost. However, a major issue in DR is determination of the number of dimensions, q, to be retained by DR, where the value of q can be estimated by VD, as shown in various applications by Chang (2006a, 2006b). Nevertheless, it has also been shown in Chapter 5 that VD can adapt and vary with different applications.

Due to the fact that hundreds of spectral dimensions are needed to be dealt with, selecting an appropriate q from such a wide range of values, that is, $1 \leq q \leq L$, where L is the total number of spectral bands, can be very tricky. This chapter develops and coins a new concept, to be called dimensionality prioritization (DP), to resolve this issue. The motivation of DP arises from a need to process vast amount of hyperspectral data in a more effective manner in many applications, for example, satellite data processing, communication, where computational complexity, data archiving, storage, and transmission are of major concern. However, this is easier said than done because there are several issues that need to be addressed prior to DR. The first and foremost is to develop a credible DR transform that can compress the original data into a spectral-transformed data space in some sense of optimality. A second issue is to represent the original data in a spectral dimensionality-reduced lower data space via a DRT. Finally, a third issue is to prioritize each spectral dimension in the new reduced spectral data space, so that all spectral dimensions can be ranked in accordance with their contained information. The DP developed in this chapter attempts to resolve these issues.

In order to materialize the utility of DP in DRT, the first task is to assign priority scores to spectral dimensions after DRT according to the information contained in each spectral dimension. In other words, each spectral dimension is represented by a particular transformed component via DRT and its information is provided by projections of the entire data sample vectors onto this specific transformed component. DP first specifies a criterion to measure the content of the information provided by each of transformed components and further calculates this piece of information as a priority score to be used to rank its corresponding spectral dimension. Finally, DP prioritizes spectral dimensions in accordance with their assigned priority scores for the purpose of progressive process. As an example, the information contained in a principal component (PC) after the principal components analysis (PCA) transform is ranked by a particular eigenvalue obtained from the data sample covariance matrix. In this case, the transformed component is specified by an eigenvector associated with its corresponding eigenvalue that indicates the significance of information contained in this particular transformed component. Accordingly, PCA-transformed components are specified by eigenvectors along with priority scores ranked by their associated eigenvalues, where each eigenvector represents a spectral dimension. However, such nice properties are not necessarily applicable to any DR-transformed component. A good example is independent component analysis (ICA), which does not have a similar counterpart to pairs of eigenvectors and eigenvalues used by PCA to specify PCs in terms of ranking its generated independent components (ICs) and retaining the information of ICs by priority scores. This is mainly due to the fact that ICA does not have an analytic equation similar to the characteristic polynomial equation used by PCA that can produce eigenvalues, which can be further used to generate eigenvectors and prioritize PCs by their eigenvalues. This implies that taking advantage of the eigenvalue/eigenvector approach used by PCA does not work for ICA. Consequently, we need to find another way for ICA to be able to rank its generated ICs.

This chapter develops an approach that completely reverses the process of what has been done by PCA, namely finding eigenvalues first by solving the characteristic polynomial equation and then their corresponding eigenvectors via eigenvalues. More specifically, our proposed approach is to first find ICs and then rank the information contained in each IC. To materialize such an approach, two issues must be addressed: how to generate ICs and how to rank the information provided by each IC. Fortunately, a recent work by Wang and Chang (2006a) offered a solution that suggested three different ways to rank ICs generated by FastICA developed by Hyvarinen and Oja (1997) for IC prioritization. Such IC prioritization was further extended by Ren et al. (2006) to prioritize components generated by any arbitrary high-order statistics (HOS) component analysis. The approach presented in this chapter integrates both works in a general setting so that PCA, ICA, and HOS component analysis can be considered as special cases under a more general umbrella. The idea is to develop a projection pursuit (PP)-based DRT in which the PI-transformed components are further specified by projection index components (PICs). The resulting PI-based PP is referred to as PIPP. By means of PIPP, a transformed component can be specified by a PIC along with its priority score that can be calculated by another PI used for DP to prioritize PICs in accordance with their assigned priority scores. As noted above, the PI used to generate PICs and the PI used by DP are generally different but can be the same in many application as well.

In order to carry out DP in a progressive manner, each PIC is specified and represented by a projection vector pointing to a particular interesting direction and also prioritized according to its corresponding priority score calculated by a custom-designed PI based on the significance of the information contained in the PIC. Finally, DP is performed progressively in either a dimensionality expansion manner by adding more PICs to increase data information or a dimensionality reduction manner by removing existing PICs to decrease data information. In other words, DP allows users to perform not only dimensionality reduction but also dimensionality expansion progressively by PIPP, referred to as progressive spectral dimensionality process by PIPP (PSDP-PIPP) via DP, where two dual processes can be derived, progressive spectral dimensionality reduction by PIPP (PSDR-PIPP) via DP, and progressive spectral dimensionality expansion by PIPP (PSDE-PIPP) via DP, respectively. More specifically, PSDR-PIPP begins with a maximal number of PICs, denoted by $\max_{\mathrm{PIC}}$, and a step size, denoted by n_Δ, for example, $\max_{\mathrm{PIC}} = L$, which is the total number of spectral bands, $n_\Delta = 1$, and then gradually removes n_Δ, PICs progressively until it reaches a stopping rule that is also determined by a particular application. To the contrary, PSDE-PIPP starts with a minimal number of PICs, $\min_{\mathrm{PIC}}$ and a step size, n_Δ, for example, $\min_{\mathrm{PIC}} = 1$, $n_\Delta = 1$ and then adds n_Δ PICs progressively until it reaches a stopping rule that is also determined by a particular application. So, basically, PSDR-PIPP starts with a higher dimensional data space and gradually reduces data dimensionality to a low-dimensional space in a progressive manner, while PSDE-PIPP begins with a very low-dimensional data space and then gradually expands dimensionality to a higher dimensional data space in a progressive manner.

By virtue of PSDP-PIPP, the issue of determining the number of PICs can be resolved in the sense that there is no need to know exactly how many PICs needed to be retained but rather determined by various applications. This is what PSDP-PIPP can offer, but cannot be accomplished by the traditional DR. Interestingly, the pair of PSDR-PIPP and PSDE-PIPP implemented by PSDP is very similar to the pair of "expand" and "reduce" operations developed by Burt and Adelson (1983) that are used to construct Laplacian and Gaussian pyramids for progressive image coding. In essence, what the pair of PSDR-PIPP and PSDE-PIPP can achieve in dimensionality expansion and reduction progressively is exactly what the pair of "expand" and "reduce" operations can accomplish in image coding via a progressive construction of pyramids.

20.2 Dimensionality Prioritization

DRT takes advantage of transformed components produced by a custom-designed transformation to represent the original data in a new transformed data space with a different data representation in which each data dimension is specified by a particular transformed component. The DR is then performed by retaining a prescribed number of transformed components, q. Accordingly, the effectiveness of DRT is determined by three key factors: the DR transformation being used to produce transformed components, significance of transformed components measured by a selected information criterion, and the value of q.

Using the commonly used PCA as an example, we can illustrate these three issues as follows. First of all, PCA transforms the original 2D spatial/1D spectral data coordinate system into a new data representation system formed by a set of PCs, each of which is characterized by a specific eigenvector that corresponds to a particular eigenvalue, that is, a sample data variance. Then, the significance of each PC is further measured by the magnitude of the eigenvalue corresponding to the eigenvector that specifies this particular PC. In other words, each dimension in a PCA-transformed data space is no longer a wavelength-specified spectral dimension in the original data space. That is, the original data represented by the wavelength-based spectral dimensionality can be reduced via PCA to a small number of PCs specified by eigenvectors corresponding to large eigenvalues in a PCA-transformed data space. Finally, the third issue of "how many PCs are required for such a PC-based data representation, that is, what is the value of q?" can be determined by the virtual dimensionality (VD) developed in Chapter 5.

While PCA enjoys all the nice and desired properties such as eigenvalues and eigenvectors described above, unfortunately other DR transformations do not have such luxury. For example, the independent component analysis (ICA) (Hyvarinen et al., 2001) does not have all the properties of PCA described above. Most importantly, ICA does not have an analytic equation similar to the characteristic polynomial equation obtained for PCA from the sample covariance matrix, which that can be used to find eigenvalues. To resolve this issue, it must find projection vectors directly from the data that serve as the same purpose of eigenvectors for PCA to produce PCs to produce ICs. In doing so, a general approach is to design an algorithm to generate the desired projection vectors. To initialize such an algorithm, a common practice to randomly generate a vector as an initial projection vector that will eventually lead to desired projection vectors. However, as a consequence of using a random vector as an initial projection vector, its final converged projection vector may not be repeatable. In other words, a final converged projection vector produced by one random initial vector may be different from that produced by another random initial vector. Such randomness issue has been addressed in endmember extraction (see Chapters 9 and 10) and is also encountered in the ISODATA (C-means) clustering method in Duda and Hart (1973). Secondly, once ICs are generated, the issue of how to measure the significance of information contained in each of ICs must be addressed, since there is no counterpart of eigenvalues used by PCA in ICA that can be used to rank the generated ICs in terms of information significance. So, it leads to two challenging problems for ICA to resolve. One is to find an appropriate approach that can produce desired projection vectors. The other is to rank the orders of ICs in accordance with significance of their provided information. The PIPP and DP are developed in this chapter exactly for this purpose where PIPP generalizes the concepts of PCA and ICA by introducing PI as a criterion to identify a direction of interestingness in which case PIPP is reduced to PCA and ICA when PI is specified by data variance and mutual information, respectively, and DP then uses an information criterion that can be the same PI or another PI to measure the significance of the PI-generated components for their priority ranking. By combining PIPP with DP, we can further derive a progressive spectral dimensionality process (PSDP) that implements two dual processes, PSDR-PIPP and PSDE-PIPP, to perform dimensionality reduction and dimensionality expansion, respectively.

20.3 Representation of Transformed Components for DP

When the data are linearly transformed from the original data space to a new data space, each transformed data sample vector is essentially a linear combination or mixture of data sample vectors in the original space. In order to effectively represent the data in this new linearly transformed data space, it is desirable to find a set of basic constituent elements that can serve as a base for all the transformed data sample vectors. In this case, each basic constituent element represents a one-dimensional transformed component whose information significance can be measured by an information criterion. This section presents one such approach that can be considered as a generalization of PCA and ICA.

20.3.1 Projection Index-Based PP

In Section 6.5, an approach called projection pursuit (PP) is developed to generalize PCA and ICA to a PP-based component analysis transform in which the PP-transformed components are specified by projection vectors derived from a more general concept called projection index (PI). Such a PI-based PP is referred to as PIPP and its generated transformed components can be ranked by an information measure for DP. Although PIPP is already given in Section 6.5.1 of Chapter 6, we recap its details here for reference.

The term “PP”, as first coined by Friedman and Tukey (1974), was used to represent a technique for exploratory analysis of multivariate data. The idea is to project a high-dimensional data set into a low-dimensional data space while retaining the information of interest. It designs a PI to explore projections of interestingness. We assume that there are N data points $\{\mathbf{x}_n\}_{n=1}^{N}$, each with dimensionality K, and $\mathbf{X} = [\mathbf{x}_1\mathbf{x}_2 \cdots \mathbf{x}_N]$ is a $K \times N$ data matrix and $\mathbf{w}$ is a K-dimensional column vector that serves as a desired projection. Then $\mathbf{w}^T\mathbf{X}$ represents an N-dimensional row vector that is the orthogonal projections of all sample data points mapped onto the direction $\mathbf{w}$. Now if $H(\cdot)$ is a function measuring the degree of the interestingness of the projection $\mathbf{w}^T\mathbf{X}$ for a fixed data matrix $\mathbf{X}$, a PI is a real-valued function of $\mathbf{w}$, $I(\mathbf{w}) : R^K \to R$ defined by

$$I(\mathbf{w}) = H(\mathbf{w}^T\mathbf{X}) \tag{20.1}$$

The PI can be easily extended to multiple directions, $\{\mathbf{w}_j\}_{j=1}^{J}$. In this case, $\mathbf{W} = [\mathbf{w}_1\mathbf{w}_2 \cdots \mathbf{w}_J]$ is a $K \times J$ projection direction matrix and the corresponding projection index is also a real-valued function, $I(\mathbf{W}) : R^{K\times J} \to R$ is given by

$$I(\mathbf{W}) = H(\mathbf{W}^T\mathbf{X}) \tag{20.2}$$

The choice of the $H(\cdot)$ in (20.1) and (20.2) is application dependent. Its purpose is to reveal interesting structures within data sets such as clustering. The PP using PI specified by (20.2) is called PI-based project pursuit (PIPP). Within the context of PIPP, PCA and ICA can be considered as special cases of PIPP in which PCA uses data variance as a PI to produce eigenvectors, while ICA uses mutual information as a PI to produce statistically independent projection vectors. However, finding an optimal projection matrix $\mathbf{W}$ in (20.2) is not a simple matter, since there is no equation similar to the characteristic polynomial equation that can be used to find eigenvalues and eigenvectors analytically. In this case, the PI is confined to statistics of high orders, such as skewness, kurtosis, entropy, mutual information, information divergence (ID), etc., so that an equation can be used to solve a projection matrix $\mathbf{W}$ in (20.2) derived as follows.

We assume that the ith PI-projected transformed component is described by a random variable ζ_i, with values specified by the gray-level value of the nth pixel denoted by $z_n^i(\mathbf{z} = \mathbf{w}^T\mathbf{X})$. The original data set is first sphered to remove the mean and to make the covariance matrix an identity matrix. Let $\{\mathbf{r}_i\}_{i=1}^N$ denote the set of the sphered data sample vectors. A general form to be used to solve a projection matrix $\mathbf{W} = [\mathbf{w}_1\mathbf{w}_2 \cdots \mathbf{w}_J]$ with PI specified by the kth-order orders of statistics: kth moment is recently derived by Wang and Chang (2006a) and Ren et al. (2006) by solving the following eigen problem for $\mathbf{w}$:

$$\left(E\left\lfloor \mathbf{r}_i\left(\mathbf{r}_i^T\mathbf{w}\right)^{k-2}\mathbf{r}_i^T\right\rfloor - \lambda'\mathbf{I}\right)\mathbf{w} = 0 \text{ with } k > 2 \tag{20.3}$$

Specifically, for $k = 3, 4$, the form in (20.3) is called equations of skewness and kurtosis, respectively. Unlike the case of PCA, which solves eigenvalues of a data sample covariance matrix via the characteristic polynomial equation and then uses the obtained eigenvalues to obtain eigenvectors that specify its principal components (PCs), PIPP needs to solve (20.3) directly for the projection matrix $\mathbf{W} = [\mathbf{w}_1\mathbf{w}_2 \cdots \mathbf{w}_J]$ because there is no counterpart of a characteristic polynomial equation in PIPP that can be used to derive the $\mathbf{W}$. To do so, a general approach developed by Ren et al. (2006) and Wang and Chang (2006a) is to implement PIPP to produce components, referred to as projection index components (PICs) in which a PI is used as a criterion to find directions of interestingness of data to be processed and then represents the data in the data space specified by these new interesting directions.

Instead of finding the projection matrix $\mathbf{W} = \left[\mathbf{w}_1^*\mathbf{w}_2^* \cdots \mathbf{w}_k^*\right]$, an algorithm developed by Ren et al. (2006) for finding a sequence of projection vectors, $\mathbf{w}_1^*, \mathbf{w}_2^*, \ldots, \mathbf{w}_k^*$, to solve (20.3) can be described as follows.

Projection-index projection pursuit (PIPP) algorithm

1. Initial condition: $\mathbf{X} = [\mathbf{r}_1\mathbf{r}_2 \cdots \mathbf{r}_N]$ is a data matrix and a PI is specified.
2. The first projection vector $\mathbf{w}_1^*$ is found by maximizing the PI.
3. The obtained $\mathbf{w}_1^*$ is used to generate the first projection image $\mathbf{Z}^1 = \left(\mathbf{w}_1^*\right)^T\mathbf{X} = \left\{\mathbf{z}_i^1 | \mathbf{z}_i^1 = \left(\mathbf{w}_1^*\right)^T\mathbf{r}_i\right\}$ that can be used to detect the first projection vector.
4. The orthogonal subspace projector (OSP) specified by $P_{\mathbf{w}_1}^{\perp} = \mathbf{I} - \mathbf{w}_1(\mathbf{w}_1^T\mathbf{w}_1)^{-1}\mathbf{w}_1^T$ is applied the data set $\mathbf{X}$ to produce the first OSP-projected data set denoted by $\mathbf{X}^1$, $\mathbf{X}^1 = P_{\mathbf{w}_1}^{\perp}\mathbf{X}$.
5. The data set $\mathbf{X}^1$ is used to find the second projection vector $\mathbf{w}_2^*$ by maximizing the same PI again.
6. Let $P_{\mathbf{w}_2}^{\perp} = \mathbf{I} - \mathbf{w}_2\left(\mathbf{w}_2^T\mathbf{w}_2\right)^{-1}\mathbf{w}_2^T$ be applied to the data set $\mathbf{X}^1$ to produce the second OSP-projected data set denoted by $\mathbf{X}^2$, $\mathbf{X}^2 = P_{\mathbf{w}_2}^{\perp}\mathbf{X}^1$, which can be used to produce the third projection vector $\mathbf{w}_3^*$ by maximizing the same PI again. Or, equivalently, we define a matrix projection matrix $\mathbf{W}^2 = [\mathbf{w}_1\mathbf{w}_2]$ and apply $P_{\mathbf{W}^2}^{\perp} = \mathbf{I} - \mathbf{W}^2\left(\left(\mathbf{W}^2\right)^T\mathbf{W}^2\right)^{-1}(\mathbf{W}^2)^T$ to the data set $\mathbf{X}$ to obtain $\mathbf{X}^2 = P_{\mathbf{W}^2}^{\perp}\mathbf{X}$.
7. The procedure of steps 5 and 6 is repeated many times to produce $\mathbf{w}_3^*, \ldots, \mathbf{w}_k^*$ until a stopping criterion is met. It should be noted that a stopping criterion can be either a predetermined number of projection vectors required to be generated or a predetermined threshold for the difference between two consecutive projection vectors.

20.3.2 Mixed Projection Index-Based Prioritized PP (M-PIPP)

PIPP, as described in Section 20.3.1, uses the same PI to generate all the PICs. In general, it does not have to be this case. Since different PIs are designed to capture different details of information, it may be more effective in that PIPP can adapt its PI while it generates PICs. A similar idea proposed by Chai et al. (2007) who developed mixed PCA/ICA component analysis can also be applied to PIPP, referred to as mixed PIPP (M-PIPP) where the same PI used in step 5 in the above PIPP algorithm can be replaced and specified by different PIs. For example, the first, second, and third PICs can be produced by different PIs specified by variance, skewness, and kurtosis in order to represent the first data variance-specified principal component, the second skewness-specified component, and the kurtosis-specified component, respectively. The M-PIPP algorithm is exactly the same PIPP algorithm with the exception that the same PI implemented for all the components in step 5 can be replaced by various PIs, as specified by users. More details can be found in Safavi and Chang (2008) and Safavi (2010).

20.3.3 Projection Index-Based Prioritized PP (PI-PRPP)

According to PIPP described in Section 20.3.1, a vector is randomly generated as an initial condition used by PIPP to converge to a desired projection vector that is used to generate a PIC. As a consequence, a different randomly generated initial condition may converge to a different projection vector that also results in a different PIC. In other words, if PIPP is performed at different times or by different users, the resulting final PICs will also be different due to the use of different sets of random vectors. In order to correct this problem, this section presents a PI-based prioritized PP (PI-PRPP) that also uses a PI as a prioritization criterion to rank PIPP-generated PICs so that all PICs will be prioritized in accordance with the priorities measured by the given PI. Such a PI is called the PIC prioritization index. In this case, the PICs will always be ranked and prioritized by the PIC prioritization index in the same order regardless of what initial vectors are used to produce projection vectors. It must be noted that there is a major distinction between PIPP and PI-PRPP. While PIPP uses a PI as a criterion to produce a desired projection vector for each of PICs, the PI-PRPP uses a PIC prioritization index to prioritize PIPP-generated PICs. Therefore, the PIs used in both PP and PI-PRPP are not necessarily the same PI. In other words, the PI used to prioritize PICs as a PIC prioritization index can be different from the PI used to generate the PICs. As a matter of fact, on many occasions, different PIs can be used in applications. In what follows, we describe various criteria that use statistics beyond the second order and can be used to define a PIC prioritization index.

Projection index (PI)-based criteria

1. Sample mean of third-order statistics: skewness for ζ_j:

$$\mathrm{PI}_{\mathrm{skewness}}(\mathrm{PIC}_j) = \left[\kappa_j^3\right]^2 \tag{20.4}$$

where $\kappa_j^3 = E\left[\zeta_j^3\right] = (1/KN)\sum_{n=1}^{KN}\left(z_n^j\right)^3$ is the sample mean of the third order of statistics in the PIC_j.

2. Sample mean of fourth-order statistics: kurtosis for ζ_i:

$$\mathrm{PI}_{\mathrm{kurtosis}}(\mathrm{PIC}_j) = \left[\kappa_j^4\right]^2 \tag{20.5}$$

where $\kappa_j^4 = E\left[\zeta_j^4\right] = (1/KN)\sum_{n=1}^{KN}\left(z_n^j\right)^4$ is the sample mean of the fourth order of statistics in the PIC_j.

3. Sample mean of kth-order statistics: kth central moments for ζ_j:

$$\mathrm{PI}_{\text{k-moment}}(\mathrm{PIC}_j) = \left[\kappa_j^k\right]^2 \tag{20.6}$$

where $\kappa_j^k = E\left[\zeta_j^k\right] = (1/KN)\sum_{n=1}^{KN}\left(z_n^j\right)^k$ is the sample mean of the kth moment of statistics in the PIC$_j$.

4. Neg-entropy: combination of third and fourth orders of statistics for ζ_j:

$$\mathrm{PI}_{\text{negentropy}}(\mathrm{PIC}_j) = (1/12)\left[\kappa_j^3\right]^2 + (1/48)\left[\kappa_j^4 - 3\right]^2 \tag{20.7}$$

It should be noted that (20.7) is taken from (5.35) in Hyvarinen et al. (2001, p. 115), which is used to measure the neg-entropy by high-order statistics.

5. Entropy

$$\mathrm{PI}_{\text{entropy}}(\mathrm{PIC}_j) = -\sum_{j=1}^{KN} p_{ji}\log p_j \tag{20.8}$$

where $p_j = \left(p_{j1}, p_{j2}, \ldots, p_{jKN}\right)^T$ is the probability distribution derived from the image histogram of PIC$_i$.

6. Information divergence (ID)

$$\mathrm{PI}_{\mathrm{ID}}(\mathrm{PIC}_j) = \sum_{j=1}^{KN} p_{ji}\log\left(p_{ji}/q_i\right) \tag{20.9}$$

where $p_j = \left(p_{j1}, p_{j2}, \ldots, p_{jKN}\right)^T$ is the probability distribution derived from the image histogram of PIC$_i$ and $\mathbf{q}_j = \left(q_{j1}, q_{j2}, \ldots, q_{jKN}\right)^T$ is the Gaussian probability distribution with the mean and variance calculated from PIC$_{i.}$.

20.3.4 Initialization-Driven PIPP (ID-PIPP)

The PI-PRPP presented in Section 20.3.3 intended to remedy the issue that PICs could appear in a random order due to the use of randomly generated initial vectors. PI-PRPP allows users to prioritize PICs according to the information significance measured by a specific PIC prioritization index. Despite the fact that the PICs ranked by PI-PRPP may appear in the same order independent of different sets of random initial conditions, they are not necessarily identical because the slight discrepancy in two corresponding PICs at the same appearing order may be caused by randomness introduced by their used initial conditions. Although such a variation may be minor compared to different appearing orders of PICs without prioritization, the inconsistency may still cause difficulty in data analysis. Therefore, this section further develops a new approach, called initialization-driven PP (ID-PIPP), that custom-designs an initialization algorithm to produce a specific set of initial conditions for PIPP so that the same initial condition is always used whenever PIPP is implemented. Therefore, ID-PIPP-generated PICs are always identical. When a particular initial algorithm, say A, is used to produce a specific initial set of vectors for ID-PIPP to generate PICs, the resulting ID-PIPP is referred to as A-ID-PIPP.

One good candidate algorithm that can be used for this purpose is the automatic target generation process (ATGP) developed previously by Ren and Chang (2003). It makes use of an orthogonal subspace projector defined by (2.78)

$$P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\mathbf{U}^{\#} \tag{20.10}$$

where $\mathbf{U}^{\#} = \left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T$ is the pseudoinverse of the $\mathbf{U}$, in a repetitive manner to find target pixel vectors of interest from the data without prior knowledge regardless of what types of pixels these targets are. Details of implementing ATGP can be found in Section 8.5.1 and redescribed in the following steps.

Automatic target generation process

1. Initial condition:

   Let L be the total number of spectral bands.

   An initial target pixel vector of interest denoted by $\mathbf{t}_0$ is selected. In order to initialize ATGP without knowing $\mathbf{t}_0$, we select a target pixel vector with the maximum length as the initial target $\mathbf{t}_0$, namely, $\mathbf{t}_0 = \arg\{\max_{\mathbf{r}} \mathbf{r}^T\mathbf{r}\}$, which has the highest response, that is, the brightest pixel vector in the image scene. Set $n = 1$ and $\mathbf{U}_0 = [\mathbf{t}_0]$.

   (It is worth noting that this selection may not be necessarily the best selection. However, according to our experiments it was found that the brightest pixel vector was always extracted later on, provided that it was not selected as an initial target pixel vector in the initialization.)
2. At the nth iteration, $P_{\mathbf{t}_0}^{\perp}$ is applied via (20.10) to all image pixels $\mathbf{r}$ in the image and the nth target $\mathbf{t}_n$ generated at the nth stage is found, which has the maximum orthogonal projection as follows:

$$\mathbf{t}_n = \arg\left\{\max_{\mathbf{r}}\left\lfloor\left(P_{[\mathbf{U}_{n-1}\mathbf{t}_n]}^{\perp}\mathbf{r}\right)^T\left(P_{[\mathbf{U}_{n-1}\mathbf{t}_n]}^{\perp}\mathbf{r}\right)\right\rfloor\right\} \tag{20.11}$$

   where $\mathbf{U}_{n-1} = [\mathbf{t}_1\mathbf{t}_2\cdots\mathbf{t}_{n-1}]$ is the target matrix generated at the $(n-1)$st stage.
3. Stopping rule:

   If $n < L - 1$, let $\mathbf{U}_n = [\mathbf{U}_{n-1}\mathbf{t}_n] = [\mathbf{t}_1\mathbf{t}_2\cdots\mathbf{t}_n]$ be the nth target matrix, go to step 2. Otherwise, continue.
4. At this stage, ATGP is terminated. At this point, the target matrix is $\mathbf{U}_{L-1}$, which contains $L-1$ target pixel vectors as its column vectors, which do not include the initial target pixel vector $\mathbf{t}_0$.

As a result of ATGP, the final set of L target pixel vectors produced by ATGP at step 4, $S_{\text{ATGP}} = \{\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{L-1}\} = \{\mathbf{t}_0\} \cup \{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{L-1}\}$, will be used as an initial set of vectors to produce L PICs, where each of the L target pixels in S_{ATGP} is used to generate a particular PIC. An ID-PIPP using ATGP as its initialization algorithm is called ATGP-ID-PIPP.

20.4 Progressive Spectral Dimensionality Process

The main goal of introducing DP is to represent the original data in another data space via a linear transformation so that the transformed dimensions can be prioritized according to the significance of their contained information. Then, the data dimensionality in the new transformed data space can be increased or decreased by adding or removing transformed dimensions component-by-component progressively. The PI-PRPP and ID-PIPP introduced in Sections 20.3.3 and 20.3.4 are such linear transformations to satisfy the needed properties and can be used to realize DP. To be more specific, DP via PIPP is carried out by a set of pair parameters $\left\{\left(\mathbf{w}_j^*, \rho_j\right)\right\}_{j=1}^{L}$, where $\mathbf{w}_j^*$ is a projection vector produced by a PI that is used to generate the jth PIC and ρ_j is the priority score calculated for the jth PIC$_j$, that is, $\rho_j = \mathrm{PI}(\mathrm{PIC}_j)$ such that

$$\rho_j > \rho_k \Leftrightarrow \mathrm{priority_{PI}}(\rho_j) > \mathrm{priority_{PI}}(\rho_k)\ \mathrm{for}\, j, k \in \{1, 2, \ldots, L\}. \tag{20.12}$$

Using (20.12), all the L PIPP-generated PICs can be ranked from 1 to L according to their priorities in descending order as follows.

Let the priorities of all the PICs be arranged in descending order according to their priorities, $\mathrm{priority_{PI}}(\rho_{j_1}) \geq \mathrm{priority_{PI}}(\rho_{j_2}) \geq \cdots \geq \mathrm{prioirty_{PI}}(\rho_{j_L})$, where

$$j_1 = \arg\left\{\max_j \rho_j\right\} \quad j_2 = \arg\left\{\max_{j \neq j_1} \rho_j\right\}, \ldots,\ \mathrm{and}\, j_L = \arg\left\{min_j\, \rho_j\right\} \tag{20.13}$$

and $\{j_1, j_2, \ldots, j_L\}$ is a simply permutation of $\{1, 2, \ldots, L\}$. Then, the rank of all the L PICs is arranged by $\mathrm{rank}(\mathrm{PIC}_{j_1}) = 1, \mathrm{rank}(\mathrm{PIC}_{j_2}) = 2, \ldots, \mathrm{rank}(\mathrm{PIC}_{j_L}) = L.$ That is, $\mathrm{rank_{PI}}(\rho_j) \in \{1, 2, \ldots, L\}$ and

$$\mathrm{priority_{PI}}(\rho_j) > \mathrm{priority_{PI}}(\rho_k) \Leftrightarrow \mathrm{rank_{PI}}(\mathrm{PIC}_j) < \mathrm{rank_{PI}}(\mathrm{PIC}_k) \tag{20.14}$$

where the smaller number the rank (PIC$_j$), the higher priority the PIC$_j$. It is worth noting that for a generic representation there is no particular PI specified in $\rho_j = \mathrm{PI}(\mathrm{PIC}_j)$ in (20.12). However, for example, if PI is specified by skewness, then $\rho_j = \mathrm{PI}(\mathrm{PIC}_j)$ and $\mathrm{priority_{PI}}(\rho_j)$ are indicated as $\rho_j = \mathrm{PI_{skewness}}(\mathrm{PIC}_j)$ and $\mathrm{priority_{skewness}}(\rho_j)$, respectively. It should be also noted that the PIs used to generate $\mathbf{w}_j^*$ and the priority score of PIC$_j$ are not necessarily the same. This is because the projection vectors $\left\{\mathbf{w}_j^*\right\}_{j=1}^{L}$ are randomly generated by PIPP due to the use of random initial conditions. As a result, $\left\{\mathbf{w}_j^*\right\}_{j=1}^{L}$ appear in a random order where the PICs generated earlier do not imply that it is more significant than subsequently generated PICs. Even though the same PI is used for both $\mathbf{w}_j^*$ and ρ_j, the PICs may not appear in the same order. That is the reason why the PI-PRPP and ID-PIPP are introduced to fix this problem. When PI-PRPP is used, ρ_j is defined by PI such as (20.4)–(20.9), for example, $\rho_j = \mathrm{PI_{skewness}}(\mathrm{PIC}_j)$, $\mathrm{PI_{kurtosis}}(\mathrm{PIC}_j)$, $\mathrm{PI_{k\text{-}moment}}(\mathrm{PIC}_j)$, $\mathrm{PI_{neg\text{-}entropy}}(\mathrm{PIC}_j)$, $\mathrm{PI_{entropy}}(\mathrm{PIC}_j)$, $\mathrm{PI_{ID}}(\mathrm{PIC}_j)$, etc. One the other hand, if ID-PIPP is used, the ρ_j is defined by the order that the jth target is generated by the used initialization algorithm, that is, $\mathrm{priority_{PI}}(\rho_j) = j$. For example, when ATGP is used as an initialization algorithm, $\rho_j = j$, when the jth target $\mathbf{t}_j$ is generated by ATGP to specify the jth PIC, PIC$_j$.

By virtue of $\left\{\left(\mathbf{w}_j^*, \rho_j\right)\right\}_{j=1}^{L}$ spectral dimensionality processing can be performed by starting from any number and then progressively adding or removing data dimensions specified by their corresponding projection vectors $\left\{\mathbf{w}_j^*\right\}_{j=1}^{L}$ in accordance with their assigned ranks or priority scores by $\left\{\text{rank}_{\text{PI}}(\text{PIC}_j)\right\}_{j=1}^{L}$ or $\left\{\text{priority}_{\text{PI}}(\rho_j)\right\}_{j=1}^{L}$ specified by (20.14). In what follows, various progressive versions of commonly used component analyses specified by PIPP are developed for the progressive spectral dimensionality process (PSDP).

20.4.1 Progressive Principal Components Analysis

The simplest PIPP is PCA where PI is specified by data variance. Using the same notations and equations in Chapter 6, assume that $\{\mathbf{r}_i\}_{i=1}^{N}$ is a set of L-dimensional data sample vectors and $\mathbf{K}$ is the data sample covariance matrix with eigenvlaues given by $\{\lambda_l\}_{l=1}^{L}$ that are arranged in the decreasing order of their magnitudes and associated eigenvectors denoted by $\{\mathbf{v}_l\}_{l=1}^{L}$. We can define an eigenvector matrix $\mathbf{\Lambda}$ specified by $\{\mathbf{v}_l\}_{l=1}^{L}$ as

$$\mathbf{\Lambda} = [\mathbf{v}_1 \mathbf{v}_2 \ldots \mathbf{v}_L] \tag{6.11}$$

With the eigenvector matrix $\mathbf{\Lambda}$, a linear transform $\xi_{\mathbf{\Lambda}}$ defined by

$$\xi_{\mathbf{\Lambda}}(\mathbf{r}) = \mathbf{\Lambda}^T \mathbf{r} \tag{6.13}$$

transforms every data sample $\mathbf{r}_i$ into a new data sample in such a manner that all the $\xi_{\mathbf{\Lambda}}$-transferred data samples are uncorrelated. The transform $\xi_{\mathbf{\Lambda}}$ defined by (6.13) is, in general, called eigen-transformation. Let $\mathbf{X}$ and $\tilde{\mathbf{X}}$ be the original data matrix and the $\xi_{\mathbf{\Lambda}}$-transformed matrix, respectively. The lth component of $\tilde{\mathbf{X}}$ is formed by

$$\xi_{\mathbf{v}_l}(\mathbf{X}) = \mathbf{v}_l^T \mathbf{X} \tag{6.16}$$

and is called the lth PC that consists of $\{\tilde{\mathbf{r}}_i\}_{i=1}^{N}$ with $\tilde{\mathbf{r}}_i = \mathbf{v}_l^T \mathbf{r}_i$ that are $\xi_{\mathbf{v}_l}$-transformed data samples corresponding to the lth eigenvalue λ_l.

Depending on how to generate projection vectors as eigenvectors, four different versions of implementing PCA are derived to generate PCs, each of which has its own right. The first version is the commonly used PCA, referred to as simultaneous PCA (SM-PCA) that finds all eigenvectors to produce all PCs simultaneously by solving the characteristic polynomial equation. The second one is referred to as progressive PCA (PG-PCA) that finds the projection vectors that correspond to the maximal eigenvalues in a nested sequence of orthogonal subspaces to produce all the PCs one at a time progressively. The third one is referred to as sequential PCA (SQ-PCA) that produces a projection vector to generate a PC at a time sequentially. The fourth one is referred to as initialization-driven PCA (ID-PCA) that appeals for a custom-designed initialization algorithm to prioritize projection vectors so that the PCs can be ranked and generated sequentially and progressively. It should be noted that the ideas of developing the SM-PCA, SQ-PCA, and ID-PCA are similar to those used to develop SM-EEAs in Chapter 7, SQ-EEAs in Chapter 8, and ID-EEAs in Chapter 9.

20.4.1.1 Simultaneous PCA

The simultaneously PCA (SM-PCA) has been the common approach used in signal processing. It solves the so-called characteristic polynomial equation obtained by the data sample covariance matrix $\mathbf{K}$ to find all the eigenvalues $\{\lambda_l\}_{l=1}^{L}$ and their corresponding eigenvectors $\{\mathbf{v}_l\}_{l=1}^{L}$. Then, each eigenvector found is used to specify one PC via (6.16). Theoretically, the SMPCA can generate all the PCs simultaneously.

20.4.1.2 Progressive PCA

In many applications, there is no need to generate all PCs for data analysis since those PCs that correspond to smallest eigenvalues may correspond to noise components and contain no useful information. In this case, it may be more realistic to develop a progressive version of PCA, referred to as progressive PCA (PG-PCA) that can generate PCs in a progressive manner so that the information contained in PCs is gradually decreased in terms of a certain criterion. Of course, the SM-PCA can also be implemented as PG-PCA by generating PCs that are specified by eigenvectors in accordance with the magnitude of their corresponding eigenvalues progressively one at a time. However, in this section, we describe another progressive version of PCA that provides an alternative approach to PG-PCA.

First, we find the largest eigenvalue of the data sample covariance matrix $\mathbf{K}_{L\times L}$ denoted by $\lambda_{\max}$, where the subscript "$L \times L$" is included to indicate the dimensionality of the matrix. Let $\mathbf{v}_1$ be an eigenvector associated with $\lambda_{\max}$ that represents the first PC. Then, we apply an orthogonal subspace projection (OSP) operator defined in (6.76) by $P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T$ with $\mathbf{U} = [\mathbf{v}_{\max}]$ to obtain a subspace $\tilde{\mathbf{X}}^1$ that is orthogonal to the space linearly spanned by $\langle \mathbf{v}_{\max} \rangle$ with the covariance matrix denoted by $\mathbf{K}^1_{L\times L}$. Next, we find the largest eigenvalue of the data sample covariance matrix, $\mathbf{K}^1_{L\times L}$, denoted by λ_2, and let $\mathbf{v}_2$ be an eigenvector associated with λ_1 that represents the second PC. Once again, we apply $P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T$ with $\mathbf{U} = \left[\mathbf{v}_{\max}\mathbf{v}^1_{\max}\right]$ to obtain a subspace $\tilde{\mathbf{X}}^2$ that is orthogonal to the space spanned linearly by $\langle \mathbf{v}_{\max}, \mathbf{v}^1_{\max} \rangle$ with the covariance matrix denoted by $\mathbf{K}^2_{L\times L}$. The same process is then repeated until the last PC is generated and described as follows.

PG-PCA algorithm

1. *Initialization*: form the original data sample covariance matrix, $\mathbf{K}^{(0)}_{L\times L}$, and find the largest eigenvalue $\lambda^{(0)}_{\max}$ along with its corresponding eigenvector $\mathbf{v}^{(0)}_{\max}$. Let $n = 0$.
2. Apply an OSP operator $P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T$ with $\mathbf{U}^{(0)} = \left[\mathbf{v}^{(0)}_{\max}\right]$ to obtain a subspace $\tilde{\mathbf{X}}^1$ that is orthogonal to the space spanned linearly by $\langle \mathbf{v}^{(0)}_{\max} \rangle$.
3. Let $n \leftarrow n + 1$. Form the data covariance matrix $\mathbf{K}^{(n)}_{L\times L}$ from the space $\tilde{\mathbf{X}}^{(n)}$ and find the largest eigenvalue $\lambda^{(n)}_{\max}$ along with its corresponding eigenvector $\mathbf{v}^{(n)}_{\max}$.
4. Apply $P_{\mathbf{U}^{(n)}}^{\perp} = \mathbf{I} - \mathbf{U}^{(n)}\left[\left(\mathbf{U}^{(n)}\right)^T\mathbf{U}^{(n)}\right]^{-1}\left(\mathbf{U}^{(n)}\right)^T$ with $\mathbf{U}^{(n)} = \left[\mathbf{v}_{\max}\mathbf{v}^{(1)}_{\max}\ldots\mathbf{v}^{(n)}_{\max}\right]$ to obtain a subspace $\tilde{\mathbf{X}}^{(n+1)}$ that is orthogonal to the space spanned linearly by $\langle \mathbf{v}_{\max}, \mathbf{v}^{(1)}_{\max}, \ldots, \mathbf{v}^{(n)}_{\max} \rangle$.
5. If $n = L$, the algorithm is terminated. Otherwise, go to step 3.

Using the above PG-PCA algorithm, a sequence of desired PCs can be generated progressively with the information contained in PCs being gradually reduced in accordance with decreasing data variances. The key difference between the SM-PCA and PG-PCA is that the former requires finding all eigenvalues of the sample covariance matrix formed by the original data space $\mathbf{X}$, while the latter only needs to find the maximal variance of each of the sample covariance matrices $\left\{\mathbf{K}^{(n)}_{L\times L}\right\}_{n=0}^{L-1}$ formed by a nest sequence of orthogonal subspaces, $\mathbf{X} \supset \tilde{\mathbf{X}}^1 \supset \tilde{\mathbf{X}}^2 \supset \cdots \supset \tilde{\mathbf{X}}^{L-1}$.

20.4.1.3 Sequential PCA

In this section, we describe another version of PCA, referred to as sequential PCA (SC-PCA), that can generate PCs without solving eigenvalues in the same way as SM-PCA and PG-PCA generate their PCs in Sections 20.4.1.1 and 20.4.1.2.

It begins with an initial projection vector $\mathbf{v}_1^{(0)}$ generated in a random manner. Then, we find the maximum of $\left(\tilde{\mathbf{r}}_j^{(0)}\right)^T \tilde{\mathbf{r}}_j^{(0)}$ over $\tilde{\mathbf{r}}_j^{(0)} = \mathbf{v}_1^{(0)}\left(\mathbf{v}_1^{(0)}\right)^T \mathbf{r}_j$, denoted by $\mathbf{v}_1^{(1)} = \arg\left(\max_{\tilde{\mathbf{r}}_j^{(0)}}\left[\left(\mathbf{r}_j^{(0)}\right)^T \tilde{\mathbf{r}}_j^{(0)}\right]\right)$ that will be used in the next iteration. This process is repeated until the two vectors, $\mathbf{v}_1^{(n)}$ and $\mathbf{v}_1^{(n+1)}$, generated in two consecutive iterations are sufficiently close within a threshold ε and the iterative procedure is terminated. In this case, $\mathbf{v}_1^n \to \mathbf{v}_1$ converges to the vector $\mathbf{v}_1$ that is the desired eigenvector. Then, we apply $P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T$ with $\mathbf{U} = [\mathbf{v}_1]$ to $\mathbf{X}$ in order to obtain a subspace $\tilde{\mathbf{X}}^1$ that is orthogonal to the space spanned linearly by $\langle \mathbf{v}_1 \rangle$. The same procedure described above is repeated for finding a second eigenvector $\mathbf{v}_2$ by randomly generating an initial vector $\mathbf{v}_2^{(0)}$. It should be noted that there are two processes that are involved to find random PCs. The first process is to use a random initial vector to generate an eigenvector and is then followed by a second process that is the same one carried out by PG-PCA in Section 20.4.1.2. However, due to its use of random initial vectors, these PCs are generated sequentially but not progressively. This is because the projection vectors $\left\{\mathbf{v}_j\right\}$ produced for PCs are not necessarily exactly the same eigenvectors generated by the SM-PCA and PG-PCA despite that they may be very close. Even in the case in which the projection vectors $\left\{\mathbf{v}_j\right\}$ are eigenvectors, their generated PCs are not necessarily in descending order of their corresponding eigenvalues as the order that PCs are generated and ranked by SM-PCA and PG-PCA. Nevertheless, a major advantage of SQ-PCA is that it does not require calculation of eigenvalues as SM-PCA and PG-PCA do. Because of that, for each $1 \le j \le L$, a learning algorithm is needed to find the desired projection vector from $\tilde{\mathbf{X}}^j$ without finding eigenvalues which can be obtained by solving the characteristic polynomial equation. Two learning algorithms are designed for this purpose and allow a randomly generated projection vector to automatically converge to a desired projection vector which will be used to produce a desired PC. One is referred to as a maximal projection learning algorithm and is described as follows.

Maximal projection learning algorithm for $\tilde{\mathbf{X}}^j$

1. *Initialization*: generate randomly a projection vector $\mathbf{v}_j^{(0)}$. Set a threshold ε.
2. Find the maximum of $\left(\tilde{\mathbf{r}}_k^{(0)}\right)^T \tilde{\mathbf{r}}_k^{(0)}$ over $\tilde{\mathbf{r}}_k^{(0)} = \mathbf{v}_1^{(0)}\left(\mathbf{v}_1^{(0)}\right)^T \mathbf{r}_k$ with $\mathbf{r}_k \in \tilde{\mathbf{X}}^j$, denoted by

$$\mathbf{v}_1^{(1)} = \arg\left\{\max_{\tilde{\mathbf{r}}_k^{(0)}}\left[\left(\mathbf{r}_k^{(0)}\right)^T \tilde{\mathbf{r}}_k^{(0)}\right]\right\} \tag{20.15}$$

   If the distance between $\mathbf{v}_j^{(0)}$ and $\mathbf{v}_j^{(1)}$ is less than ε, the algorithm is terminated. Otherwise, continue. The distance can be any measure, such as Euclidean distance or spectral angle mapper (SAM).
3. Let $n \leftarrow n + 1$.
4. Find the maximum of $\left(\tilde{\mathbf{r}}_k^{(n)}\right)^T \tilde{\mathbf{r}}_k^{(n)}$ over $\tilde{\mathbf{r}}_k^{(n)} = \mathbf{v}_j^{(n)}\left(\mathbf{v}_j^{(n)}\right)^T \mathbf{r}_k$, denoted by $\mathbf{v}_1^{(n+1)} = \arg\left\{\max_{\tilde{\mathbf{r}}_k^{(n)}}\left[\left(\mathbf{r}_k^{(n)}\right)^T \tilde{\mathbf{r}}_k^{(n)}\right]\right\}$. If the distance between $\mathbf{v}_j^{(n)}$ and $\mathbf{v}_j^{(n+1)}$ is less than ε, the algorithm is terminated and let $\mathbf{v}_j^n = \mathbf{v}_j^{(n+1)}$. Otherwise, go back to step 3.

A second learning algorithm is based on the concept of gradient descent (Bischop, 1995, pp. 95–96) that can also be considered as modified from a fixed-point algorithm developed by Hyvarinen and Oja (1997).

Assume that the mean squared error is given by

$$J(\mathbf{w}) = E\left[\left(\mathbf{w}^T\mathbf{x} - E\left[\mathbf{w}^T\mathbf{x}\right]\right)^2\right] \tag{20.16}$$

Also, the weighting vector $\mathbf{w}^{(n)}$ at the nth iteration can be updated by

$$\mathbf{w}^{(n+1)} = \mathbf{w}^{(n)} - \eta \left.\frac{\partial J(\mathbf{w}^{(n)})}{\partial \mathbf{w}^{(n)}}\right|_{w^{(n)}} \tag{20.17}$$

where η is the learning parameter to be determined by applications and the gradient of the $J(\mathbf{w})$ in (20.16) can be calculated as follows:

$$\begin{aligned}
\frac{\partial J(\mathbf{w}^{(n)})}{\partial \mathbf{w}^{(n)}} &= E\left[\frac{\partial}{\partial \mathbf{w}}\left(\left(\left(\mathbf{w}^{(n)}\right)^T\mathbf{x} - E\left[\left(\mathbf{w}^{(n)}\right)^T\mathbf{x}\right]\right)\right)^2\right] \\
&= E\left[2\left(\left(\mathbf{w}^{(n)}\right)^T\mathbf{x} - E\left[\left(\mathbf{w}^{(n)}\right)^T\mathbf{x}\right]\right)\left(\mathbf{x} - \frac{\partial E\left[\left(\mathbf{w}^{(n)}\right)^T\mathbf{x}\right]}{\partial \mathbf{w}^{(n)}}\right)\right] \\
&= E\left[2\left(\left(\mathbf{w}^{(n)}\right)^T\mathbf{x}\right)\mathbf{x} - 2\left(\left(\mathbf{w}^{(n)}\right)^T\mathbf{x}\right)E[\mathbf{x}] - 2E\left[\left(\mathbf{w}^{(n)}\right)^T\mathbf{x}\right]\mathbf{x} + 2E\left[\left(\mathbf{w}^{(n)}\right)^T\mathbf{x}\right]E[\mathbf{x}]\right] \\
&= E\left[2\mathbf{x}\left(\mathbf{x}^T\left(\mathbf{w}^{(n)}\right)\right) - 2E[\mathbf{x}]\left(\mathbf{x}^T\left(\mathbf{w}^{(n)}\right)\right)\right] \\
&= 2E[\mathbf{x}\mathbf{x}^T]\mathbf{w}^{(n)} - 2E[\mathbf{x}]E[\mathbf{x}^T]\mathbf{w}^{(n)} = 2\mathbf{K}\mathbf{w}^{(n)}
\end{aligned} \tag{20.18}$$

Substituting (20.18) for $\left.\frac{\partial J(\mathbf{w}^{(n)})}{\partial \mathbf{w}^{(n)}}\right|_{w^{(n)}}$ into (20.17) yields the following learning rule for iterations:

$$\mathbf{w}^{(n+1)} = \mathbf{w}^{(n)} - 2\eta\mathbf{K}\mathbf{w}^{(n)} \tag{20.19}$$

Gradient descent learning algorithm for $\tilde{\mathbf{X}}^j$

1. *Initialization*: generate randomly a normalized projection vector $\mathbf{v}_j^{(0)}$ to unit vector. Set a threshold ε. Set $n = 0$.
2. At the nth iteration, use (20.19) as a learning rule with the weight vector $\mathbf{w}^{(n)}$ in (20.19) being replaced by $\mathbf{v}_j^{(n)}$ to produce a normalized projection vector $\mathbf{v}_j^{(n+1)}$ to a unit vector to be used in the next iteration.
3. If the distance between $\mathbf{v}_j^{(0)}$ and $\mathbf{v}_j^{(1)}$ is less than ε, the algorithm is terminated. Otherwise, let $n \leftarrow n + 1$ and go to step 2. It should be noted that the distance can be any measure, such as Euclidean distance or SAM.

With the help of the two learning algorithms developed above, we are ready to describe an algorithm that implements SQ-PCA to generate PCs successively in sequence.

SQ-PCA algorithm

1. *Initialization*: for the first PC, let $j = 1$.
2. Apply either "maximal projection learning algorithm" or "gradient descent learning algorithm" to find the jth projection vector $\mathbf{v}_j$ that produces the jth random PC.
3. Apply $P_{\mathbf{U}_j}^{\perp} = \mathbf{I} - \mathbf{U}_j\left[(\mathbf{U}_j)^T\mathbf{U}_j\right]^{-1}(\mathbf{U}_j)^T$ with $\mathbf{U}_j = [\mathbf{v}_1\mathbf{v}_2 \ldots \mathbf{v}_j]$ to obtain a subspace $\tilde{\mathbf{X}}^{(j+1)}$ that is orthogonal to the space spanned linearly by $\langle \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_j \rangle$. Let $j \leftarrow j + 1$
4. If $j = L$, the algorithm is terminated, in which case all L PCs have been generated. Otherwise, go to step 2.

20.4.1.4 Initialization-Driven PCA

A fourth version of PCA, referred to as initialization-driven PCA (ID-PCA), is derived from an idea similar to that used to develop initialization-driven EEAs (ID-EEAs) in Chapter 9. Unlike SQ-PCA that uses random initial projection vectors, it makes use of a custom-designed initialization algorithm to produce an appropriate set of initial projection vectors for SQ-PCA. As a consequence, the PCs generated by SQ-PCA in conjunction with an initialization algorithm can be prioritized by the projection vectors generated by the initialization algorithm. Such prioritized PCs make ID-PCA not only sequential but also progressive, because these PCs can be ranked by their associated priority scores that represent significance of information contained in PCs.

One algorithm that can be used for ID-PCA is once again ATGP, as described in Section 8.5.1 Using ATGP as an initialization algorithm, ID-PCA can be derived as follows.

ID-PCA algorithm

1. Use ATGP to produce a set of projection vectors $\{\mathbf{t}_j\}_{j=1}^{L}$ arranged in their appearing orders.
2. Using each of $\{\mathbf{t}_j\}_{j=1}^{L}$ generated in step 1 as an initial projection vector, implement SC-PCA to produce $\{\mathrm{PC_i}\}_{i=1}^{L}$ that maximize $E\left[\tilde{\mathbf{r}}_i\tilde{\mathbf{r}}_i^T\right]$ in accordance with priorities determined by the order that $\{\mathbf{t}_j\}_{j=1}^{L}$ are generated by ATGP.

Several remarks are noteworthy and summarized as follows:

1. Despite the fact that PG-PCA only finds the maximal variance in each of nested orthogonal subspaces sequentially, it actually finds all the eigenvalues as SM-PCA in decreasing order does.
2. Theoretically, SM-PCA and PG-PCA can analytically solve the characteristics polynomial equation for all the eigenvalues that are distinct. For each distinct eigenvalue, an eigenvector can be generated to produce a PC. Therefore, a total of L PCs can be produced with L being the total number of bands where some PCs generated by eigenvectors corresponding to smaller eigenvalues may turn out to be noise components. By contrast, SQ-PCA and ID-PCA do not solve the characteristics polynomial equation for all the eigenvalues. Instead, they use either random initial projection vectors or initialization-driven projection vectors to generate desired PCs. In this case, those PCs that are specified by projection vectors as eigenvectors with very close data variances as eigenvalues will be considered the same PCs. As a result, only one PC may be generated to represent these PCs. Such circumstances often happen to noise components whose corresponding eigenvalues are usually relatively small and nearly the same. Consequently, the number of all possible PCs that SQ-PCA and ID-PCA can generate is, in general, much smaller than the total number of bands, L.

3. One of the advantages that SQ-PCA and ID-PCA have over SM-PCA and PG-PCA is no need of determination of how many components are needed to be preserved. Since SM-PCA and PG-PCA produce all the L PCs that may include noise components, they must rely on a criterion to determine the number of components, q, needed to be retained after PG-CA. The VD concept introduced in Chapter 5 can be used to estimate the value of q for this purpose. Conversely, SQ-PCA and ID-PCA only produce PCs with their data variances as eigenvalues distinct from each other without accounting for noise components; the number of their generated PCs should reflect and indicate the number of spectral distinct spectral signatures present in the data, in which case they automatically determine the value of p. Since VD was originally developed to estimate the number of spectrally distinct signatures, the value of q determined by SQ-PCA and ID-PCA is expected to be greater than that estimated by VD due to the fact that VD is determined by the false-alarm probability and the number of PCs generated by SQ-PCA and ID-PCA is determined by how close the projection vectors are that produce the PCs. With this interpretation, we can always use the value of q estimated by VD and the value of q determined by SQ-PCA and ID-PCA as lower and upper bounds to determine how many PCs are needed to be retained after PG-PCA.

20.4.2 Progressive High-Order Statistics Component Analysis

When PCA and ICA are presented in Chapter 6, they are considered two different approaches. It was shown by Wang and Chang (2006a) that PCA can only capture information characterized by second-order statistics and small targets could not be preserved in PCs but rather in independent components (ICs). This evidence was further confirmed by Ren et al. (2006), where high-order statistics (HOS) are also able to detect subtle targets, such as anomalies and small targets. However, there is a significant difference between PCA and ICA.

20.4.3 Progressive Independent Component Analysis

When PCA and ICA were presented in Chapter 6, they are considered two different approaches. This is because PCA-generated PCs are specified by eigenvectors whose corresponding eigenvalues can be found by solving the characteristic polynomial equation and ICA-generated ICs are produced by FastICA that uses a learning algorithm, called fixed-point algorithm to generate each of ICs by a convergent process beginning with a random initial projection vector. In order to explore the connection between PCA and ICA in the sense of how their components are generated, PCA was investigated and rederived in Sections 20.4.1.3 and 20.4.1.4 by using learning algorithms to directly find projection vectors that produce PCs instead of finding eigenvalues and their corresponding eigenvectors to specify PCs. As a result, two new learning algorithm-based PCA, SQ-PCA, and ID-PCA were developed in Sections 20.4.1.3 and 20.4.1.4. As recalled in Section 6.4, there were three versions of ICA, SP-ICA-DR, R-ICA-DR, and ID-ICA-DR derived for DR. Interestingly, if we conduct a one-to-one correspondence comparison between PCA derived in Section 20.4.1 and ICA derived in Section 6.4, PG-PCA/SQ-PCA and ID-PCA turn out to be the counterparts of the SP-ICA-DR and ID-ICA-DR, respectively, where a learning algorithm used in FastICA to produce ICs is analogous to the "gradient descent learning algorithm" used in SQ-PCA and ATGP was used in both ID-PCA and ID-ICA-DR for initialization. Such a close relationship allows us: (1) to develop high-order statistics-based component analysis that bridges the gap between the second-order statistics-based PCA and statistical independence-based ICA, and (2) also to design criteria for DP in the following section.

20.5 Hyperspectral Compression by PSDP

PSDP provides a backbone for two major dual processes, progressive spectral dimensionality reduction (PSDR) and progressive spectral dimensionality expansion (PSDE), that can be used as a data dimensionality process for hyperspectral information compression.

20.5.1 Progressive Spectral Dimensionality Reduction

PSDR is a process that allows users to reduce a number of spectral components gradually by removing one spectral component at a time with decreased spectral component dimensionality priorities. It starts with a maximal number of spectral components and begins to reduce a small number of spectral components at a time by eliminating spectral components with lowest priorities until performance of data processing is not satisfied or it reaches the minimal number of spectral components that can be determined by VD. By implementing PSDR more and more hyperspectral information compression can be achieved by gradually reducing spectral information with removal of additional spectral components from the set of spectral components currently being considered. In what follows, we describe its implementation in detail.

PSDR

1. Prioritize all the transformed components via a DP criterion.
2. *Initialization*: Use VD to determine the minimal number of transformed components required to be retained, denoted by n_{VD}. Let n_{initial} be the number of spectral components for the process to begin with and n_Δ be the step size of dimensions to reduce. Set $k = 0$.
3. Evaluate the performance of data processing to see if it is satisfied. If it is not, the process is terminated. Otherwise, let $k \leftarrow k + 1$ and continue.
4. At the kth reduction, eliminate the next set of n_Δ least prioritized spectral components from the set of spectral components currently being processed. In this case, the total number of spectral components to be processed is reduced from $n_{\mathrm{initial}} - (k-1) \cdot n_\Delta$ to $n_{\mathrm{initial}} - k \cdot n_\Delta$.
5. If $(n_{\mathrm{initial}} - k \cdot n_\Delta) > n_{\mathrm{VD}}$, go to step 6. Otherwise, the process is terminated.

20.5.2 Progressive Spectral Dimensionality Expansion

As a completely opposite process of PSDR, PSDE performs reversely in the way that PSDR does. The process is terminated only if the performance of data processing is satisfied or it reaches the maximal number of spectral components preset *a priori*. By means of PSDE the information available to data processing is gradually increased by adding more spectral components according to their priority scores in descending order, in which case less and less hyperspectral information compression can be achieved. A detailed implementation of PSDE is summarized as follows.

PSDE

1. Prioritize all the transformed components via a DP criterion.
2. *Initialization*: use VD to determine an initial number of transformed components needed to be started with, denoted by n_{VD}. Let n_{T} be the total number of spectral components that are pre determined to terminate the process and n_Δ be the step size of dimensions to expand. Set $k = 0$.
3. Start with the first n_{VD} highest prioritized transformed components.
4. Evaluate the performance of data processing to see if it is satisfied. If it is, the process is terminated. Otherwise, let $k \leftarrow k + 1$ and continue.

5. At the kth expansion, add the next n_Δ highest prioritized spectral components. In this case, the total number of resultant spectral components is increased from $n_{\mathrm{VD}} + (k-1) \cdot n_\Delta$ to $n_{\mathrm{VD}} + k \cdot n_\Delta$.
6. If $(n_{\mathrm{VD}} + k \cdot n_\Delta) < n_{\mathrm{T}}$, go to step 6. Otherwise, the algorithm is terminated.

Two notes are worth mentioning.

a. One is the n_{T} used to terminate to PSDE. In many applications, it is usually sufficient by setting $n_{\mathrm{T}} = 2n_{\mathrm{VD}}$.
b. In step 5 of PSDE, the number of spectral components added at each iteration, n_Δ, is assumed to be arbitrary. In general, this number can be set by one, that is, $n_\Delta = 1$ in which case at each iteration in step 5, the number of added components will be increased from $n_{\mathrm{VD}} + (k-1)$ to $n_{\mathrm{VD}} + k$.

The above two notes made for PSDE are also applied to PSDR summarized as follows.

a. In many applications, the n_{T} used to start PSDR is usually sufficient by setting $n_{\mathrm{T}} = 2n_{\mathrm{VD}}$.
b. Like PSDE, the step size n_Δ can also be set to 1 so that the number of spectral components is decreased by one at a time during each iteration of PSDR.

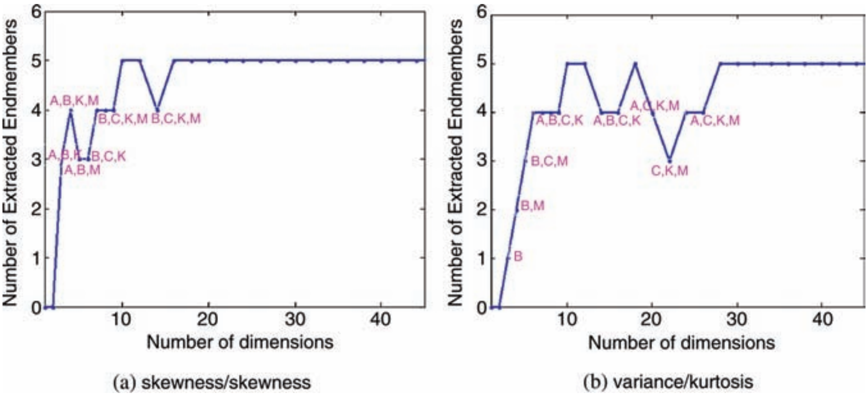
Finally, it is worth noting that according to the manner in which PSDR and PSDE operate on spectral components, PSDE can be considered a sequential forward process that increases spectral information by adding more spectral components, while PSDR can be viewed as a sequential backward process, which is a reverse process of PSDE and reduces spectral information by eliminating more spectral components from the set of spectral components currently being considered. Interestingly, if we consider the projection vectors $\left\{\mathbf{w}_j^*\right\}_{j=1}^{L}$ as feature vectors, the pair of PSDR and PSDE is similar to the pair of sequential forward selection (SFS) and sequential backward selection (SBS) that were developed for feature selection by Serpico and Bruzzone (2001). But, there is a key difference between them that is the latter pair does not prioritize features for feature selection. So, if the concept of prioritizing projection vectors is introduced into SFS and SBF, the resulting pair will become progressive SFS and progressive SBF, in which case the features can be selected based on their priority scores progressively without repeatedly resolving feature optimization problems.

20.6 Experiments for PSDP

This section conducts real image experiments to substantiate the utility of PSDP in a wide range of data exploitation. Since PSDP is preprocessing and very crucial to subsequent data analysis, its effectiveness can only be demonstrated through applications.

20.6.1 Endmember Extraction

A first immediate application of PSDP is endmember extraction that generally requires data dimensionality reduction. The data used for experiments for endmember extraction was the Cuprite data in Figure 1.12(a), where IN-FINDR was used as an endmember extraction algorithm to extract the five mineral signatures, A, B, C, K, and M, of major interest in the scene. Because of too many combinations of PI/PI that can be used for PSDP for an illustrative purpose, Figure 20.1 (a) and (b) only plots results of extracting the mineral signatures by N-FINDR using PSDP with



(a) skewness/skewness

(b) variance/kurtosis

Figure 20.1 Endmembers extracted by IN-FINDR as PSDP implemented.

PI/PI = skewness/skewness and variance/kurtosis, respectively, where the x-axis is the number of prioritized dimensions and the y-axis is the endmembers extracted by N-FINDR. When the number of extracted endmembers was less than five, we need to know which endmembers were extracted in which case the extracted endmembers were also specified for reference. For example, in Figure 20.1(a) when PSDP was implemented by a pair of PIs, PI = skewness/PI = skewness for $q = 4$, 5, the number of extracted endmembers was 3 with extracted endmembers being identified as A, B, M. However, as q was increased to 6, the number of extracted endmembers was still 3. But this time, three extracted signatures B, C, K had only one signature B in common. This implies that if q was too small, the results were not stable even though the same number of endmembers was extracted by different values of q. Interestingly, when $q = 10$ all the five mineral signatures were successfully extracted by IN-FINDR which failed again at $q = 16$ in which case it missed the signature A until it reached $q = 18$ where the stability was also reached. In other words, after q went beyond 18 all the five endmembers were extracted. This shows that $n_{\mathrm{VD}} = 22$ provided a good estimate for q.

However, if the commonly used PCA was used to perform DR in which case PI = variance and the PICs (in this case, PICs are reduced to PCs) were ranked by another PI = kurtosis, Figure 20.1 (b) shows the results of endmember extraction versus the number of prioritized dimensions where the stability reached after $q = 28$. A surprising coincidence was observed by comparing Figure 20.1(a) and (b), where IN-FINDR could extract all five mineral signatures for both cases when $q = 10$–12 and 18. This simple example demonstrated that an appropriate value of q must be determined by the PI used to generate PICs and another PI used to rank the order of PICs. None of DR techniques can claim to perform better than another unless criteria for DP are specified.

20.6.2 Land Cover/Use Classification

As another example, the Purdue Indiana Indian Pine Test Site in Figure 1.13 was also used for experiments. This particular scene is probably one of most studied test sites in hyperspectral data exploitation for land cover/use classification. Because of its low 20 m spatial resolution, most data sample vectors in this scene are heavily mixed because of large spectral variations of class signatures. As a consequence, spectral unmixing did not work effectively for this scene because finding an appropriate set of creditable signatures to unmix data sample vectors was very challenging, if not impossible. Instead, the maximum likelihood classifier (MLC) has widely been used for this particular scene in the literature. Here, MLC was also used for this purpose. Assume that S_j represents a training sample set selected for the jth class and $\cup_j S_j$ is the entire training set. The mean and sample covariance matrix of $\cup_j S_j$ are calculated by $\boldsymbol{\mu} = \frac{1}{|\cup_j S_j|} \sum_{\mathbf{r} \in \cup_j S_j} \mathbf{r}$ and

$$\mathbf{K} = \frac{1}{|\cup_j S_j|} \sum_{\mathbf{r} \in \cup_j S_j} (\mathbf{r} - \boldsymbol{\mu})(\mathbf{r} - \boldsymbol{\mu})^T \tag{20.20}$$

where $|A|$ is defined as the number of elements in the set A. Furthermore, let the mean of the jth class calculated from S_j be given by $\boldsymbol{\mu}_j = \frac{1}{|S_j|} \sum_{\mathbf{r} \in S_j} \mathbf{r}$. Then, for each data sample vector $\mathbf{r}$, the MLC denoted by $\delta^{\mathrm{MLC}}(\mathbf{r})$ used for the following experiments is defined by

$$\delta^{\mathrm{MLC}}(\mathbf{r}) = j^* \tag{20.21}$$

that assigns the data sample vector $\mathbf{r}$ to the class j^* with the j^* found by

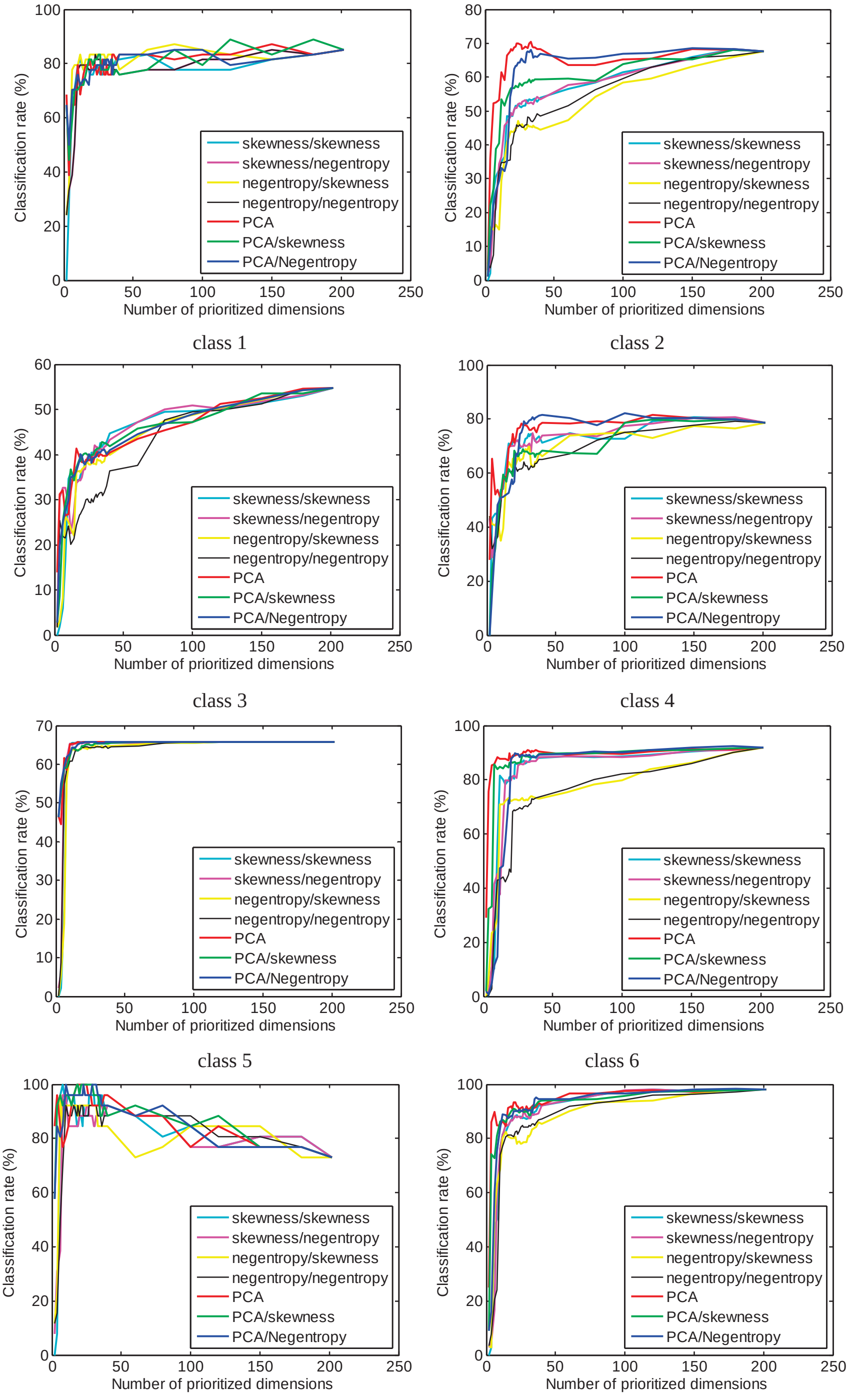
$$j^* = \arg\left\{ \min_j \left(\mathbf{r} - \boldsymbol{\mu}_j\right)^T \mathbf{K}^{-1} \left(\mathbf{r} - \boldsymbol{\mu}_j\right) \right\} \tag{20.22}$$

It should be noted that when MLC is implemented, there are three different ways of calculating the sample covariance matrix $\mathbf{K}$ in (20.22). The simplest one is the global sample covariance matrix that is calculated based on the entire data sample vectors. A second one is to calculate class sample covariance matrices for each of classes by

$$\mathbf{K}_j = \frac{1}{|S_j|} \sum_{\mathbf{r} \in S_j} \left(\mathbf{r} - \boldsymbol{\mu}_j\right)\left(\mathbf{r} - \boldsymbol{\mu}_j\right)^T \tag{20.23}$$

A third one is the one defined in (20.20) that uses the entire training sample covariance matrix. The reason for using (20.20) instead of (20.23) is based on the fact that in some cases $\mathbf{K}_j$ in (20.23) will be ill-ranked when the training set S_j is too small compared to the total number of spectral bands. In general, such a case seldom occurs in multispectral imagery, but it does happen very often when it comes to hyperspectral imagery such as class 1, class 7, class 9, and class 16 in Figure 1.13(d). On the other hand, if the training sample pool, $\cup_j S_j$, is well selected to represent the entire data set, the sample covariance matrix calculated from (20.20) should be very close to the global sample covariance matrix in which case (20.20) can be used for this purpose. So, (20.20) is the best compromise between the global sample covariance matrix and the class sample covariance matrices (20.23).

PSDR-PIPP was implemented to produce PICs. MLC defined by (20.21) and (20.22) was used to classify each class where a 50–50% cross validation was used for performance analysis in which case 50% of data sample vectors in each class was randomly selected to be used for training and the other half was used for testing. Only 16 classes were used for classification since there is no ground truth provided about the background, class 17. The classification rate of each class by MLC was calculated by the number of correct classified data sample vectors in each class divided by the total number of that class according to the ground truth. Figure 20.2 plots the classification rates of 16 classes produced by PSDR-PIPP, where PI/PI in the legends are referred to as the projection index used to produce the PICs/projection index used to prioritize PICs.



class 1

class 2

class 3

class 4

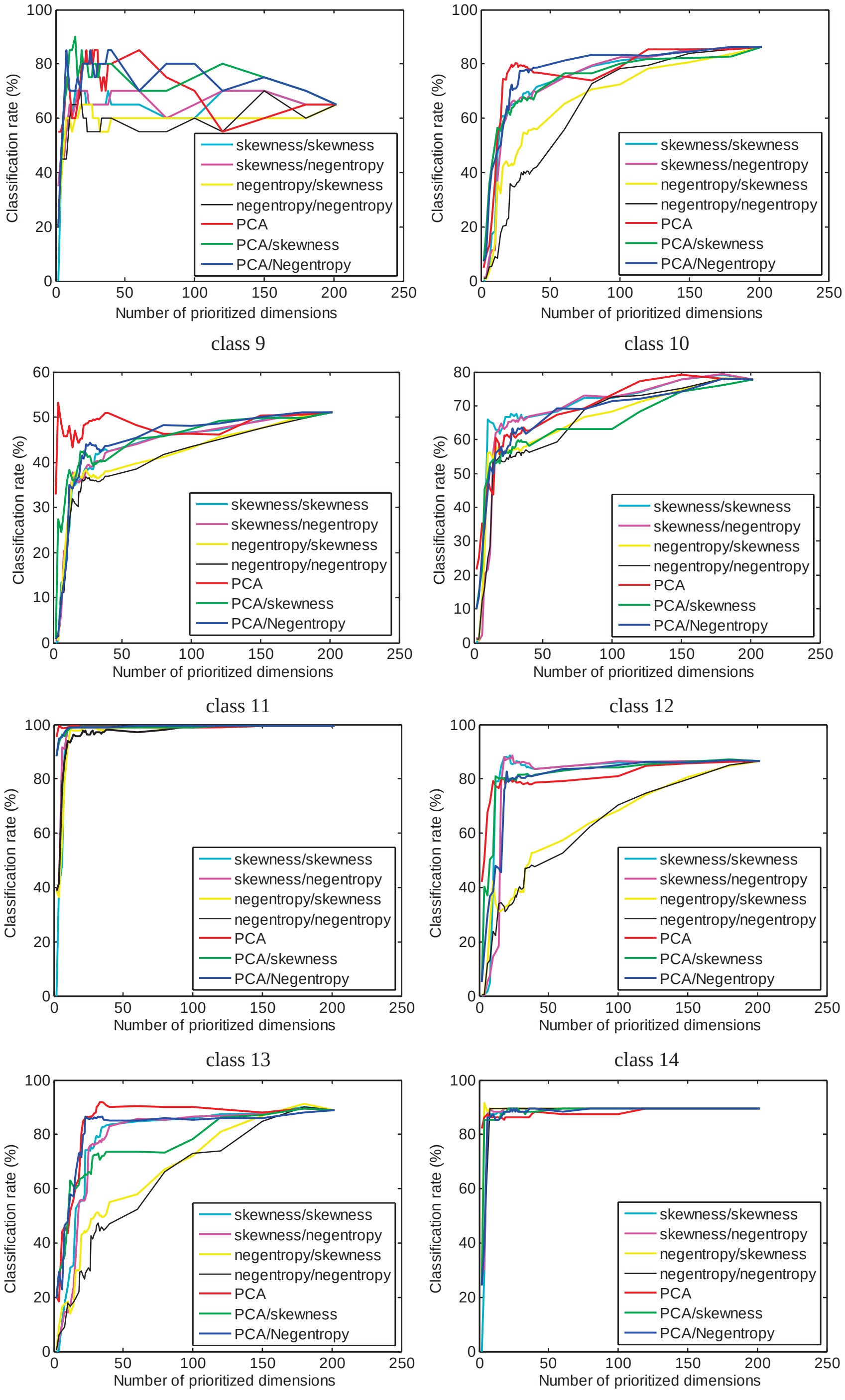
class 5

class 6

class 7

class 8

Figure 20.2 16-class classification rates versus the number of PSDE/PSDR-PIPP-prioritized dimensions used by MLC.



class 9

class 10

class 11

class 12

class 13

class 14

class 15

class 16

Figure 20.2 (*Continued*)

As shown in Figure 20.2, it is very clear that different classes required different numbers of spectral dimensions to achieve their best performance in classification. For example, the easiest cases are class 5 and class 13, where a relatively small number of prioritized dimensions (less than 10) could achieve very high classification rates. To the contrary, the most difficult cases are class 7 and class 9 because both classes only have very few data sample vectors, 26 sample vectors in class 7 and 20 sample vectors in class 9. As a result, the sample size of these two classes is too small to constitute reliable statistics to be used by MLC. This is, in particular, evident when the number of used prioritized dimensions was increased and the performance was actually degraded. Other than these extreme cases, two types of performance are of interest. One is that the classification rates never improved and saturated after a certain number of prioritized dimensions were used, such as classes 2, 5, 6, 8, 13, 14, and 16. Another is that the classification kept improving as the number of used prioritized dimensions was increased, such as classes 3, 11, and 12. There are also many other observations worth discussing based on the results in Figure 20.2 that also allows users to see how PSDE or PSDR performs if prioritized dimensions are progressively added in a forward manner or removed in a backward manner, respectively. These advantages cannot be offered by the traditional DR.

20.6.3 Linear Spectral Mixture Analysis

The land cover/use classification considered in Section 20.6.2 was performed by MLC, which is a hard decision-made classifier, and the Purdue data are, in particular, suitable for its application. However, due to the fact that data sample vectors are heavily mixed, this scene is not applicable to linear spectral unmixing that requires soft decisions in the sense that data sample vectors must be unmixed into a set of abundance fraction maps instead of classification maps produced by MLC. In this case, the 15-panel HYDICE image scene in Figure 1.15(a) was used for experiments where the five panel spectral signatures $\mathbf{p}_i$ for $i = 1, 2, \ldots, 5$ plotted in Figure 1.16 were used for spectral unmixing. In order to perform PSDE/PSDR on this image scene, we first estimated the range of feasible values of dimensionality needed to be retained after DR, q. In doing so, VD was used to estimate the number of spectrally distinct signatures that was $n_{\mathrm{VD}} = 9$ with false-alarm probability P_F greater than or equal to 10^{-4}. So, if we interpret that a spectrally distinct signature can only be specified by one spectral dimension, this implies that it requires at least nine dimensions to accommodate nine spectrally distinct signatures. In other words, the lower bound to the value of q must be 9. According to the ground truth and visual inspection, there are at least nine signatures present in the scene, as shown in Figure 20.3(a). The fully constrained least-squares

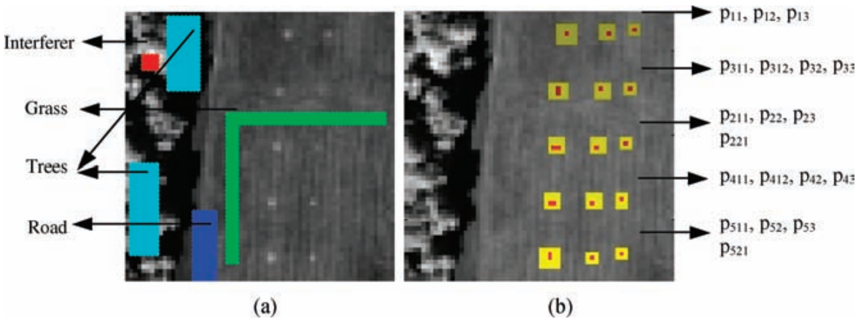

Interferer
Grass
Trees
Road
(a)
p_{11}, p_{12}, p_{13}
p_{311}, p_{312}, p_{32}, p_{33}
p_{211}, p_{22}, p_{23}
p_{221}
p_{411}, p_{412}, p_{42}, p_{43}
p_{511}, p_{52}, p_{53}
p_{521}
(b)

Figure 20.3 (a) A HYDICE panel scene with nine signatures identified by prior knowledge via the ground truth given in (b) which contains 15 panels with ground-truth map of spatial locations of the 15 panels.

(FCLS) method was used to perform spectral unmixing where the signature matrix comprising of the five panel signatures in Figure 1.16 we were obtained from 19 R panel pixels in Figure 20.3(b) and the other four background signatures were obtained by prior knowledge from the areas marked in Figure 20.3(a) as interferer, tree, grass, road. These signatures represented exactly nine signatures estimated by VD. An upper bound to the value of q is then further set by a value twice of VD, that is, $2n_{\mathrm{VD}} = 18$. Therefore, in the following experiments for PSDE/PSDR, we use $q = 9$ as a base to back track to $q = 5$ that represents five panel signatures and to expand forward by extending q to the upper bound, 18.

To simplify our experiments, only one DR technique was used for experiments that were ATGP-FastICA implemented as ID-ICA-DR, where ATGP and FastICA were used as the initialization driven algorithm and DRT, respectively. ATGP-FastICA was then performed by PSDE with $q = 5$ growing to $q = 18$ and by PSDR with $q = 18$ reduced to $q = 5$. Figure 20.4 shows the first 18 independent components (ICs) produced by ATGP-FastICA. As we can see from Figure 20.4, the interferers and the panels in the five rows were already separated by FastICA in its first few prioritized ICs.

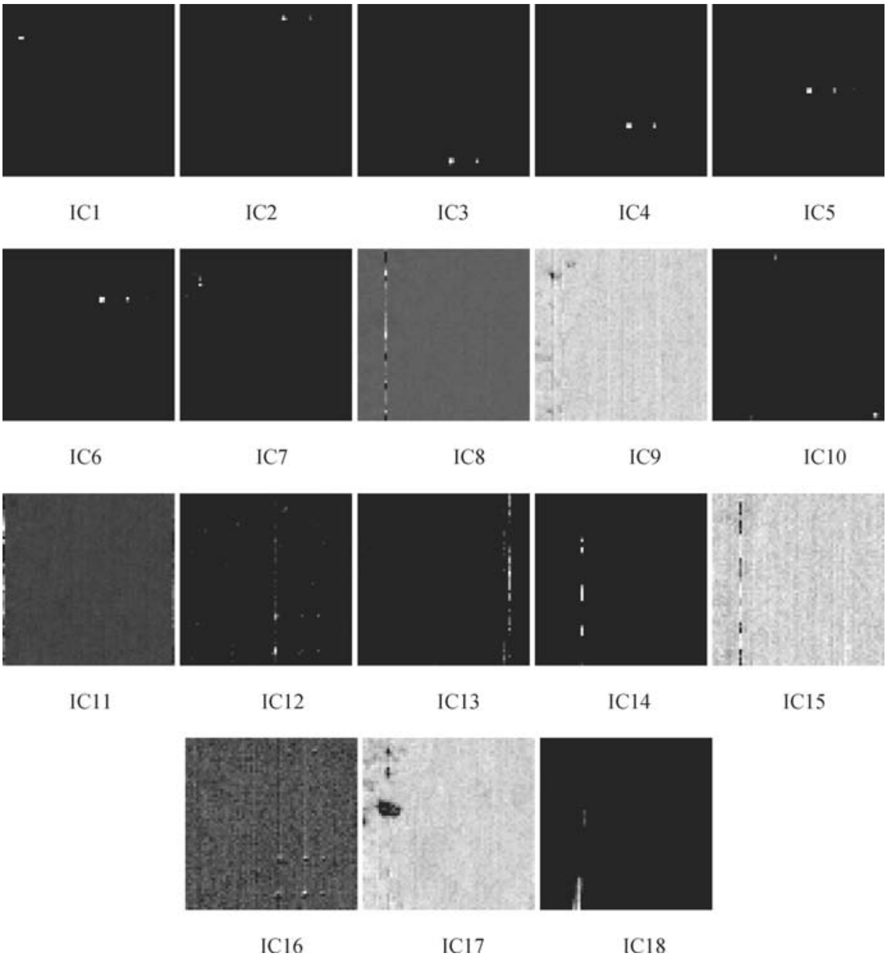
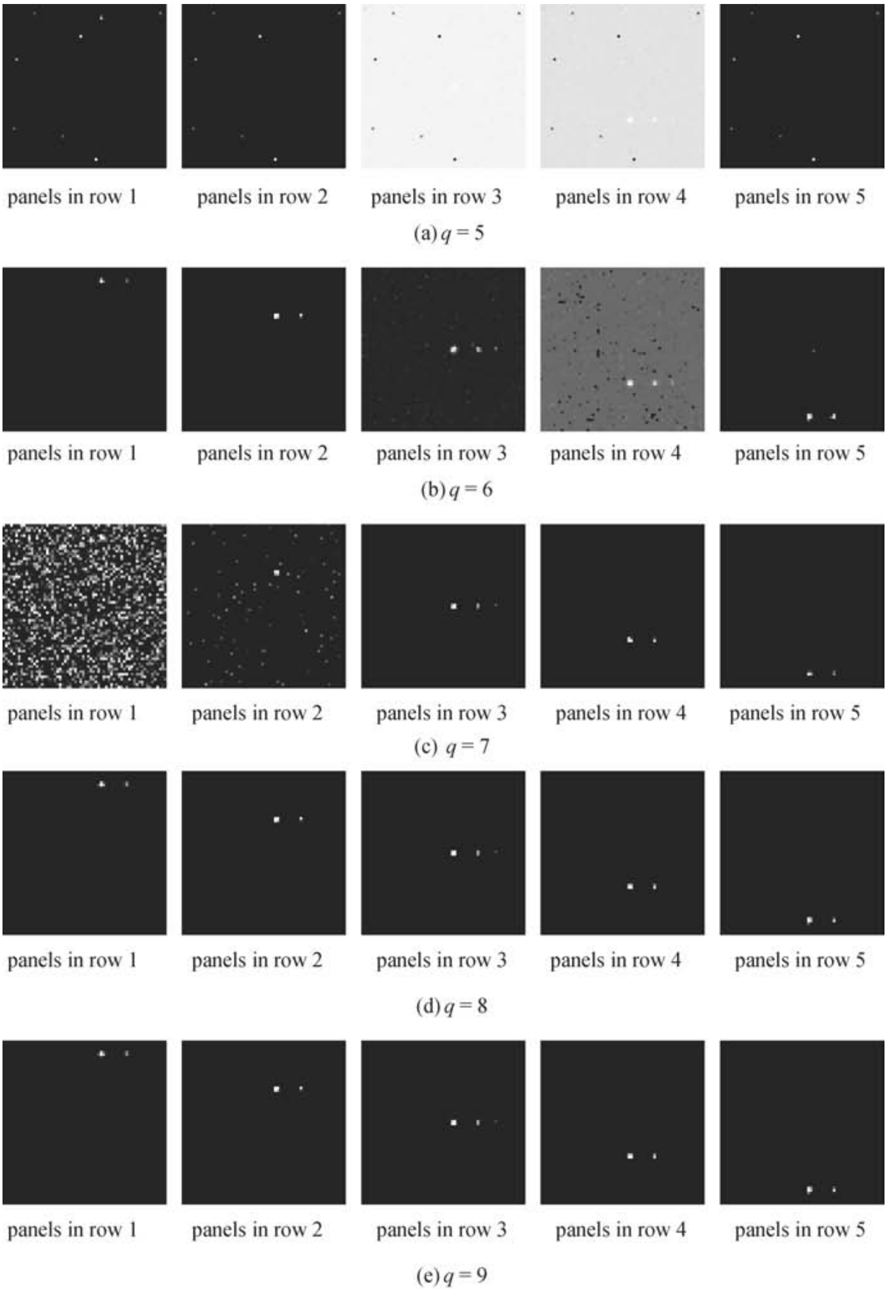


Figure 20.4 First 18 ICs produced by ATGP-FastICA algorithm.

Now, using the prioritized 18 ICs in Figure 20.4, we further used PSDE to expand image cubes with dimensionality from 5 to 18 to be used as data cubes for FCLS to perform spectral unmixing. Figure 20.5(a)–(n) shows the unmixed results produced by the FCLS operating on the image cubes formed by ICs from 5 to 18.



(a) $q = 5$

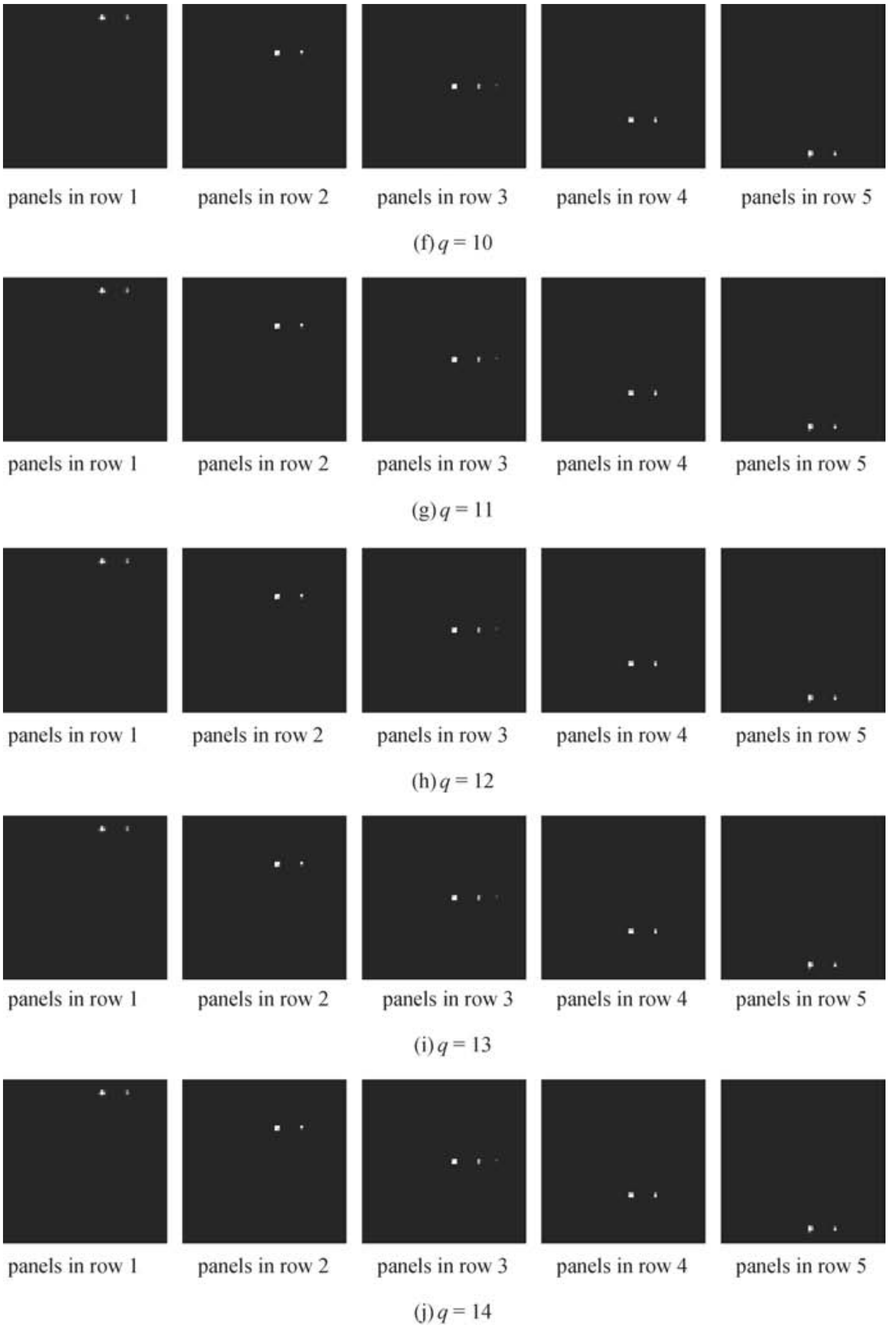
(b) $q = 6$

(c) $q = 7$

(d) $q = 8$

(e) $q = 9$

Figure 20.5 FCLS-classification results using image cubes obtained by PSDR from $q = 5$.



panels in row 1 panels in row 2 panels in row 3 panels in row 4 panels in row 5

(f) $q = 10$

panels in row 1 panels in row 2 panels in row 3 panels in row 4 panels in row 5

(g) $q = 11$

panels in row 1 panels in row 2 panels in row 3 panels in row 4 panels in row 5

(h) $q = 12$

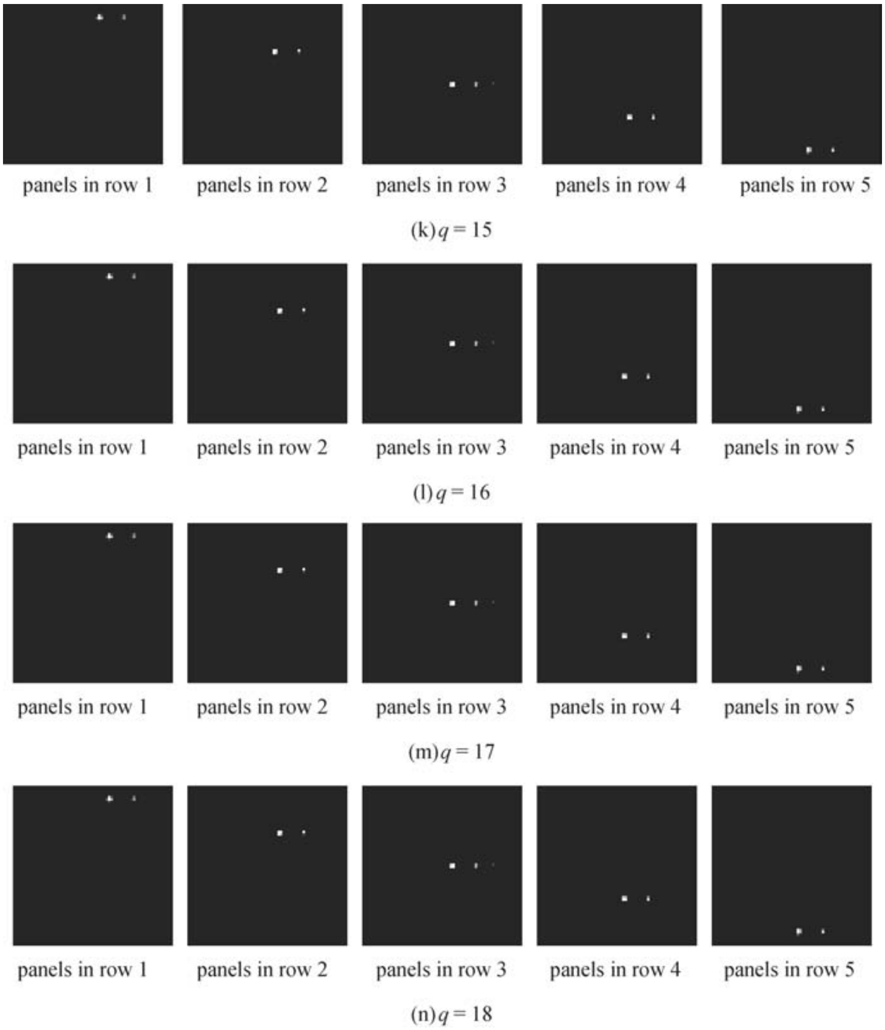
panels in row 1 panels in row 2 panels in row 3 panels in row 4 panels in row 5

(i) $q = 13$

panels in row 1 panels in row 2 panels in row 3 panels in row 4 panels in row 5

(j) $q = 14$

Figure 20.5 (*Continued*)



panels in row 1 panels in row 2 panels in row 3 panels in row 4 panels in row 5

(k) $q = 15$

panels in row 1 panels in row 2 panels in row 3 panels in row 4 panels in row 5

(l) $q = 16$

panels in row 1 panels in row 2 panels in row 3 panels in row 4 panels in row 5

(m) $q = 17$

panels in row 1 panels in row 2 panels in row 3 panels in row 4 panels in row 5

(n) $q = 18$

Figure 20.5 (*Continued*)

For example, Figure 20.5(a) and (n) was the FCLS-unmixed results of the 15 panels in five rows using the image cube formed by the first five ICs and the entire 18 ICs in Figure 20.4, respectively. The results in Figure 20.5(a)–(n) also revealed that $q = 8$ was the least number of spectral dimensions to make FCLS work effectively. When q was less than 8, the FCLS-unmixed results were not good. To the contrary, as q was increased from 8 to 18 FCLS consistently performed well in terms of unmixing the 19 panel pixels and there was no visible difference among all the unmixed results. This suggested that $q = 8$ was the minimal number of spectral dimensions that must be retained after DR and it was very close to the value estimated by VD, $n_{\text{VD}} = 9$. To further confirm this finding, Table 20.1 tabulates the FCLS-unmixed results for $q = 8$ and 9, where the results obtained for $q = 9$ are included in parentheses. As shown in Table 20.1, the unmixed results obtained for $q = 8$ and 9 by FCLS were very close and nearly identical. This implies that there was no additional gain by including more spectral dimensions.

Table 20.1 FCLS quantification for ATGP-FastICA cube with $q = 8$ ($q = 9$)

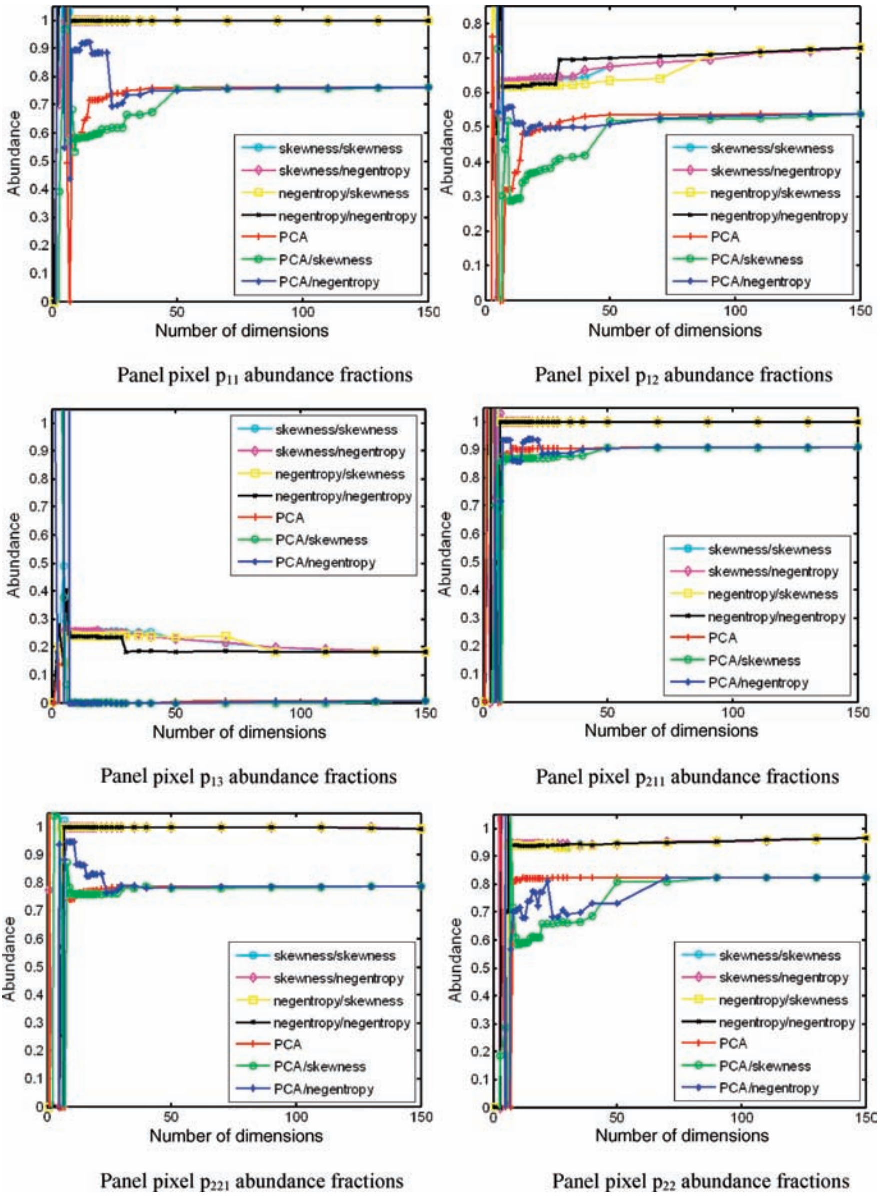
	Panels in row 1	Panels in row 2	Panels in row 3	Panels in row 4	Panels in row 5
p_{11}	1	0	0	0	0
p_{12}	0.6172 (0.6174)	0	0.0382 (0.038)	0	0
p_{13}	0.2413 (2413)	0.0606 (0.0606)	0	0.0432 (0.0432)	0
p_{211}	0	1	0	0	0
p_{212}	0	1	0	0	0
p_{22}	0.0067 (0.071)	0.9432 (0.944)	0	0	0.0058 (0.0068)
p_{23}	0	0.2862 (0.286)	0.0229 (0.0232)	0.0072 (0.0072)	0
p_{311}	0	0.0024 (0.0026)	0.9976 (0.9974)	0	0
p_{312}	0	0	1	0	0
p_{32}	0.0126 (0..013)	0	0.663 (0.6627)	0.0259 (0.0257)	0
p_{33}	0.0045 (0.0047)	0	0.414 (0.4139)	0.0121 (0.0118)	0
p_{411}	0	0	0	1	0
p_{412}	0	0	0	0.9873 (0.987)	0.0127 (0.013)
p_{42}	0.0258 (0.258)	0	0.0149 (0.149)	0.8792 (0.8791)	0
p_{43}	0	0	0.0036 (0.0037)	0.2131 (0.2131)	0
p_{511}	0	0.005 (0.0054)	0	0	0.8617 (0.8621)
p_{512}	0	0	0	0	1
p_{52}	0.0375 (0.0377)	0	0.0388 (0.387)	0.022 (0.0219)	0.9008 (0.9012)
p_{53}	0	0.0256 (0.0255)	0	0	0.1605 (0.1603)

Figure 20.6 illustrates the abundance fractions of 19 R pixels unmixed by FCLS via PSDE/PSDR using a pair of PIs, indicated by PI/PI where the first PI was used to generate projection vectors using ATGP as an ID-PIPP and the second PI was used for PIC prioritization.

In order to conduct a quantitative study for unmixed results in Figure 20.6, the 3D-ROC analysis developed in Chapter 3 was used for the study. Figure 20.7 only plots the area, A_z, calculated under 2D-ROC curves of P_D versus P_F produced by 3D ROC curves. This results confirmed that the cut-off value for the minimal number of prioritized spectral dimensions, q, to be used for spectral unmixing predicted by $n_{\mathrm{VD}} = 9$ was very accurate if not exact.

20.7 Conclusions

This chapter introduces a new concept of DP that has never been explored in the literature in the past. Its idea arises from recognition of several issues in implementing DR. One is that the number of data dimensions required to be retained, q, after DR must be known in advance. In a case that the value of q is inaccurate, the entire process of DR must be reimplemented again for a new value of q. This leads to an issue, "Can the data dimensions previously obtained by a smaller value of q be used without re-running DR for a new larger value of q?" In addition, once DR is performed, how can these DR-transformed dimensions be represented in terms of information contained in these new spectral dimensions? Despite that PCA resolves the above issues by solving eigenvalues from the characteristic polynomial equation, it is unfortunate that the same approach cannot be extended or generalized to any linear transformation. For example, ICA does not have such nice properties. The PIPP and DP presented in this chapter provide a feasible solution to resolving this dilemma where PI plays a twofold role in producing projection vectors and prioritizing projection vector-generated PICs. However, it should be noted that the used PIs for both cases are not necessarily the same and can be different. In other words, when PIPP is implemented in conjunction with DP, a pair of two different PIs must be used, one for projection vector generation and the other for PIC prioritization. This is quite different from PCA that uses data variance as the same PI for both finding eigenvectors as projection vectors and using eigenvalues for PC prioritization. One of major advantages and benefits provided by PIPP via DP is no requirement of prior knowledge about the value of q, which is the number of spectral dimensions needed to be retained for DR. It allows users to perform PSDP with two dual processes, PSDE via DP and PSDR via DP which can expand and reduce spectral dimensions in a forward and a backward manner progressively, and one is a reverse process of another. By means of these two processes the hyperspectral information



Panel pixel p_{11} abundance fractions

Panel pixel p_{12} abundance fractions

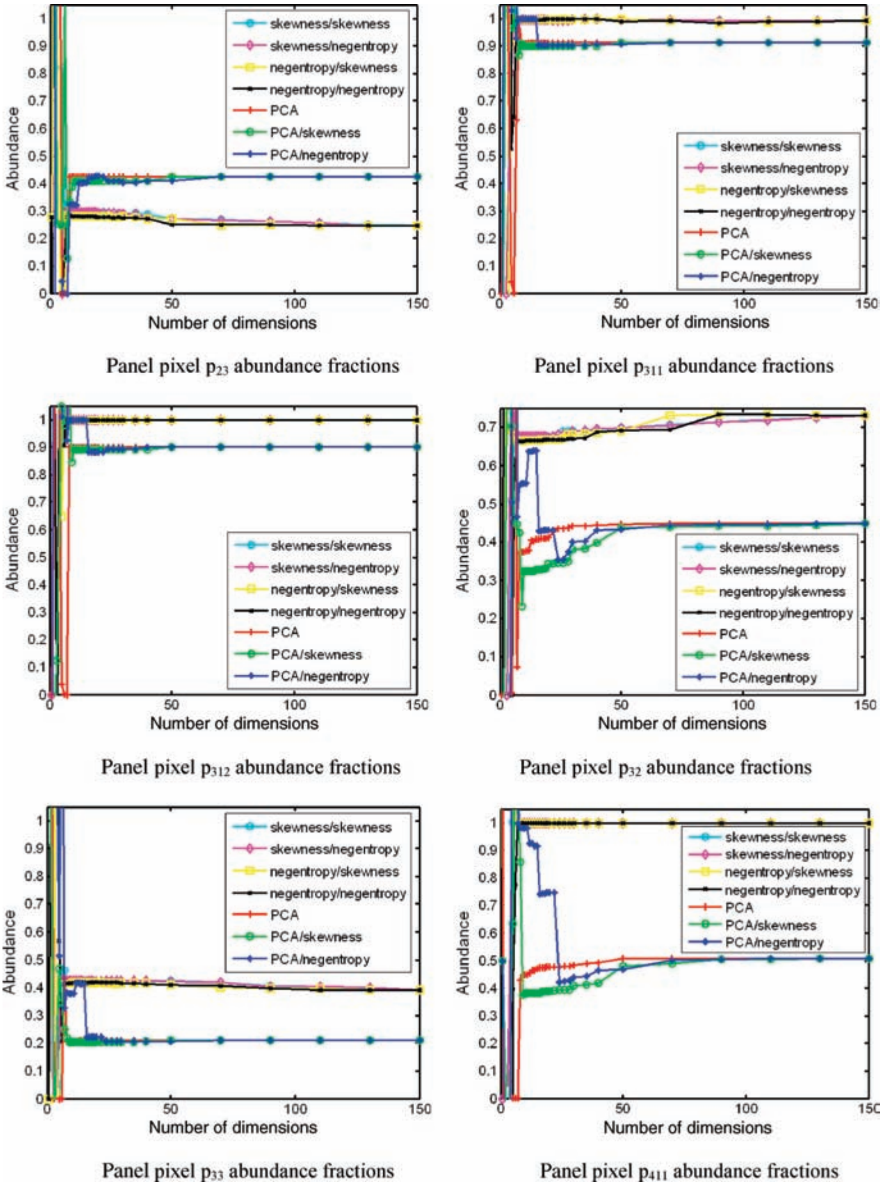
Panel pixel p_{13} abundance fractions

Panel pixel p_{211} abundance fractions

Panel pixel p_{221} abundance fractions

Panel pixel p_{22} abundance fractions

Figure 20.6 FCLS-unmixed abundance fractions of 19 R panel pixels using ATGP-PIPP.



Panel pixel p_{23} abundance fractions

Panel pixel p_{311} abundance fractions

Panel pixel p_{312} abundance fractions

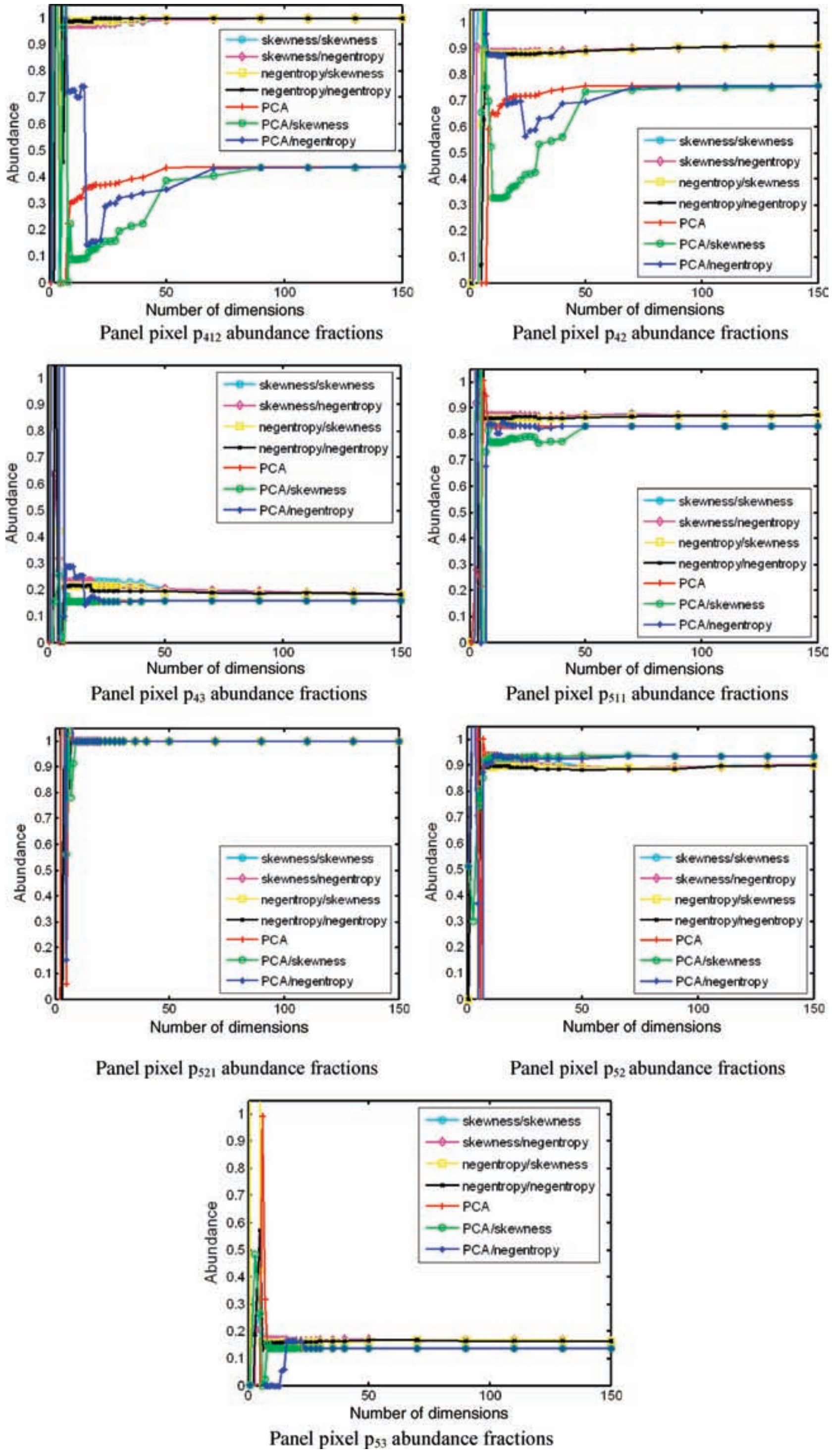
Panel pixel p_{32} abundance fractions

Panel pixel p_{33} abundance fractions

Panel pixel p_{411} abundance fractions

Figure 20.6 (*Continued*)

compression can be best utilized in many applications, such as data compression, communication, and transmission. Technically, speaking, PSDP can start off any value of q, for example, $q = 1$ for PSDE and $q = L$ for PSDR. However, this may not be practical in applications. Therefore, as shown in Wang and Chang (2006a), VD may not be exactly accurate but is a good estimate for q. By taking advantage of VD a feasible range of $[n_{\mathrm{VD}}, 2n_{\mathrm{VD}}]$ can be derived for PSDP to implement PSDE and PSDR as demonstrated in our experiments as well as in some recent works, Safavi (2010), Fisher and Chang (2011), Chang and Safavi (2011), and Chang et al. (2011). Nevertheless, if n_{VD} is too large the range of $[n_{\mathrm{VD}}, 2n_{\mathrm{VD}}]$ may be too broad. To further address this issue, another



Panel pixel p_{412} abundance fractions

Panel pixel p_{42} abundance fractions

Panel pixel p_{43} abundance fractions

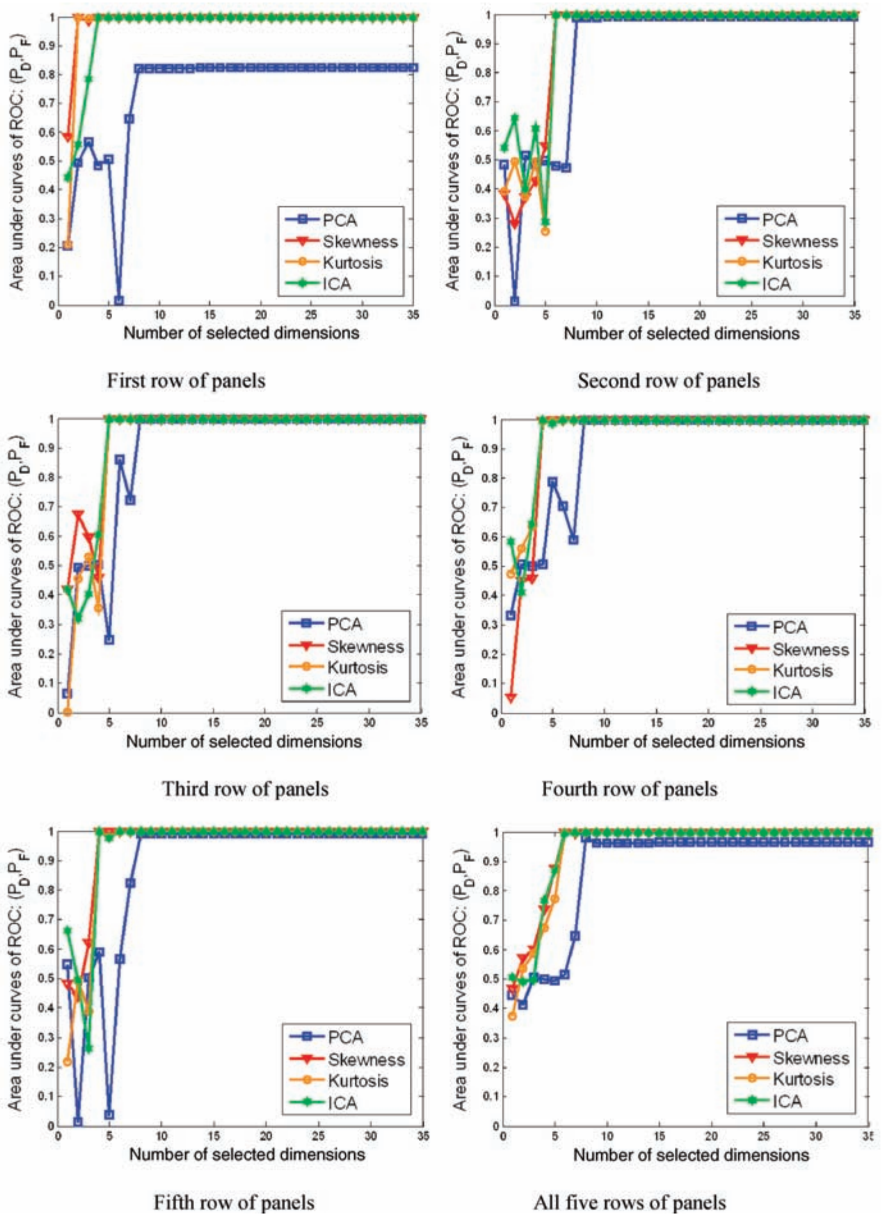
Panel pixel p_{511} abundance fractions

Panel pixel p_{521} abundance fractions

Panel pixel p_{52} abundance fractions

Panel pixel p_{53} abundance fractions

Figure 20.6 (*Continued*)



First row of panels

Second row of panels

Third row of panels

Fourth row of panels

Fifth row of panels

All five rows of panels

Figure 20.7 Plots of areas under 2D ROC curves for averaged classification rate of 19 R panel pixels in Figure 20.6 versus number of prioritized dimensions.

new concept of DDA introduced in Chapter 22 can also be applied to narrow down the range by replacing $2n_{\mathrm{VD}}$ with DDA. This approach was studied by Safavi (2010), where experiments have shown that since n_{VD} already provides a good estimate for q, the reduction of the upper bound from $2n_{\mathrm{VD}}$ to DDA does not have many advantages. But, this is not the case for BS, as will be demonstrated in Chapter 23 as well as Liu (2011).

21

Progressive Band Dimensionality Process

Using DR as a means to perform progressive spectral dimensionality process (PSDP) by dimensionality prioritization (DP) has been studied in great detail in Chapter 20. It is natural to include Chapter 21 as a companion chapter of Chapter 20 with its main theme focused on band dimensionality for progressive process. In other words, what has been done for PSDP via DP can also be applied to progressive band dimensionality process (PBDP) via band prioritization (BP). However, there are a few significant differences between PSDP and PBDP that make the techniques developed for PSDP in Chapter 20 not necessarily applicable to PBDP. One is how data information is represented by DP and BP. In PSDP the data information is compacted via a transformation from the original data space to a new transformed data space compared to PBDP that only retains the information of selected bands but completely discards the information of unselected bands. So, PSDP requires the complete hyperspectral image cubes to perform transformation as opposed to PBDP that only processes the hyperspectral image cubes formed by selected bands. Another major difference is that PSDP requires a pair of parameters, projection vectors to specify transformed components in the transformed data space and priority scores to prioritize the information contained in the transformed components, whereas PBDP does not require projection vectors but only needs priority scores to prioritize all the spectral bands for BS since each spectral band can be considered as a projection vector. A third key difference is design rationales for PSDP and PBDP. While the projection index-based projection pursuit (PIPP) is a key to make PSDP work, the pigeon-hole principle described in Section 1.3.2 of Chapter 1 is the main idea behind PBDP. In spite of these differences both PSDP and PBDP share the same design philosophy in common that intends to resolve the same issue of how data information is represented and prioritized in a lower data representation space so that the data information can be utilized more effectively and efficiently in various tasks for data analysis. A general issue arising in both DR and BS is determining the number of data dimensions required to be retained by data dimensionality reduction, q, and the number of bands to be selected by band selection, $\tilde{q}$. It has been shown that the virtual dimensionality (VD) in Chapter 5 can be used to estimate these two numbers, q and $\tilde{q}$. Nevertheless, it is worth noting that VD can only serve as an estimate and may not be necessarily accurate. For this reason PSDP and PBDP offer an exit strategy by processing data in progression without actually knowing the values of q and $\tilde{q}$.

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

21.1 Introduction

Hyperspectral images are collected in hundreds of contiguous spectral channels. Therefore, not only the data volume to be processed is considered to be huge, but also the spectral correlation among bands is expected to be very high due to high spectral resolution. Band selection (BS) is one of commonly used approaches that take advantage of such high interband correlation to remove band redundancy. Over the past years, many research efforts have been directed to BS (Mausel et al., 1990; Conese and Maselli, 1993; Stearns et al., 1993; Chang et al., 1999; Huang and He, 2005; Chang and Wang, 2006; Du et al., 2007) in order to achieve a wide range of applications such as data compression, data storage, data transmission, and communication. Generally, two crucial issues arising in BS need to be resolved, which are (1) number of bands required for BS and (2) what criterion to be used to select bands. Instead of dealing with BS directly this chapter introduces a new concept of band prioritization (BP) that is similar to the DP developed in Chapter 20 to simultaneously address these two issues by prioritizing all spectral bands in some optimal sense. It ranks all the spectral bands in accordance with their corresponding priority scores that can be calculated by a custom-designed BP criterion in a similar way that spectral dimensions are prioritized by DP in Chapter 20. Then PBDP can be performed by BP in such a manner that two dual processes, progressive band dimensionality reduction (PBDR) via BP and progressive band dimensionality expansion (PBDE) via BP, similar to progressive spectral dimensionality reduction (PSDR) via DP and progressive spectral dimensionality expansion (PSDE) via DP can also be derived. In other words, rather than determining the number of bands needed to be selected $\tilde{q}$ as required by BS, PBDP allows users to terminate its process either in band reduction or in band expansion by applications. Although the value of $\tilde{q}$ can be estimated by VD, a true value of $\tilde{q}$ is never known in real-world problems. In this case, a better way to circumvent this problem is to let applications determine when the process should be terminated. Nevertheless, VD has been shown to provide a reasonable estimate of a lower bound on $\tilde{q}$ in Chang and Wang (2006). This value can be used to suggest an initial guess for $\tilde{q}$ to initialize PBDP. So, a next key issue in PBDP is to design an effective criterion to meet a specific application. In Chang et al. (1999) BS was performed by prioritizing bands according to a specific criterion and followed by band de-correlation (BD) to remove bands that are highly correlated with those bands already selected until a specific number of bands $\tilde{q}$ is achieved. Recently, Du et al. (2007) made use of a similar idea to perform BS by repeatedly calculating the information divergence among all the unselected bands and selecting and removing the one with the maximal divergence from the set of unselected bands to form a new set of unselected bands for the next round BS. The process is continued until it reaches a specific number of bands, $\tilde{q}$. The proposed PBDP is fundamentally different from this type of the conventional BS and revolutionizes the concept of BS in several aspects. The first and foremost is that PBDP does not need to find and fix the value of $\tilde{q}$. Instead, the $\tilde{q}$ can be tuned by a specific application or image analysts. Secondly, it prioritizes the entire set of L spectral bands according to their contained information measured by a custom-designed information criterion and then selects bands progressively in a forward or backward manner depending on how to retain band information in increasing or decreasing order. To realize this concept two dual processes derived from PBDP are further developed. One is referred to as PBDR via BP that performs PBDP in a backward manner by beginning with a higher number of highest-prioritized bands and gradually removes bands according to their increasing priority order from the selected prioritized bands. As a complete opposite to PBDR, the second process is referred to as PBDE via BP that performs PBDP in a forward manner by starting with a lower number of highest-prioritized bands and gradually adding new unselected bands according to their priorities in descending order. Thirdly, both processes can be terminated when a stopping rule is satisfied and is determined by various applications. In this case, VD can be used to provide a reasonable lower bound on $\tilde{q}$ for PBDR and an upper bound on $\tilde{q}$ for PBDE. This was not done in Chang et al. (1999) and nor was in Du et al. (2007). Such a progressive nature significantly reduces computational complexity because the previously selected bands are always a part of future augmented band subsets without repeatedly solving new combinatorial problems.

Unlike PSDP that takes advantage of the projection index-based projection pursuit (PIPP) to preserve information provided by projection index components, PBDP must retain information provided by selected bands in a more effective fashion. Since each spectral band is acquired by a specific wavelength designed to extract certain material substances present in the range, the selection of bands must be determined by substances of interest in applications. As noted in Section 1.3.2, the pigeon-hole principle offers a means of interpreting each of spectral bands as a pigeon hole and substance of interest needed to be analyzed as pigeons. By virtue of this pigeon-hole principle a spectral band can be used to specify one particular material substance that indeed determines which band should be used for its accommodation. Accordingly, the effectiveness of PBDP should be therefore determined by material substances of interest in applications. Due to this fact VD which is also derived from the pigeon-hole principle can be used to estimate the value of $\tilde{q}$ selected for PBDP which should be very closely related to the value estimated by VD that is indeed the case shown in Chang and Wang (2006).

21.2 Band Prioritization

The concept of BP was implicitly used by Chang et al. (1999) in conjunction with information divergence-based BD to perform BS. Its potential in various applications has not been realized since then. This section revisits BP and extends it to PBDP while postponing progressive BS (PBS) to Chapter 23, which can be considered as an application of PSDP to BS.

For each spectral band $\mathbf{B}_l$ BP prioritizes $\mathbf{B}_l$ according to its contained information measured by a custom-designed information criterion. PBDP selects bands progressively by expanding or reducing band dimensionality. Such a progressive band dimensionality process can be terminated by a stopping rule that is determined by various applications instead of a specific value of p that must be selected by BS in advance. Nevertheless, in this case, VD can be used to provide a reasonable lower bound and a reliable upper bound on the value of the $\tilde{q}$. In addition to the second-order statistics proposed in Chang et al. (1999) this chapter extends criteria to high-order statistics. Specifically, four categories of criteria are considered in this chapter for BP. The first category is made up of second-order statistics-based criteria, which include second-order statistics-based variance and signal-to-noise ratio (SNR). The second category contains higher-order statistics-based criteria including skewness, kurtosis, entropy, and information divergence (ID). The third category consists of classification-based criteria, which characterize data features by Fisher's linear discriminant analysis (FLDA) and orthogonal subspace projection (OSP). The fourth category comprises detection-based criteria, band correlation/dependence minimization, and band correlation/dependence constraint recently developed in Chang and Wang (2006) that characterize target signatures derived from the concepts of linearly constrained minimum variance (LCMV) (Frost, 1972) and constrained energy minimization (CEM) (Harsanyi, 1993) in Chapter 2. These BP criteria can be used to measure and calculate the priority scores to rank all spectral bands. PBDP is finally performed by adding or removing bands progressively according to their associated priority scores in descending order until a desired performance is achieved. In other words, BP allows PBDP to add or remove bands in a progressive manner via their descending priority scores. But, it should be pointed out that PBDP is different from the joint BP/BD-based BS proposed by Chang et al. (1999) because the former does not use any technique to remove interband redundancy as the latter did via an ID criterion. Since the BD is an independent process of PBDP, it can be always incorporated in PBDP to derive PBDP/BD that implements PBDP in conjunction with BD to remove highly spectral correlated bands. The resulting process is called progressive band selection (PBS) that will be further discussed in Chapter 23.

There are several significant differences between BS and PBDP. One major advantage of PBDP over BS is that the former allows users to select bands in accordance with their associated priorities, while the latter requires users to first determine how many bands needed to be selected and then to find an optimal band set among all possible combinations from the original band set. As a result, if the total number of bands and the number of bands to be selected are assumed to be L and $\tilde{q}$, respectively, an exhaustive search for finding an optimal band subset conducted by BS requires solving $\begin{pmatrix} L \\ \tilde{q} \end{pmatrix} = \frac{L!}{(L-\tilde{q})!\tilde{q}!}$ optimization problems. This becomes a formidable task for hyperspectral image data. In order to mitigate this dilemma Chang et al. (1999) developed a joint band prioritization and band de-correlation (BP/BD) approach by designing an appropriate criterion to measure band information coupled with an ID-based BD to remove interband redundancy. The information measured by BP within a single band is considered as intraband information that is quite different from interband information measured by BD that can be used to remove the interband redundancy. Both BP and BD are generally considered as disjoint and different techniques for information processing. That is why both BP and BD are proposed to address this issue in Chang et al. (1999) where BP was used to avoid an exhaustive search for finding an optimal band subset while BD was used to remove bands which are highly correlated with those bands being selected. The benefit of incorporating BD in PBDP can be only gained, provided that the criterion used for BD is effective. On many cases, the unknown subtle targets may appear in one particular band that is highly correlated with other bands in terms of one BD criterion such as second order of statistics, but may not have much correlation with the same bands using another BD criterion such as high-order statistics. Under this circumstance, a better approach is to leave BD an option if no particular application is specified. Such a BD issue will be investigated in Chapter 23 when we come to discuss PBS.

Another major advantage of using PBDP over the conventional BS is that no optimization problem needs to be solved and bands can be selected up to any number at users' discretion. A third major advantage is the order of bands to be selected. When BS is performed in a conventional manner, bands are selected by solving an objective function in some sense of an optimal criterion where there is no concept of BP involved. However, when PBDP is used, bands can be selected either by their highest-priority scores or by their least-priority scores depending upon applications, a task that cannot be accomplished by the conventional BS. It has been a common sense that high interband correlation makes people believe that many highly correlated bands may provide overlapped information and can be removed. However, if the targets of interest are rare and relatively small such as anomalies, these targets are very likely captured by nonoverlapping information among highly correlated bands. Moreover, the limited presence of these targets can be characterized more effectively by bands with lowest-priority scores rather than higher-priority scores due to their small spatial extent. A good example is PCA where all the components are ranked by data variances with principal components (PCs) and minor components (MCs) corresponding to two respective sets of eigenvalues, a set of larger eigenvalues and a set of smaller eigenvalues. It has been shown in Oja (1992) that MCs can also be as important as PCs. In hyperspectral image analysis, targets with weak spectral sample correlation in terms of eigenvalues such as anomalies or endmembers generally do not contribute to large spectral data variances. As a consequence, such targets generally cannot be captured by PCs, but rather by MCs. This indicates that for BS to be effective including bands with lower-priority scores is crucial to capture such weak targets. As demonstrated in experiments in this chapter, a comparative study and analysis between using bands with both highest- and least-priority scores shows that using bands selected by lower-priority scores could sometimes perform even better than using bands with higher-priority scores in finding these targets. Additionally, the experiments also provide evidence that the results using the higher- and lower-prioritized bands are quite different in most cases. On some cases, combining both bands selected by highest and lowest priorities may provide better results.

Finally, despite that two dual processes analogous to the two dual processes, PSDR and PSDE developed for PSDP, referred to as PBDR via BP and PBDE via BP, can also be derived for PBDP, it should be noted that unlike DP, which is mainly developed for preprocessing, the effectiveness of BP must be justified by its utility in hyperspectral data exploitation. So, applications of PBDP play a major part in evaluating performance of PBDP.

21.3 Criteria for Band Prioritization

In order to implement BP, a criterion is needed to measure the significance of a spectral band in terms of its priority score. In what follows, four classes of criteria are considered.

21.3.1 Second-Order Statistics-Based BPC

The first category of BP criterion (BPC) is derived from second-order statistics that are based on variance and signal-to-noise ratio (SNR).

21.3.1.1 Variance-Based BPC

A natural and logical approach to band prioritization is to compute variances for all spectral band images $\{\mathbf{B}_l\}_{l=1}^{L}$ in a hyperspectral image cube, denoted by $\{\sigma_l^2\}_{l=1}^{L}$ and further use band variances to define a priority score for each of band images as follows:

$$\mathrm{VAR}_{\mathrm{priority}}(\mathbf{B}_l) = \sigma_l^2 \text{ for all } l = 1, 2, \ldots, L \tag{21.1}$$

Another alternative interpretation is to use a set of loading factors $\{\xi_{kl}\}_{k=1,l=1}^{L,L}$ proposed in Tu et al. (1998) and Chang et al. (1999) that can be defined by

$$\xi_{kl} = \sqrt{\lambda_k} v_{kl} \text{ for } k, l = 1, 2, \ldots, L \tag{21.2}$$

It is easy to show that for each $l = 1, 2, \ldots, L$, ρ_l defined by

$$\rho_l^{\mathrm{PCA}} = \sum\nolimits_{k=1}^{L} \xi_{kl}^2 \tag{21.3}$$

turns out to be the variance σ_l^2 of the lth spectral band image. As a result of (21.3), the priority score calculated by (21.1) for the lth spectral band image $\mathbf{B}_l$ is also equivalent to the PCA-based priority score defined by

$$\mathrm{PCA}_{\mathrm{priority}}(\mathbf{B}_l) = \rho_l^{\mathrm{PCA}} \tag{21.4}$$

21.3.1.2 Signal-to-Noise-Ratio-Based BPC

It was noted in Green et al. (1988) that variance was not an appropriate criterion to measure image quality. In order to alleviate this dilemma, an SNR-based criterion was first developed by Green et al. (1988) to improve the PCA. The resulting transform was called maximum noise fraction transform and later reinterpreted by Lee et al. (1990) as noise-adjusted principal component (NAPC) transform. In analogy with the criterion specified by (21.4) for the variance-based PCA, a similar criterion to (21.4) can also be derived from the SNR-based NAPC as follows.

Assume that $\{\lambda_{adj,l}\}_{l=1}^{L}$ is the set of eigenvalues of noise-adjusted sample covariance matrix and $\{\mathbf{v}_{adj,l}\}_{l=1}^{L}$ are their associated orthonormal eigenvectors. We can define the loading factors in a similar manner to (21.2) for an NAPC by

$$\xi_{adj,kl} = \sqrt{\lambda_{adj,k}}v_{kl} \text{ for } k,l = 1,2,\ldots,L \tag{21.5}$$

Using (21.5), a noise-adjusted variance-based priority score can be calculated for the lth spectral band image $\mathbf{B}_l$ via (21.5) defined by the NAPC-based priority score:

$$\mathrm{NAPC}_{\mathrm{priority}}(\mathbf{B}_l) = \sum\nolimits_{k=1}^{L} \xi_{\mathrm{adj},kl}^2 = \rho_l^{\mathrm{NAPC}} \tag{21.6}$$

21.3.2 High-Order Statistics-Based BPC

In many applications, the information of interest may not be captured by second-order statistics, but rather be characterized by higher-order statistics. In order to take this into account, three higher-order statistics-based criteria are derived in this section for BPC.

21.3.2.1 Skewness

The simplest high-order statistics is the third central moment, referred to as skewness and defined by

$$k_3(\mathbf{B}_l) = (1/N)\sum\nolimits_{i=1}^{N} [(r_{il} - \mu_l)/\sigma_l]^3 = \rho_l^{\mathrm{skewness}} \tag{21.7}$$

21.3.2.2 Kurtosis

A fourth central moment, referred to as kurtosis, is defined by

$$k_4(\mathbf{B}_l) = (1/N)\sum\nolimits_{k=1}^{N} [(r_{lk} - \mu_l)/\sigma_l]^4 = \rho_l^{\mathrm{kurtosis}} \tag{21.8}$$

21.3.3 Infinite-Order Statistics-Based BPC

It should be noted that according to our experience, criteria for BP based on statistics higher than 4 do not have much significant advantage compared to skewness and kurtosis (Ren et al., 2006). Therefore, only ∞-order statistics-based BP criteria, entropy and information divergence, are discussed in this section.

21.3.3.1 Entropy

One of the simplest and most widely used ∞-order statistic-based BPC is entropy that requires an infinite number of moments. Let $H(\mathbf{B}_l)$ be the entropy calculated for the lth band image $\mathbf{B}_l$. The entropy-based priority score for $\mathbf{B}_l$ is defined by

$$\text{Entropy}_{\text{priority}}(\mathbf{B}_l) = H(\mathbf{B}_l) = \rho_l^{\text{entropy}} \tag{21.9}$$

It should be noted that the entropy $H(\mathbf{B}_l.)$ in (21.9) is calculated based on the gray-level histogram produced by the lth spectral band image $\mathbf{B}_l$ where the number of bins is totally determined by the difference between the maximal and minimal gray levels present in the spectral band image $\mathbf{B}_l$. For example, if the maximal and minimal gray levels are 255 and 0, respectively, then there are together 256 bins needed to estimate the entropy.

21.3.3.2 Information Divergence

As an alternative to entropy defined by (21.9), an information theoretic measure, called information divergence (ID), can also be used as a BPC. Assume that the $\mathbf{p}_l$ is the image histogram of the lth spectral band image, $\mathbf{B}_l$ normalized as a probability distribution and $\mathbf{g}_l$ is its associated Gaussian distribution with mean and variance determined by sample mean and sample variance of the $\mathbf{B}_l$. BP criterion of interest is to measure the deviation far away from a Gaussian distribution for a given spectral band image, that is, the discrepancy between $\mathbf{p}_l$ and $\mathbf{g}_l$ defined by

$$\text{ID}_{\text{priority}}(\mathbf{B}_l) = \text{D}(\mathbf{p}_l; \mathbf{g}_l) = \rho_l^{\text{ID}} \tag{21.10}$$

where $\text{D}(\mathbf{p}_l;\mathbf{g}_l)$ is called information divergence (Kullback, 1968)

$$\text{D}(\mathbf{p}_l; \mathbf{g}_l) = \sum\nolimits_i p_{li}\log(p_{li}/g_{li}) + \sum\nolimits_i g_{li}\log(g_{li}/p_{li}) \tag{21.11}$$

The higher the value of $\text{D}(21.\mathbf{p}_l;\mathbf{g}_l)$ in (21.11), the greater deviation of $\mathbf{p}_l$ from the Gaussian distribution, $\mathbf{g}_l$ is. This implies that ID is used to measure the degree of non-Gaussianity of a band. It should be noted that if both $\mathbf{p}_l$ and $\mathbf{g}_l$ are replaced with two spectral signatures, the $\text{D}(\mathbf{p}_l;\mathbf{g}_l)$ defined by (21.11) becomes spectral information divergence (SID) in Chang (2000) and Chang (2003a).

21.3.4 Classification-Based BPC

In the previous two subsections, BPC are designed based on statistics. Additionally, they are also unsupervised in the sense that no prior knowledge is involved in these criteria. However, in some applications, prior knowledge may be available and can be taken advantage of to design BPC. In this subsection, two supervised classification-based criteria are developed for BP. Such classification-based BP criteria are different from statistics-based BP criteria such as variance-, SNR- or high-order statistics-based BP criteria in the sense that the former is developed for target detection and classification applications while the latter is completely determined by statistics that has little to do applications.

21.3.4.1 Fisher's Linear Discriminant Analysis (FLDA)-Based BPC

Minimum misclassification canonical analysis (MMCA) derived from Fisher's linear discriminant analysis (FLDA) was used in Tu et al. (1998) to minimize the misclassification error. For any given band number $\tilde{q}$, we can use (21.2)–(21.3) with eigenvalues and unit eigenvectors replaced by the eigenvalues $\{\lambda_j\}_{j=1}^{\tilde{q}-1}$ and normalized unit feature vectors $\{\mathbf{w}_j^{\text{FLDA}}\}_{j=1}^{\tilde{q}-1}\mathbf{w}_j^{\text{FLDA}} = \left(w_{j1}^{\text{FLDA}}, w_{j2}^{\text{FLDA}}, \ldots, w_{jL}^{\text{FLDA}}\right)^T$ as used in Tu et al. (1998) to define the loading factors as follows:

$$\xi_{jl}^{\text{FLDA}} = \sqrt{\lambda_j} w_{jl}^{\text{FLDA}} \tag{21.12}$$

for $j = 1, 2, \ldots, \tilde{q} - 1$ and $l = 1, 2, \ldots, L$ In light of (21.12), the priority score can be calculated for the lth spectral band image $\mathbf{B}_l$ by

$$\rho_l^{\text{FLDA}} = \sum\nolimits_{j=1}^{\tilde{q}-1} \left(\xi_{jl}^{\text{FLDA}}\right)^2 \tag{21.13}$$

21.3.4.2 OSP-Based BPC

Another classification-based criterion is derived from the orthogonal subspace projection (OSP) (Harsanyi and Chang, 1994) that is based on the linear mixture model as follows:

$$\mathbf{r} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n} \tag{21.14}$$

where $\mathbf{M} = \left[\mathbf{m}_1\mathbf{m}_2 \ldots \mathbf{m}_p\right]$, $\boldsymbol{\alpha} = \left(\alpha_1, \alpha_2, \ldots, \alpha_p\right)^T$ and $\mathbf{n}$ is noise or model error. If we further assume that the p image endmembers $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ can be divided into two classes of endmembers, one class of $n_{\mathbf{D}}$ desired image endmembers denoted by $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{n_{\mathbf{D}}}$ and the other class of undesired endmembers denoted by $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{n_{\mathbf{U}}}$ with $p = n_{\mathbf{D}} + n_{\mathbf{U}}$. Then the OSP classifier for a particular desired endmember $\mathbf{m}_j$, $\mathbf{w}_j^{\text{OSP}}$ can be actually obtained by $\mathbf{w}_j^{\text{OSP}} = \mathbf{m}_j^T P_{\mathbf{U}_j}^{\perp}$ with $\mathbf{w}_j^{\text{OSP}} = \left(w_{j1}^{\text{OSP}}, w_{j2}^{\text{OSP}}, \ldots, w_{jL}^{\text{OSP}}\right)^T$ where $\mathbf{U}_j = \left[\mathbf{m}_1 \cdots \mathbf{m}_{j-1}\mathbf{m}_{j+1} \cdots \mathbf{m}_{n_{\mathbf{D}}}\mathbf{u}_1\mathbf{u}_2 \cdots \mathbf{u}_{n_{\mathbf{U}}}\right]$ and $P_{\mathbf{U}_j}^{\perp} = \mathbf{I} - \mathbf{U}_j\mathbf{U}_j^{\#}$, $\mathbf{U}_j^{\#}$ is the pseudo-inverse of $\mathbf{U}_j$. Now following the same argument outlined by (21.12) and (21.13) we can define loading factors for the OSP classifiers $\{\mathbf{w}_j^{\text{OSP}}\}_{j=1}^{p}$ as

$$\xi_{jl}^{\text{OSP}} = \sqrt{\bar{\lambda}_j} w_{jl}^{\text{OSP}} \quad \text{for } j = 1, 2, \ldots, n_{\mathbf{D}} \text{ and } l = 1, 2, \ldots, L \tag{21.15}$$

and

$$\rho_l^{\text{OSP}} = \sum\nolimits_{j=1}^{n_{\mathbf{D}}} \left(\xi_{jl}^{\text{OSP}}\right)^2 \tag{21.16}$$

where $\bar{\lambda}_j = (1/N)\sum_{i=1}^{N} \lambda_j(\mathbf{r}_i)$ and $\lambda_j(\mathbf{r}_i) = \left(\alpha_j(\mathbf{r}_i)/\sigma\right)^2\left(\mathbf{m}_j^T P_{\mathbf{U}_j}^{\perp}\mathbf{m}_j\right)^{-1}$ obtained in Chang et al. (1999). By means of (21.16), the priority score assigned to the lth spectral band image $\mathbf{B}_l$ can be calculated by

$$\text{OSP}_{\text{priority}}(\mathbf{B}_l) = \rho_l^{\text{OSP}} \tag{21.17}$$

21.3.5 Constrained Band Correlation/Dependence Minimization

Taking a rather different approach from the ideas used to design previous BP criteria, a recent new approach, called constrained band selection (CBS) developed in Chang and Wang (2006), suggested a new criterion for BP, which linearly constrained a particular band image while minimizing band correlation/dependence resulting from other band images. In other words, the priority score of a spectral band can be calculated according to the degree of correlation or dependence between this particular band image and other band images measured by least squares errors. Its idea can be briefly described as follows.

Assume that the size of all the spectral band images $\mathbf{B}_l$ is $n_r \times n_c$. Since each spectral band image $\mathbf{B}_l$ can be represented by a column vector of dimensions $n_r n_c$, denoted by $\mathbf{b}_l$, we have a total number of L spectral band image vectors $\{\mathbf{b}_l\}_{l=1}^{L}$. For any given spectral band image vector $\mathbf{b}_l$ we can design a finite impulse response (FIR) specified by a set of L weighting vectors, $\{\mathbf{w}_l\}_{l=1}^{L}$ that constrains $\mathbf{b}_l$ while minimizing least squares error caused by other band image vectors $\{\mathbf{b}_k\}_{k=1,k\neq l}^{L}$. More specifically, let y_l be the filter output obtained by

$$y_l = \mathbf{w}_l^T \mathbf{b}_l \tag{21.18}$$

The averaged least squares filter output is given by

$$(1/L)\sum_{l=1}^{L} y_l^2 = (1/L)\sum_{l=1}^{L} \left(\mathbf{w}_l^T \mathbf{b}_l\right)\left(\mathbf{w}_l^T \mathbf{b}_l\right)^T = \mathbf{w}_l^T \left((1/L)\sum_{l=1}^{L} \mathbf{b}_l \mathbf{b}_l^T\right)\mathbf{w}_l \tag{21.19}$$

Let $\mathbf{Q} = (1/L)\sum_{l=1}^{L} \mathbf{b}_l \mathbf{b}_l^T$ denote the band image correlation matrix. A similar optimization problem to the constrained energy minimization (CEM) in Chapter 2 can be obtained as follows:

$$\min_{\mathbf{w}_l}\left\{\mathbf{w}_l^T \mathbf{Q} \mathbf{w}_l\right\} \text{ subject to } \mathbf{b}_l^T \mathbf{w}_l = 1 \tag{21.20}$$

The solution to (21.20), denoted by $\mathbf{w}_l^{\text{CEM}}$ is given by

$$\mathbf{w}_l^{\text{CEM}} = \left(\mathbf{b}_l^T \mathbf{Q}^{-1} \mathbf{b}_l\right)^{-1} \mathbf{Q}^{-1} \mathbf{b}_l \tag{21.21}$$

Alternatively, we can exclude the spectral band image $\mathbf{b}_l$ from the band correlation matrix $\mathbf{Q}$ and further define $\tilde{\mathbf{Q}} = (1/(L-1))\sum_{j=1,j\neq l}^{L} \mathbf{b}_j \mathbf{b}_j^T$ as the band dependence matrix. Replacing $\mathbf{Q}$ in (21.20) with $\tilde{\mathbf{Q}}$ results in a similar constrained band selection problem

$$\min_{\mathbf{w}_l}\left\{\mathbf{w}_l^T \tilde{\mathbf{Q}} \mathbf{w}_l\right\} \text{ subject to } \mathbf{b}_l^T \mathbf{w}_l = 1 \tag{21.22}$$

The solution to (21.22), $\tilde{\mathbf{w}}_l^{\text{CEM}}$ is the same as the one in (21.21) with the $\mathbf{Q}$ replaced by $\tilde{\mathbf{Q}}$ that is given by

$$\tilde{\mathbf{w}}_l^{\text{CEM}} = \left(\mathbf{b}_l^T \tilde{\mathbf{Q}}^{-1} \mathbf{b}_l\right)^{-1} \tilde{\mathbf{Q}}^{-1} \mathbf{b}_l \tag{21.23}$$

21.3.5.1 Band Correlation/Dependence Minimization

By means of (21.21) and (21.23) we can calculate the following least squares errors (LSEs):

$$\rho_l^{\text{CEM}} = \left(\mathbf{w}_l^{\text{CEM}}\right)^T \mathbf{Q} \mathbf{w}_l^{\text{CEM}} \tag{21.24}$$

$$\tilde{\rho}_l^{\text{CEM}} = \left(\tilde{\mathbf{w}}_l^{\text{CEM}}\right)^T \tilde{\mathbf{Q}} \tilde{\mathbf{w}}_l^{\text{CEM}} \tag{21.25}$$

that can be used to measure degree of the spectral band image vector $\mathbf{b}_l$ correlated with and dependent on other spectral band image vectors $\{\mathbf{b}_k\}_{k=1,k\neq l}^{L}$, respectively. That is, the greater the LSE in (21.24) or (21.25), the higher the correlation of $\mathbf{b}_l$ with other band image vectors. So, we can use (21.24) and (21.25) to derive two criteria for BP, called band correlation minimization (BCM) defined by

$$\mathrm{BCM}_{\mathrm{priority}}(\mathbf{B}_l) = \rho_l^{\mathrm{CEM}} \tag{21.26}$$

and band dependence minimization (BDM) defined by

$$\mathrm{BDM}_{\mathrm{priority}}(\mathbf{B}_l) = \tilde{\rho}_l^{\mathrm{CEM}} \tag{21.27}$$

21.3.5.2 Band Correlation Constraint

Comparing to (21.24) and (21.25), an alternative approach is to calculate band correlation constraint (BCC)

$$\eta_l^{\mathrm{CEM}} = \sum\nolimits_{k=1,k\neq l}^{L} \left(\mathbf{w}_l^{\mathrm{CEM}}\right)^T \mathbf{b}_k \tag{21.28}$$

and band dependence constraint (BDC)

$$\tilde{\eta}_l^{\mathrm{CEM}} = \sum\nolimits_{k=1,k\neq l}^{L} \left(\tilde{\mathbf{w}}_l^{\mathrm{CEM}}\right)^T \mathbf{b}_k \tag{21.29}$$

which can also be used to measure the correlation between the spectral band image vector $\mathbf{b}_l$ and any other spectral band image vector $\mathbf{b}_k$ $(21.k \neq l, k = 1, \ldots, L)$. By comparing the value of $\mathbf{w}_l^{\mathrm{CEM}}\mathbf{b}_k$ with the filter constraint specified by $\mathbf{w}_l^T\mathbf{b}_l = 1$ in (21.20) or (21.22), a spectral band image $\mathbf{B}_k$ has less correlation with the spectral band image $\mathbf{B}_l$ if its band constraint $\mathbf{w}_l^T\mathbf{b}_k$ is far away from 1. In other words, the closer the $\mathbf{w}_l^T\mathbf{b}_k$ to 1, the higher the correlation of $\mathbf{B}_k$ to $\mathbf{B}_l$. With this interpretation, two criteria similar to (21.26) and (21.27) can also be derived for BP, called BCC given by

$$\mathrm{BCC}_{\mathrm{priority}}(\mathbf{B}_l) = \eta_l^{\mathrm{CEM}} \tag{21.30}$$

band dependence constraint (BDC)

$$\mathrm{BDC}_{\mathrm{priority}}(\mathbf{B}_l) = \tilde{\eta}_l^{\mathrm{CEM}} \tag{21.31}$$

One disadvantage of these CEM-based criteria is the enormous size of vectors converted from band images that causes tremendous computing time. For example, it requires a vector with 4×10^4 dimensions to represent a band image with size 200×200. In order to mitigate this dilemma, a linearly constrained minimum variance (LCMV) in (Frost, 1972; Van Veen and Buckley, 1988) is developed to derive four criteria similar to four CEM-based criteria specified by (21.24) and (21.25) and (21.28) and (21.29). Instead of constraining a band image as a vector, the LCMV-CBS constrains a band image as an image matrix without vector conversion. Its idea is derived from the LCMV approach, which can be traced back to Frost's work in adaptive beamforming (Frost, 1972). More specifically, assume that $\mathbf{r}_{l,1}, \mathbf{r}_{l,2}, \ \ldots, \ \mathbf{r}_{l,n_c}$ are n_c columns of the lth spectral band image $\mathbf{B}_l$, which has n_r rows and n_c columns. So, the jth column vector of $\mathbf{B}_l$ denoted by $\mathbf{r}_{l,j}$ is represented by an n_r-dimensional column vector, $\mathbf{r}_{l,j} = \left(r_{l,1j}, r_{l,2j}, \ldots, r_{l,n_rj}\right)^T$. In this case, the lth spectral band image $\mathbf{B}_l$ can be further expressed by a matrix given by

$$\mathbf{B}_l = \begin{bmatrix} r_{l,11} & r_{l,12} & \cdots & r_{l,1(n_c-1)} & r_{l,1n_c} \\ r_{l,21} & r_{l,22} & \ddots & r_{l,2(n_c-1)} & r_{l,2n_c} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ r_{l,(n_r-1)1} & \ddots & \ddots & r_{l,(n_r-1)(n_c-1)} & r_{l,(n_r-1)n_c} \\ r_{l,n_r1} & r_{l,n_r2} & \cdots & r_{l,n_r(n_c-1)} & r_{l,n_rn_c} \end{bmatrix} = \left[\mathbf{r}_{l,1}\mathbf{r}_{l,2}\cdots\mathbf{r}_{l,n_c}\right] \tag{21.32}$$

Like the CEM, the goal is to design a constrained FIR linear filter with an n_r-dimensional weight column vector $\mathbf{v}_l = \left(v_{l,1}, v_{l,2}, \ldots, v_{l,n_r}\right)^T$ specified by a set of n_r filter coefficients $\left\{v_{l,1}, v_{l,2}, \ldots, v_{l,n_r}\right\}$ that minimizes (21.19) subject to the following simultaneous n_c multiple constraints, $\mathbf{r}_{l,j}^T\mathbf{v}_l = \sum_{m=1}^{n_r} r_{l,mj}v_{l,m} = 1$, $1 \le j \le n_c$ that is equivalent to

$$\mathbf{B}_l^T\mathbf{v}_l = \mathbf{1}_{n_c} \tag{21.33}$$

where $\mathbf{1}_{n_c}$ is an n_c-dimensional column vector with all 1s in its n_c components. It should be noted that since the weight vector $\mathbf{v}_l$ is used to constrain column vector of a band image, its dimensionality is n_r compared to the n_rn_c-dimensional weight vector $\mathbf{w}_l$ used in (21.20) that constrains a band image as a vector with dimensionality n_rn_c. By virtue of the n_c multiple constraints in (21.33), the CEM problem described by (21.20) can be rederived as the following LCMV-based optimization problem:

$$\min_{\mathbf{v}_l}\left\{\mathbf{v}_l^T\mathbf{\Sigma}\mathbf{v}_l\right\} \text{ subject to } \mathbf{B}_l^T\mathbf{v}_l = \mathbf{1}_{n_c} \tag{21.34}$$

where $\mathbf{\Sigma} = (1/L)\sum_{l=1}^{L}\mathbf{B}_l\mathbf{B}_l^T$ is the sample band correlation matrix. The solution to (21.34) can be solved as

$$\mathbf{v}_l^{\text{LCMV}} = \mathbf{\Sigma}^{-1}\mathbf{B}_l\left(\mathbf{B}_l^T\mathbf{\Sigma}^{-1}\mathbf{B}_l\right)^{-1}\mathbf{1}_{n_c} \tag{21.35}$$

and

$$\rho_l^{\text{LCMV}} = \left(\mathbf{v}_l^{\text{LCMV}}\right)^T\mathbf{\Sigma}\mathbf{v}_l^{\text{LCMV}} \ \text{ for LCMV-BCM} \tag{21.36}$$

plays the same role that ρ_l does for the CEM-BCM. Similar derivations to CEM-BDM can also be obtained for $\tilde{\mathbf{v}}_l^{\text{LCMV}} = \tilde{\mathbf{\Sigma}}^{-1}\mathbf{B}_l\left(\mathbf{B}_l^T\tilde{\mathbf{\Sigma}}^{-1}\mathbf{B}_l\right)^{-1}\mathbf{1}_{n_c}$ and

$$\tilde{\rho}_l^{\text{LCMV}} = \left(\tilde{\mathbf{v}}_l^{\text{LCMV}}\right)^T\tilde{\mathbf{\Sigma}}^{-1}\tilde{\mathbf{v}}_l^{\text{LCMV}} \ \text{ for LCMV-BDM} \tag{21.37}$$

Table 21.1 Comparison among various BP criteria

	Statistics	Supervised	Complexity	Constrained
Variance	Second	No	Low	No
SNR	Second	No	Low	No
Skewness	Third	No	Low	No
Kurtosis	Fourth	No	Low	No
Entropy	∞	No	Low	No
ID	∞	No	Low	No
FLDA	Second	Yes	Low	No
OSP	Second	Yes	High	No
BCM/BDM	Second	No	Low	Yes
BCC/BDC	Second	No	Low	Yes

In analogy with the CEM-based band correlation/dependence constraint criteria (BCC/BDC)

$$\zeta_l = \sum\nolimits_{k=1,k\neq l}^{L} \mathbf{1}_{n_c}^T\left(\mathbf{B}_k^T\left(\mathbf{v}_l^{\mathrm{LCMV}}\right)\right) \quad \text{for LCMV-BCC} \tag{21.38}$$

and

$$\tilde{\zeta}_l = \sum\nolimits_{k=1,k\neq l}^{L} \mathbf{1}_{n_c}^T\left(\mathbf{B}_k^T\left(\tilde{\mathbf{v}}_l^{\mathrm{LCMV}}\right)\right) \quad \text{for LCMV-BDC} \tag{21.39}$$

can also be derived for an LCMV-based band correlation/dependence constraint criteria by replacing band image vector $\mathbf{b}_l$ and CEM-based weight vectors with band image $\mathbf{B}_l$ and LCMV-based weight vectors, respectively.

Despite the fact that the CBS described was developed in Chang and Wang (2006), the idea of BP was not introduced in their paper and nor were the priority scores specified by (21.26) and (21.27), (21.30)–(21.31). Table 21.1 summarizes all the proposed BPC in terms of their characteristics where “supervised” indicates that training samples are required for the particular criterion.

As a concluding remark, one comment is noteworthy. The effectiveness of BP is determined by its applications not criteria alone. As will be demonstrated by following experiments, a different application yields a totally different selected set of bands. With proper bands selected by BP the number of bands can be significantly reduced, while still achieving performance comparable to that accomplished by using full bands.

21.4 Experiments for BP

As noted in the introduction, there are some important differences between BP and DP. One is that BP prioritizes individual spectral bands based on their contained information, whereas DP prioritizes spectral dimensions according to the information contained in their transformed components from the entire image data. Therefore, spectral bands only share information provided by interband correlation compared to spectral components that only retain information of residuals resulting from all their previous spectral components. Accordingly, BP and DP have different utilities in applications. This section presents applications of BP using different sets of prioritized spectral bands in unsupervised spectral unmixing and endmember extraction, which are not applicable to DP. The HYDICE image scene in Figure 1.15(a) and (b) was selected for experiments to allow us to perform a quantitative analysis in performance of unmixing panel pixels and extracting panel pixels as endmembers.

21.4.1 Applications Using Highest-Prioritized Bands

When it comes to BP, a natural and intuitive approach is to select bands that have the highest-priority scores. Table 21.2 tabulates the first 30 bands with highest-priority scores selected progressively by various BP criteria developed in Section 21.3 with a backslash "/" used to separate two selected bands. It is very clear to see from the table that if one spectral band is selected with a high-priority score, so are its neighboring spectral bands due to their strong interband correlation.

With each BPC, the 30 bands in Table 21.2 are separated by three categories, first highest-prioritized 10 bands, second highest-prioritized 10 bands, and third highest-prioritized 10 bands to evaluate the effectiveness of each BP criterion in performance analysis by adding the next 10 highest-prioritized bands at a time progressively until reaching a total number of 30 bands. It is also noted that the uniform band selection is not included in Table 21.2 since it does not prioritize bands and will not be evaluated along with BPC in the following applications for fair comparison.

Table 21.2 30 highest-prioritized spectral bands selected progressively by various 10 BP criteria

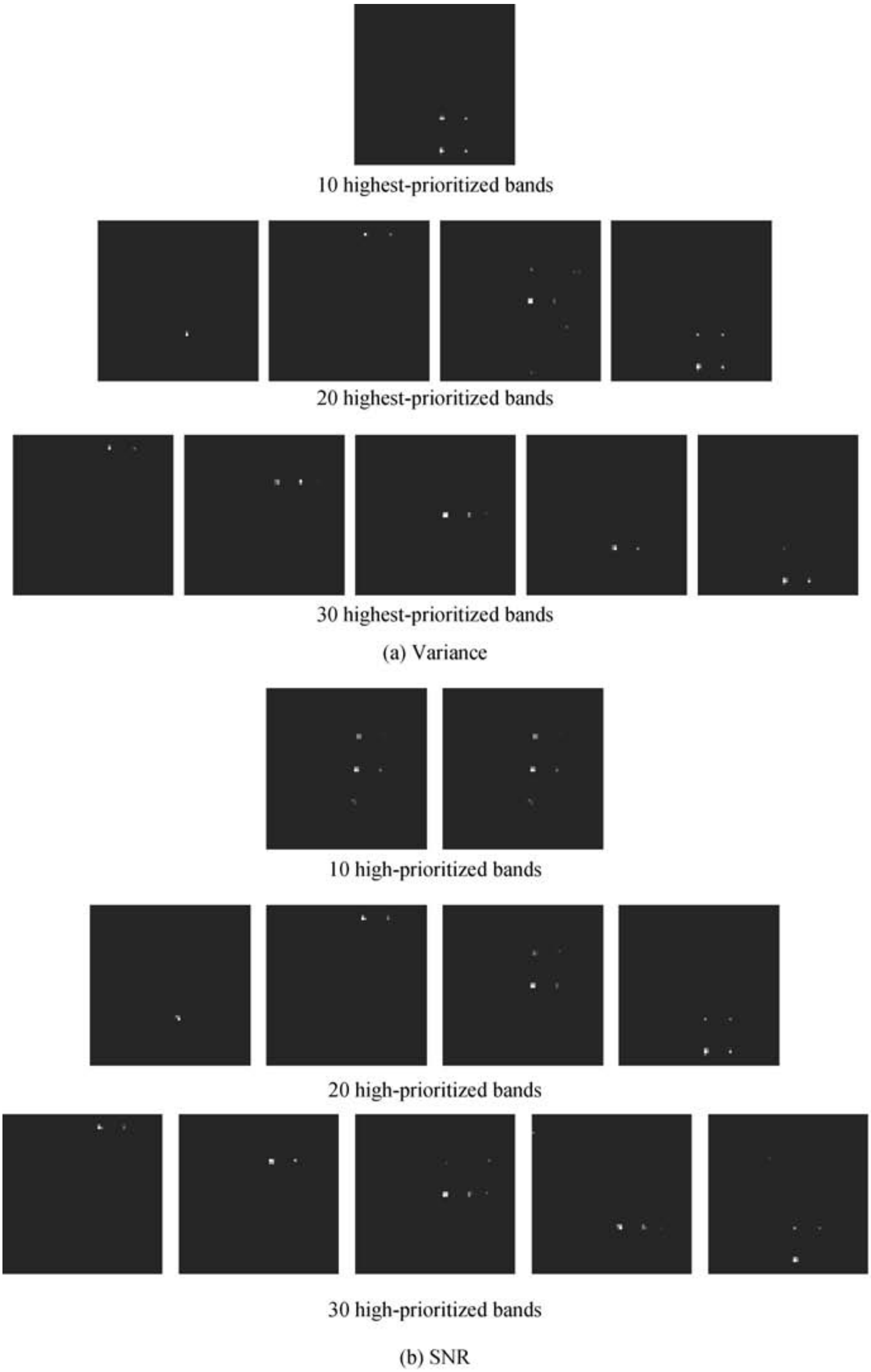
	30 highest-priority scores
Variance	(First 10 bands) 60/61/67/66/65/59/57/68/62/64
	(Second 10 bands) 56/78/77/76/79/63/53/80/58/75
	(Third 10 bands) 52/81/55/69/54/49/50/82/48/51
SNR	(First 10 bands) 78/80/93/91/92/95/89/94/90/88
	(Second 10 bands) 102/96/79/82/105/62/107/108/104/109
	(Third 10 bands) 101/110/103/63/106/77/61/111/70/112
Skewness	(First 10 bands) 1/122/123/124/125/126/50/49/127/2
	(Second 10 bands) 48/169/168/47/51/128/46/129/45/167
	(Third 10 bands) 130/22/23/166/21/44/18/20/131/24
Kurtosis	(First 10 bands) 1/59/60/61/62/64/63/65/66/58
	(Second 10 bands) 67/68/69/57/70/80/82/81/72/79
	(Third 10 bands) 78/77/76/75/71/74/83/56/84/85
Entropy	(First 10 bands) 65/60/67/53/66/61/52/68/59/64
	(Second 10 bands) 62/78/77/57/79/49/76/56/50/80/
	(Third 10 bands) 48/63/51/47/75/45/58/81/46/54
ID	(First 10 bands) 154/157/156/153/150/158/145/164/163/160
	(Second 10 bands) 142/144/148/143/141/152/155/135/146/159
	(Third 10 bands) 166/149/138/139/167/161/165/147/151/137
FLDA	(First 10 bands) 56/81/55/21/35/42/4/52/53/12
	(Second 10 bands) 2/74/8/57/11/5/1/16/19/54
	(Third 10 bands) 26/24/51/27/10/20/3/49/34/13
OSP	(First 10 bands) 60/57/59/61/56/66/65/67/62/64
	(Second 10 bands) 68/58/63/77/78/76/79/80/69/55
	(Third 10 bands) 81/75/82/70/74/54/83/71/92/93
BCM/BDM	(First 10 bands) 125/169/168/164/146/124/165/128/167/135
	(Second 10 bands) 122/160/159/161/162/163/126/166/129/151
	(Third 10 bands) 123/155/156/49/130/158/157/154/50/127
BCC/BDC	(First 10 bands) 125/169/124/164/167/161/163/168/146/129
	(Second 10 bands) 123/155/151/166/158/130/156/135/152/153
	(Third 10 bands) 133/139/150/134/141/144/147/50/148/127

21.4.1.1 Unsupervised Linear Spectral Mixture Analysis

The spectral unmixing technique used in this section was the unsupervised fully constrained least squares (UFCLS) in Chapter 8 to unmix data sample vectors into a set of abundance fractions specified by signatures that were used to form a linear mixing model. Table 21.2 tabulates those bands selected by 10 BP criteria with $\tilde{q}$ indicating the number of bands to be selected by a BPC. There were three cases considered, that is, $\tilde{q} = 10$, 20, and 30. We assume that no prior knowledge was provided about the image scene. Therefore, the unmixing was performed in an unsupervised fashion. In this case, an unsupervised algorithm, automatic target generation process (ATGP) in Chapter 8, was implemented to generate the desired signatures directly from the image scene to be used to form the signature matrix for supervised spectral unmixing. It should be noted that since a $\tilde{q}$ bands-formed image cube has only $\tilde{q}$ dimensions that could be used for orthogonality projection, only $\tilde{q}$ target pixels could be generated by ATGP. Using these $\tilde{q}$ ATGP-generated target pixels to form a desired signature matrix $\mathbf{M}$, a fully constrained least squares (FCLS) classification method (Heinz and Chang, 2001; Chang, 2003a) was then used to perform spectral unmixing. The selection of FCLS over other abundance-unconstrained or partially constrained unmixing methods such as OSP was due to the fact that the knowledge provided by the $\tilde{q}$ generated target pixels was pixel information and usually sensitive to spectral unmixing. In this case, fully constrained methods were more appropriate to reduce sensitivity. Figure 21.1(a)–(j) shows the UFCLS-mixed pixel abundance fractional maps of the 19 panel pixels resulting from $\tilde{q} = 10$, 20, and 30 bands in Table 21.2 prioritized by the 10 BP criteria where ATGP was used to generate target information being unsupervised.

As shown in Figure 21.1, when $\tilde{q} = 10$, the best UFCLS-mixed results were those produced by the ID, BCM/BDM and BCC/BDC where 19 panel pixels were classified into four separate abundance fractional maps with panel pixels in rows 2 and 3 unmixed into one single abundance fractional map. As $\tilde{q}$ was increased to 20, only the unmixed results produced by BCM/BDM and skewness were able to correctly unmix all the 19 panel pixels into five separate abundance fractional maps. When $\tilde{q}$ was progressively reached 30, UFCLS using all the 10 BP criteria could successfully unmix all the 19 panel pixels into the abundance fractional maps to which they belonged. These experiments demonstrate that $\tilde{q} = 10$ may not be sufficiently enough to preserve necessary panel pixel information for UFCLS, while $\tilde{q} = 30$ may provide information more than what we needed. Therefore, it seemed that $\tilde{q} = 20$ was an appropriate number for UFCLS to perform spectral unmixing effectively. According to the recent work in Chang and Wang (2006), VD was shown to provide a good estimate on the number of dimensions required to be preserved for DR as well as the number of endmembers for the same HYDICE image scene in Figure 1.15(a). Based on the experiments in Chapter 5, the value estimated by VD is $n_{\mathrm{VD}} = 9$ with $P_F = 10^{-3}$ or 10^{-4} in Chang (2003a). Using $n_{\mathrm{VD}} = 9$ as a guideline, it seemed that for UFCLS the twice n_{VD}, that is, $2n_{\mathrm{VD}}$ could be an appropriate value for $\tilde{q}$ to be used for a BP criterion. In this HYDICE scene, it was 18 that was close to $\tilde{q} = 20$. As a matter of fact, it was indeed the case where the unmixed results for $\tilde{q} = 18$ were very close to those with $\tilde{q} = 20$ in Figure 21.1(a)–(j) and their results are not included in here. This fact was further confirmed and justified by unsupervised LSMA in Chapter 17.

It is worth noting that $\tilde{q} = 9$ was shown to be sufficient for DR in Wang and Chang (2006), but it may not be enough for a BPC for band selection as demonstrated above. This is because DR is performed in the sense of data compaction via a transform compared to BS that only selects separate and individual bands without accounting for interband information. As expected, it will require more bands for BS than dimensions required for DR to have both to perform comparably. Although the selection of $2n_{\mathrm{VD}}$ was empirical, our experiments show that it was a reliable estimate for the $\tilde{q}$ as long as a good BP criterion such as BCM/BDM was used.



10 highest-prioritized bands

20 highest-prioritized bands

30 highest-prioritized bands

(a) Variance

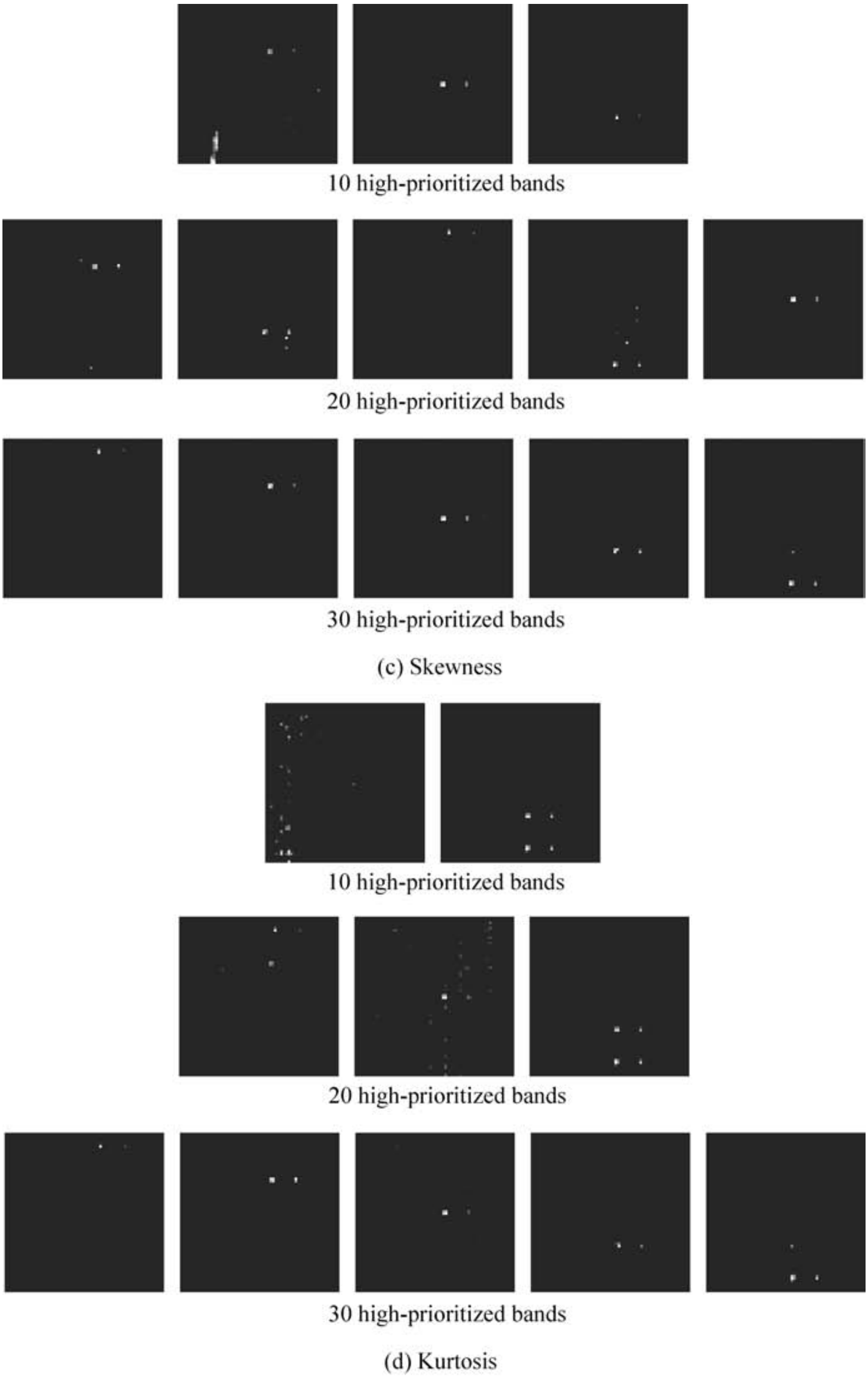
10 high-prioritized bands

20 high-prioritized bands

30 high-prioritized bands

(b) SNR

Figure 21.1 UFCLS-mixed pixel results produced by 10 various BP criteria with $\tilde{q} = 10$, 20, and 30.



10 high-prioritized bands

20 high-prioritized bands

30 high-prioritized bands

(c) Skewness

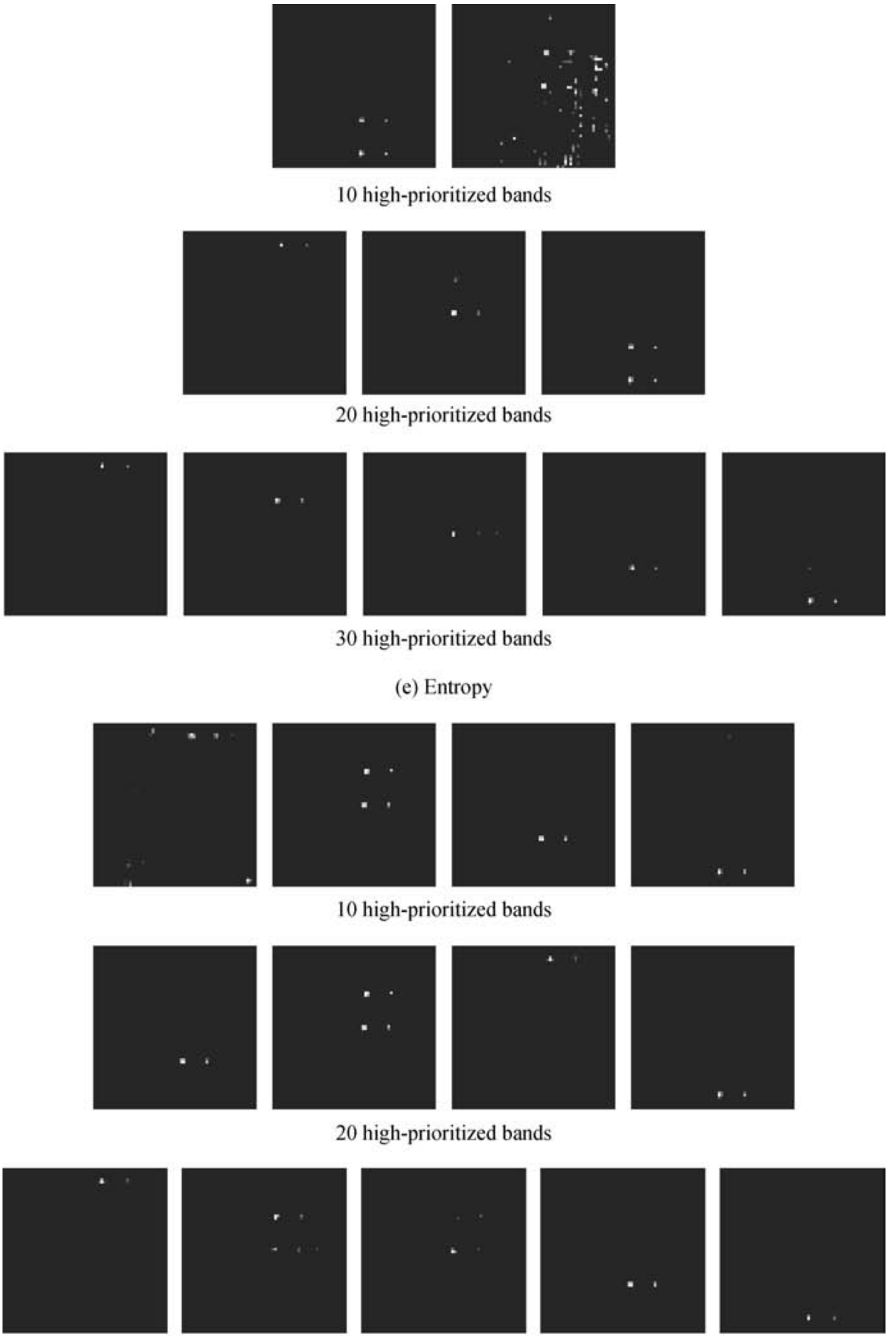
10 high-prioritized bands

20 high-prioritized bands

30 high-prioritized bands

(d) Kurtosis

Figure 21.1 (*Continued*)



10 high-prioritized bands

20 high-prioritized bands

30 high-prioritized bands

(e) Entropy

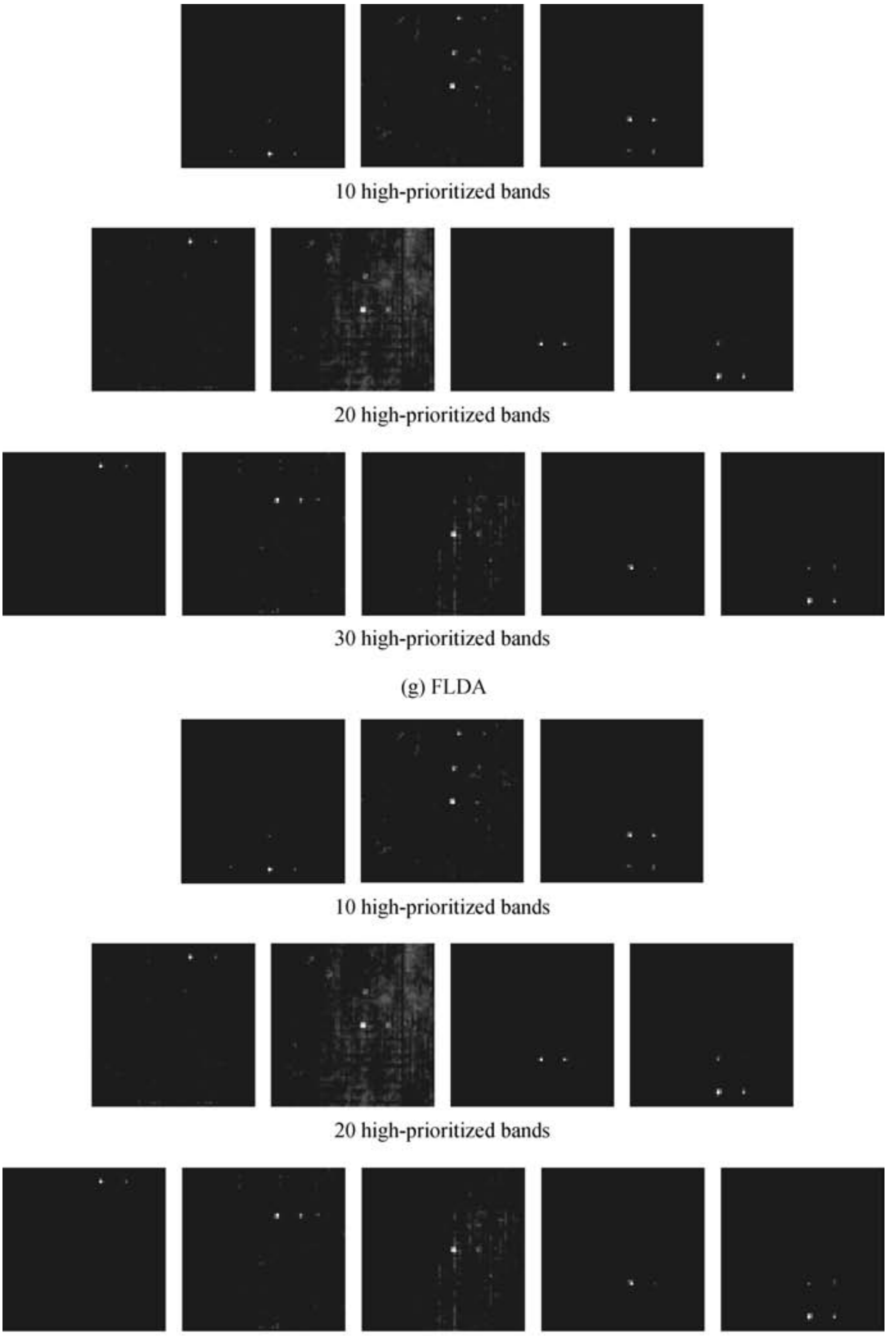
10 high-prioritized bands

20 high-prioritized bands

30 high-prioritized bands

(f) ID

Figure 21.1 (*Continued*)



10 high-prioritized bands

20 high-prioritized bands

30 high-prioritized bands

(g) FLDA

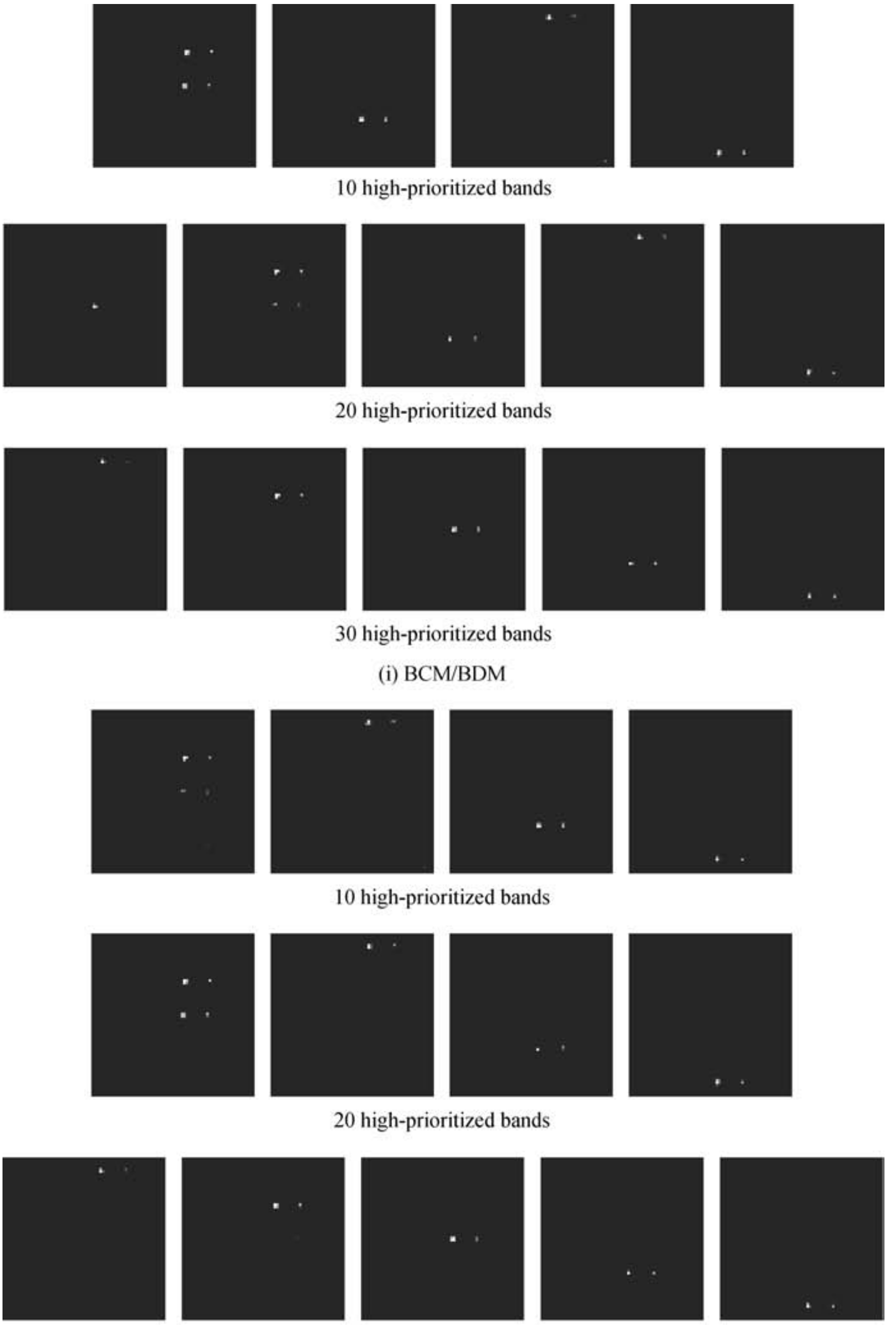
10 high-prioritized bands

20 high-prioritized bands

30 high-prioritized bands

(h) OSP

Figure 21.1 (*Continued*)



10 high-prioritized bands

20 high-prioritized bands

30 high-prioritized bands

(i) BCM/BDM

10 high-prioritized bands

20 high-prioritized bands

30 high-prioritized bands

(j) BCC/BDC

Figure 21.1 (*Continued*)

21.4.1.2 Endmember Extraction

This section presents another application, endmember extraction, which has recently received considerable interest in hyperspectral image analysis. The endmember extraction algorithm to be used in our experiments was the iterative N-finder algorithm (IN-FINDR) developed in Section 7.2.3.2. However, it should be pointed out that there was no particular reason to select IN-FINDR to perform endmember extraction. Any endmember extraction algorithm can also be used for the same purpose such as pixel purity index (PPI).

Following the same experiments conducted in Section 21.4.1.1, Figure 21.2(a)–(j) shows the endmembers extracted by IN-FINDR using highest-prioritized bands with $\tilde{q} = 10$, 20, and 30 according to the 10 various BP criteria in Table 21.2, where the extracted endmembers corresponding to panel signatures are marked by green triangles and the remaining extracted endmembers are marked by red circles.

Table 21.3 also lists the endmembers extracted in Figure 21.2 that correspond to 19 R panel pixels in Figure 1.15(b).

From Table 21.3, none of 10 BP criteria could extract five panel signatures when $\tilde{q} = 10$. The best results were those produced by SNR, skewness, ID, and BCM/BDM, which extracted three panel pixels. However, if $\tilde{q}$ was increased to 20, the skewness was able to extract five endmembers corresponding to R panel pixels that represent five pure panel signatures. If the $\tilde{q}$ was further increased to 30, SNR, skewness, kurtosis, entropy, and BCM/BDM can all extracted five endmembers corresponding to R panel pixels that represent five pure panel signatures. Like unsupervised spectral unmixing, $\tilde{q} = 20$ was shown once again to be a good estimate for $\tilde{q}$ in endmember extraction, provided that a BP criterion was appropriately selected. These two applications demonstrated two important observations. One is that as noted at the end of Section 21.4.1.1, DR required fewer dimensions than the number of bands required by BS because DR performed data compaction compared to BS that only selects a subset of bands while discarding all the information provided by un-selected bands. As a result, VD provided a very good estimate for DR (Wang and Chang, 2006b). Nevertheless, VD can be actually used to estimate the value of $\tilde{q}$ for BP criteria, which is twice VD-estimated value, $2n_{\text{VD}}$. The second observation shows a significance difference between DR and BS in the sense of information preservation. Since DR performs data compaction via a transform, it is expected that most significant information has been transformed and preserved in the first few principal dimensions. By contrast, BS does not compact information as DR does.

Table 21.3 Endmembers corresponding to R panel pixels extracted in Figure 21.2

BP criteria	Endmembers corresponding to panel pixels		
	$\tilde{q} = 10$	$\tilde{q} = 20$	$\tilde{q} = 30$
Variance	p_{11}, p_{521}	p_{521}	$p_{11}, p_{311}, p_{312}, p_{412}, p_{521}$
SNR	p_{11}, p_{211}, p_{411}	$p_{11}, p_{221}, p_{412}, p_{521}, p_{52}$	$p_{11}, p_{22}, p_{311}, p_{412}, p_{521}, p_{52}$
Skewness	p_{11}, p_{312}, p_{521}	$p_{11}, p_{221}, p_{312}, p_{411}, p_{521}$	$p_{11}, p_{221}, p_{312}, p_{411}, p_{412}, p_{521}$
Kurtosis	p_{521}	p_{11}, p_{412}, p_{52}	$p_{11}, p_{22}, p_{312}, p_{412}, p_{521}$
Entropy	p_{311}, p_{521}	p_{311}, p_{521}	$p_{11}, p_{22}, p_{312}, p_{412}, p_{52}$
ID	p_{11}, p_{221}, p_{411}	$p_{11}, p_{211}, p_{412}, p_{521}$	$p_{11}, p_{221}, p_{412}, p_{521}$
FLDA	p_{312}, p_{521}	p_{211},p_{312}, p_{521}	$p_{11}, p_{211}, p_{312}, p_{521}$
OSP	p_{311}, p_{521}	$p_{211}, p_{312}, p_{521}$	$p_{11}, p_{211}, p_{312}, p_{521}$
BCM/BDM	p_{11}, p_{22}, p_{411}	$p_{11}, p_{22}, p_{411}, p_{52}$	$p_{11}, p_{22}, p_{311}, p_{411}, p_{412}, p_{52}$
BCC/BDC	p_{22}, p_{411}	$p_{11}, p_{221}, p_{411}, p_{412}, p_{521}$	$p_{11}, p_{221}, p_{411}, p_{521}$

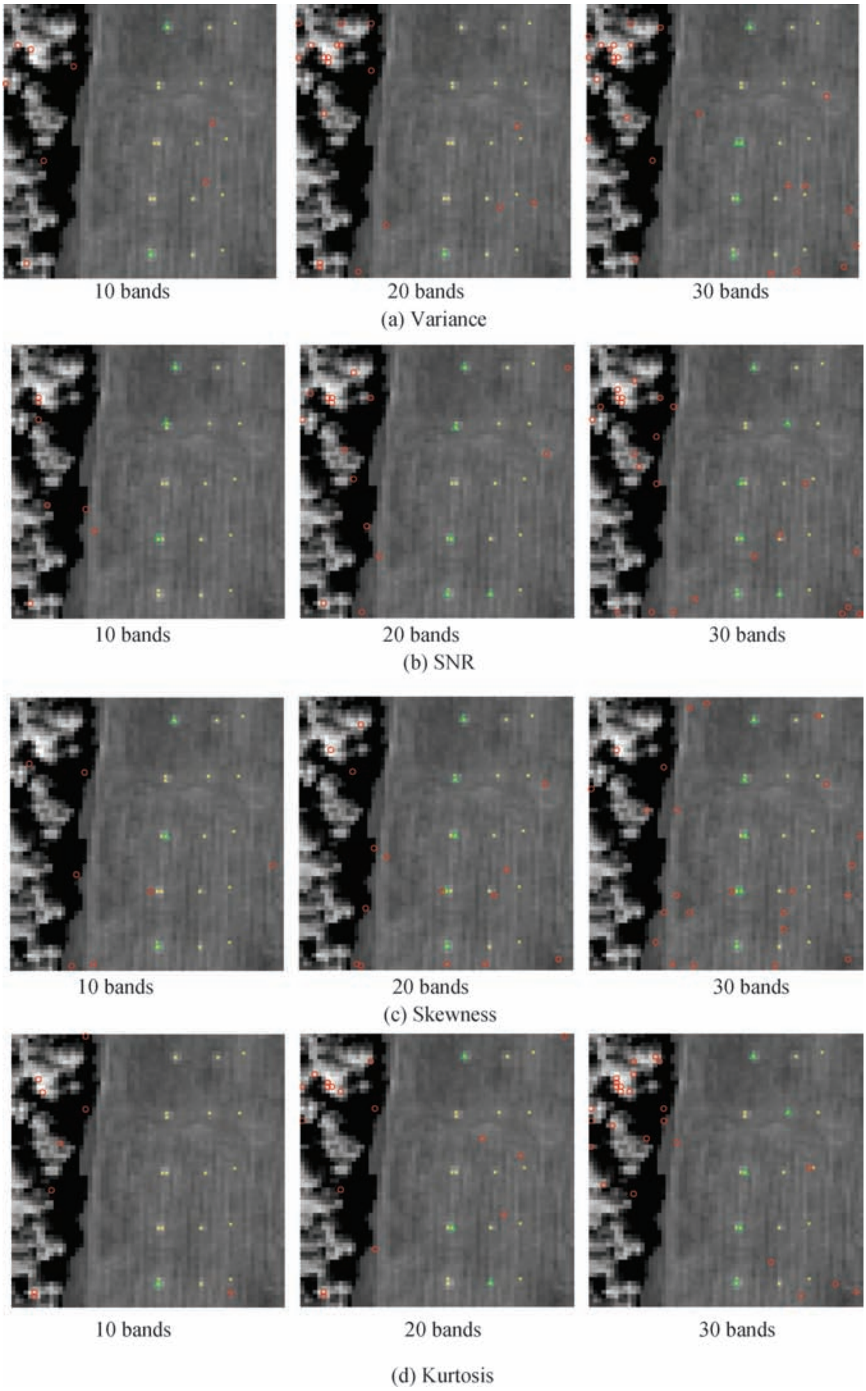
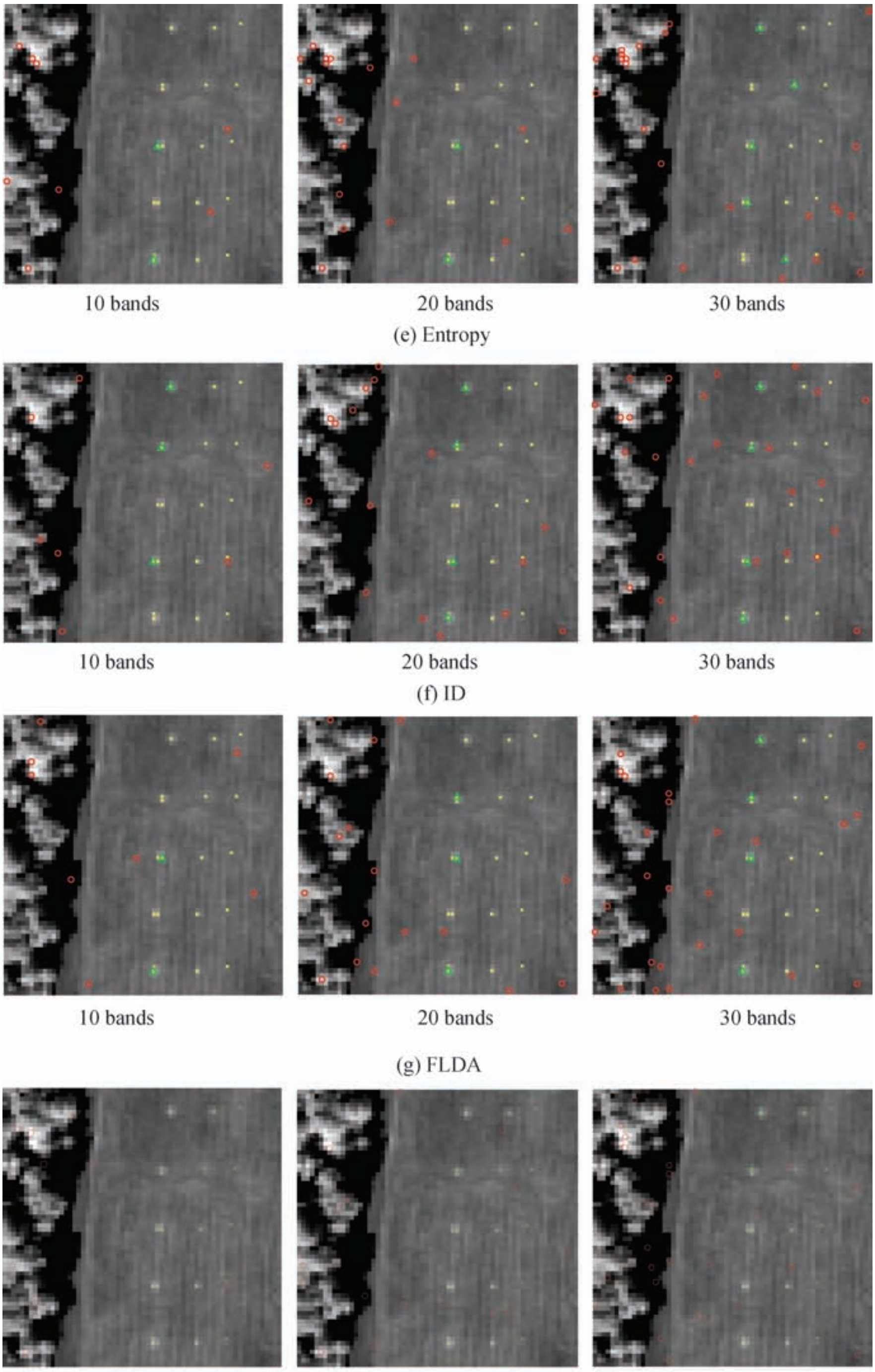


Figure 21.2 Endmembers extracted by IN-FINDR with $\tilde{q} = 10$, 20, and 30.



10 bands 20 bands 30 bands

(e) Entropy

10 bands 20 bands 30 bands

(f) ID

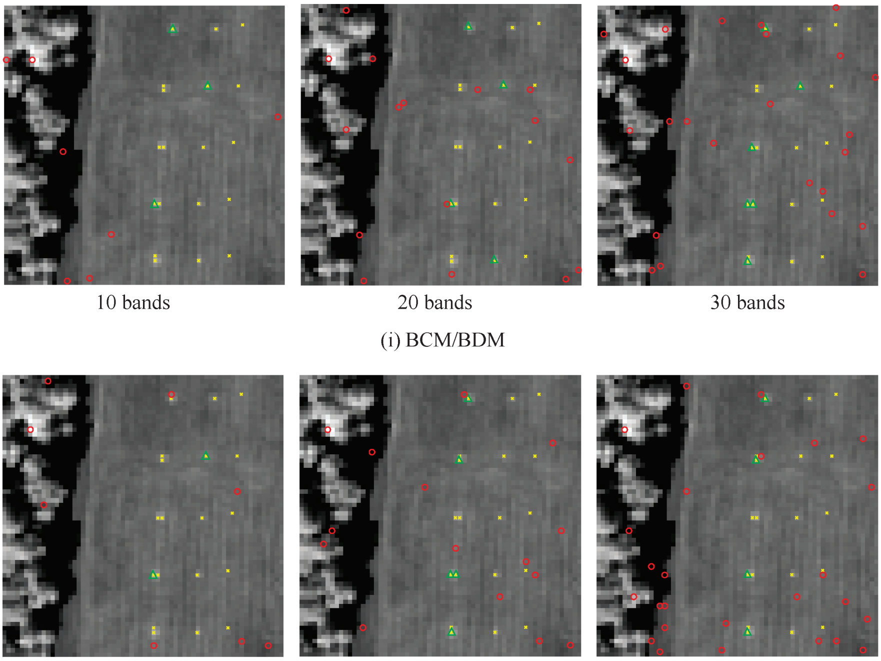
10 bands 20 bands 30 bands

(g) FLDA

10 bands 20 bands 30 bands

(h) OSP

Figure 21.2 (*Continued*)



10 bands 20 bands 30 bands

(i) BCM/BDM

10 bands 20 bands 30 bands

(j) BCC/BDC

Figure 21.2 (*Continued*)

Instead, it selects the most significant bands to preserve desired information without taking advantage of transformed information. Consequently, it is anticipated that given the same number of bands and dimensions/components, the information preserved by DR is generally greater than that by BS. Consequently, BS requires more bands than DR in order for BS to preserve the same level of information by DR.

21.4.2 Applications Using Least-Prioritized Bands

It is a common practice that BS must select bands with highest-priority scores. However, it turns out that such an approach does not necessarily yield the best performance in some applications. More specifically, on some occasions selecting bands with the least-priority scores may produce better results. The significance of selecting least-priority bands may be due to the fact that subtle information such as small targets or anomalies can be only preserved and captured by bands with least-priority scores rather than those with highest-priority scores if a selected BP criterion is second-order statistics. In the following, we investigated this interesting issue, which has not been explored in the past.

To shed light on this issue, a simple experiment using principal components analysis (PCA) was performed to provide insight into the use of bands with least-priority scores. Since the 19 panel pixels in the HYDICE scene in Figure 1.15(a) can be considered as small and rare targets with respect to the entire image scene, the PCA-based principal components and second-order statistics-based BP criteria may not be effective to capture the characteristics of these 19 panel pixels, which instead may be hidden in minor components. In order to validate our assumption, once again IN-FINDR using nine PCA-generated principal components (PCs) and nine minor

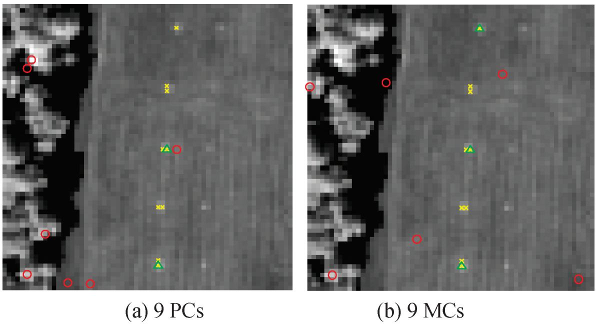

(a) 9 PCs (b) 9 MCs

Figure 21.3 Nine endmembers extracted by IN-FINDR using nine PCs and nine MCs.

components (MCs) were implemented where Figure 21.3(a) and (b) shows their respective results. As we can see from the figure, three panel pixels p_{11}, p_{312}, p_{521} were extracted by using nine MCs in Figure 21.3(b) compared to the only two panel pixels p_{312}, p_{521} extracted by using nine PCs in Figure 21.3(a).

Inspired by the results in Figure 21.3, we repeated the same experiments conducted in previous sections and compared the results against those results obtained by highest-prioritized bands. Table 21.4 lists the first 30 bands with least-priority scores selected by 10 various BP criteria with a backslash "/" used to separate two selected bands. As also noted, if one band had lower-priority scores, its neighboring bands also had similar lower-priority scores. Due to the fact that the uniform band selection does not involve BP, its experiments are not included.

With each BPC the 30 bands in Table 21.4 are separated by three categories, first set of least-prioritized 10 bands, second set of follow-up least-prioritized 10 bands, and third set of the next follow-up least-prioritized 10 bands to evaluate the effectiveness of each BPC in performance analysis by adding the next 10 least-prioritized bands at a time until reaching a total number of 30 least-prioritized bands.

21.4.2.1 Unsupervised Linear Spectral Mixture Analysis

Like the experiments in Section 21.4.1.1 the unsupervised spectral unmixing was performed in exactly the same manner for $\tilde{q} = 10$, 20, and 30 except that the least-prioritized bands in Table 21.4 were selected to form new data sets to replace the original 169-band HYDICE image scene. ATGP was also used to generate $\tilde{q}$ target pixels to form the desired signature matrix for FCLS. Figure 21.4 (a)–(j) shows the UFCLS-mixed pixel abundance fractional maps of the 19 panel pixels resulting from using $\tilde{q} = 10$, 20, and 30 least-prioritized bands selected by the 10 BP criteria in Table 21.4.

Comparing the results in Figure 21.4 to those in Figure 21.1, the unmixing performances using highest-prioritized and least-prioritized bands were quite different. For example, the best performance using the first 20 least-prioritized bands was produced by the BCC/BDC in Figure 21.4(j) compared to the results produced by both the skewness in Figure 21.1(c) and BCM/BDM in Figure 21.1(i) using the first 20 highest-prioritized bands.

It should be noted that the panels in HYDICE data are relatively small and they are only of the one-pixel or two-pixel size. As a result, their spectral characteristics may not be captured by some BP criteria such as variance using highest-prioritized band, but rather by least-prioritized bands since these panels do not generally contribute much to data variance and thus do not appear in highest-prioritized bands, a fact also justified by minor components when PCA is used. Therefore,

Table 21.4 30 least-prioritized bands selected progressively by various 10 BP criteria

	30 least-priority scores
Variance	(First 10 bands) 168/169/167/164/125/124/165/166/163/162
	(Second 10 bands) 160/161/159/157/158/156/154/155/123/153
	(Third 10 bands) 126/130/152/129/122/151/150/131/149/148
SNR	(First 10 bands) 1/2/3/4/5/6/7/8/168/9
	(Second 10 bands) 10/169/167/11/12/164/165/13/166/163
	(Third 10 bands) 14/15/68/16/124/125/162/160/161/18
Skewness	(First 10 bands) 95/93/94/92/96/91/90/89/88/87
	(Second 10 bands) 97/55/85/86/81/82/79/83/80/77
	(Third 10 bands) 76/78/75/74/84/111/112/110/113/54
Kurtosis	(First 10 bands) 50/49/48/47/51/46/45/44/43/42
	(Second 10 bands) 41/22/23/24/40/21/25/20/39/19
	(Third 10 bands) 18/26/52/17/38/16/27/15/14/37
Entropy	(First 10 bands) 168/169/167/164/165/124/125/166/163/162
	(Second 10 bands) 160/161/159/157/158/156/154/155/153/123
	(Third 10 bands) 152/130/126/151/129/122/150/149/131/148
ID	(First 10 bands) 43/9/23/42/52/45/37/71/49/47
	(Second 10 bands) 39/48/50/26/38/40/44/51/20/6
	(Third 10 bands) 41/73/29/36/46/17/27/25/12/21
FLDA	(First 10 bands) 166/163/161/76/164/169/157/120/167/75
	(Second 10 bands) 153/165/159/155/156/162/160/168/154/148
	(Third 10 bands) 125/32/70/113/152/88/150/143/73/146
OSP	(First 10 bands) 168/169/167/124/125/164/165/166/163/162
	(Second 10 bands) 160/161/159/158/157/156/123/130/126/155
	(Third 10 bands) 151/129/150/154/152/122/149/153/131/148
BCM/BDM	(First 10 bands) 59/60/66/67/65/57/61/64/56/68
	(Second 10 bands) 62/78/77/76/58/80/55/81/69/63
	(Third 10 bands) 70/79/75/82/74/92/54/93/91/95
BCC/BDC	(First 10 bands) 165/128/162/49/126/159/160/17/45/43
	(Second 10 bands) 25/8/38/16/19/14/59/65/61/60
	(Third 10 bands) 66/67/57/62/64/56/68/63/78/79

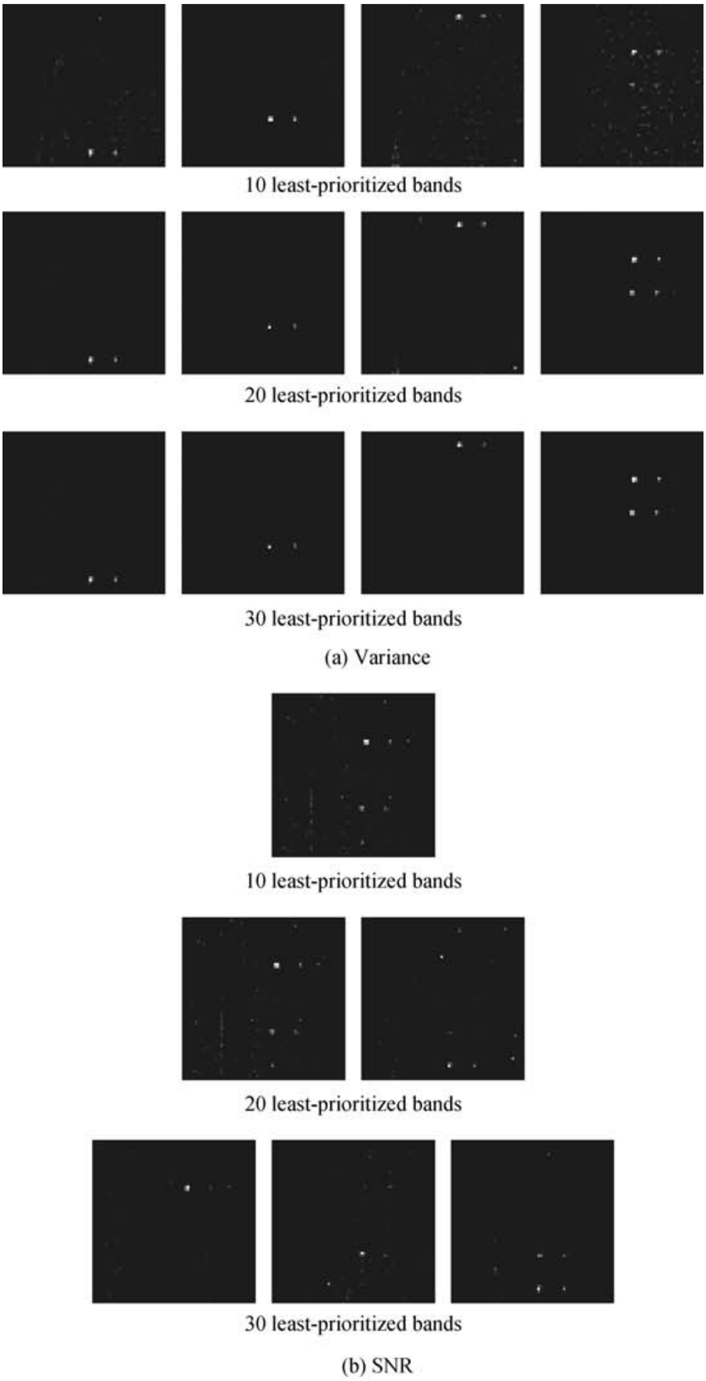
the above experiments demonstrated that selecting bands with least-priority scores could be an alternative BS to improve performance.

21.4.2.2 Endmember Extraction

Once again, the same experiments conducted in Section 21.4.1.2 were also performed for endmember extraction except that the first 30 least-prioritized bands were used. Figure 21.5(a)–(j) shows the endmembers extracted by IN-FINDR using $\tilde{q}$ least-prioritized bands with $\tilde{q} = 10$, 20, and 30 according to the 10 various BP criteria in Table 21.4, where the extracted endmembers corresponding to panel signatures are marked by green triangles and the remaining extracted endmembers are marked by red circles.

Table 21.5 summarizes the IN-FINDR-extracted endmembers in Figure 21.5 that corresponded to panel pixels representing the five distinct panel signatures.

From the results in Figure 21.5 and Table 21.5, the only one to extract the five distinct panel signatures with $\tilde{q} = 20$ bands with least-priority scores is the BCC/BDC. This was different from



10 least-prioritized bands

20 least-prioritized bands

30 least-prioritized bands

(a) Variance

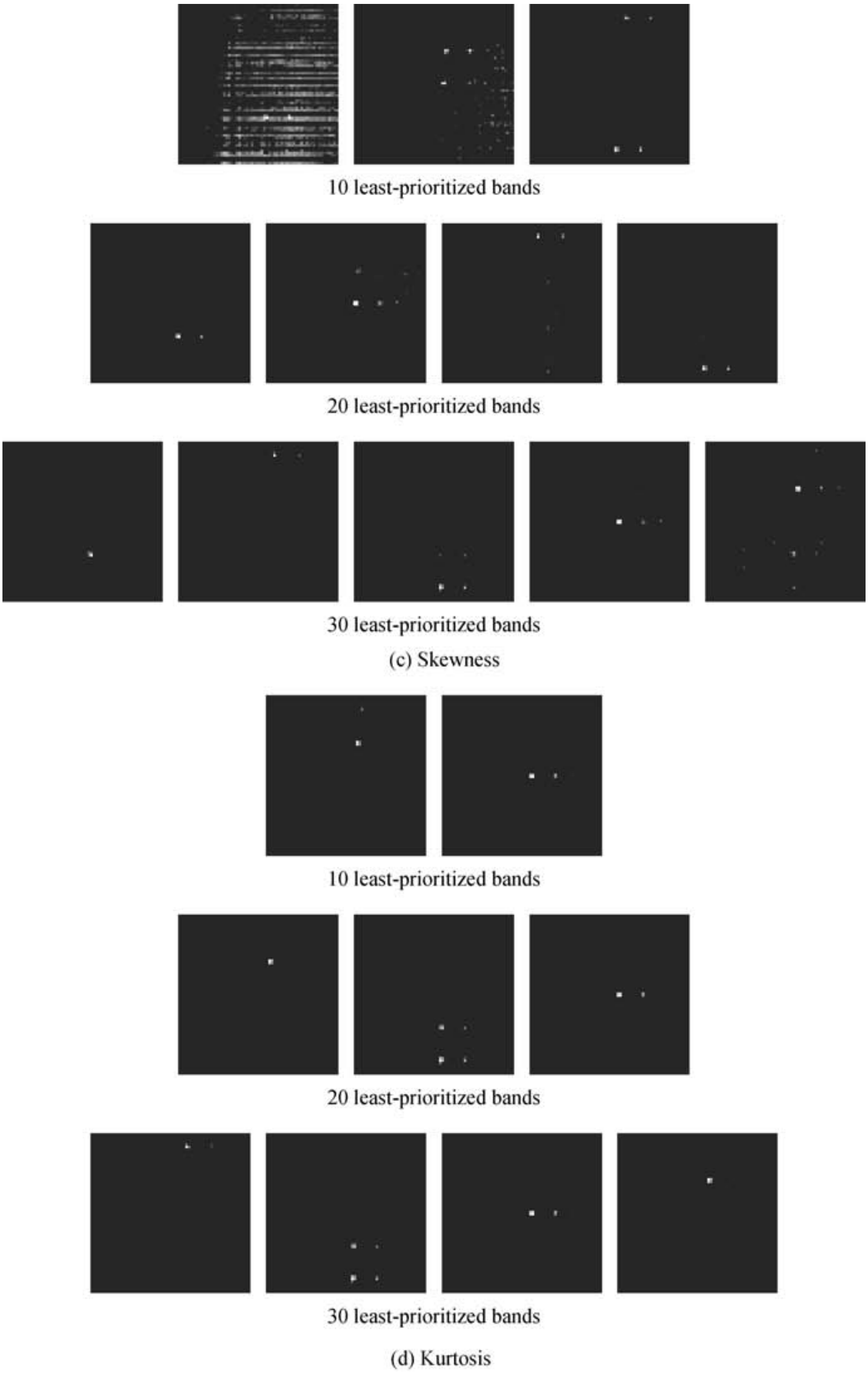
10 least-prioritized bands

20 least-prioritized bands

30 least-prioritized bands

(b) SNR

Figure 21.4 UFCLS-mixed pixel results produced by 10 various BP criteria with $\tilde{q} = 10$, 20, and 30.



10 least-prioritized bands

20 least-prioritized bands

30 least-prioritized bands

(c) Skewness

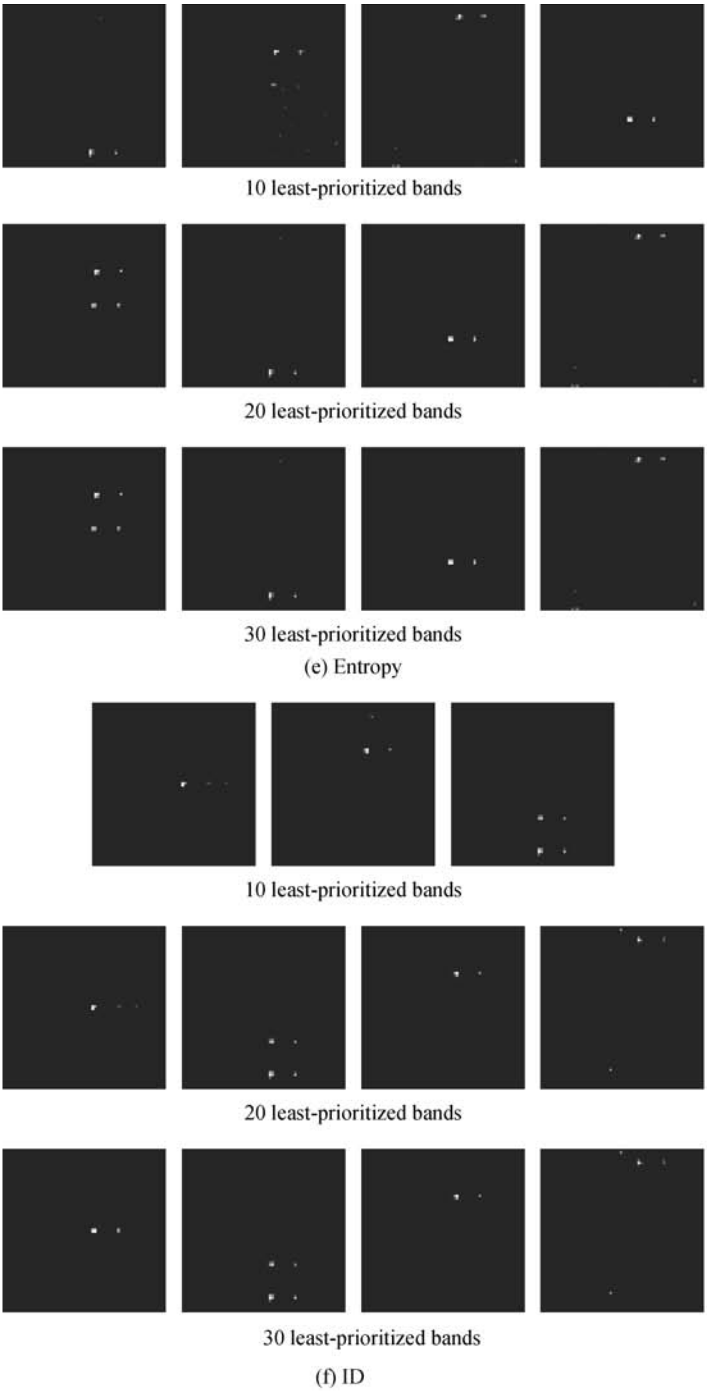
10 least-prioritized bands

20 least-prioritized bands

30 least-prioritized bands

(d) Kurtosis

Figure 21.4 (*Continued*)



10 least-prioritized bands

20 least-prioritized bands

30 least-prioritized bands

(e) Entropy

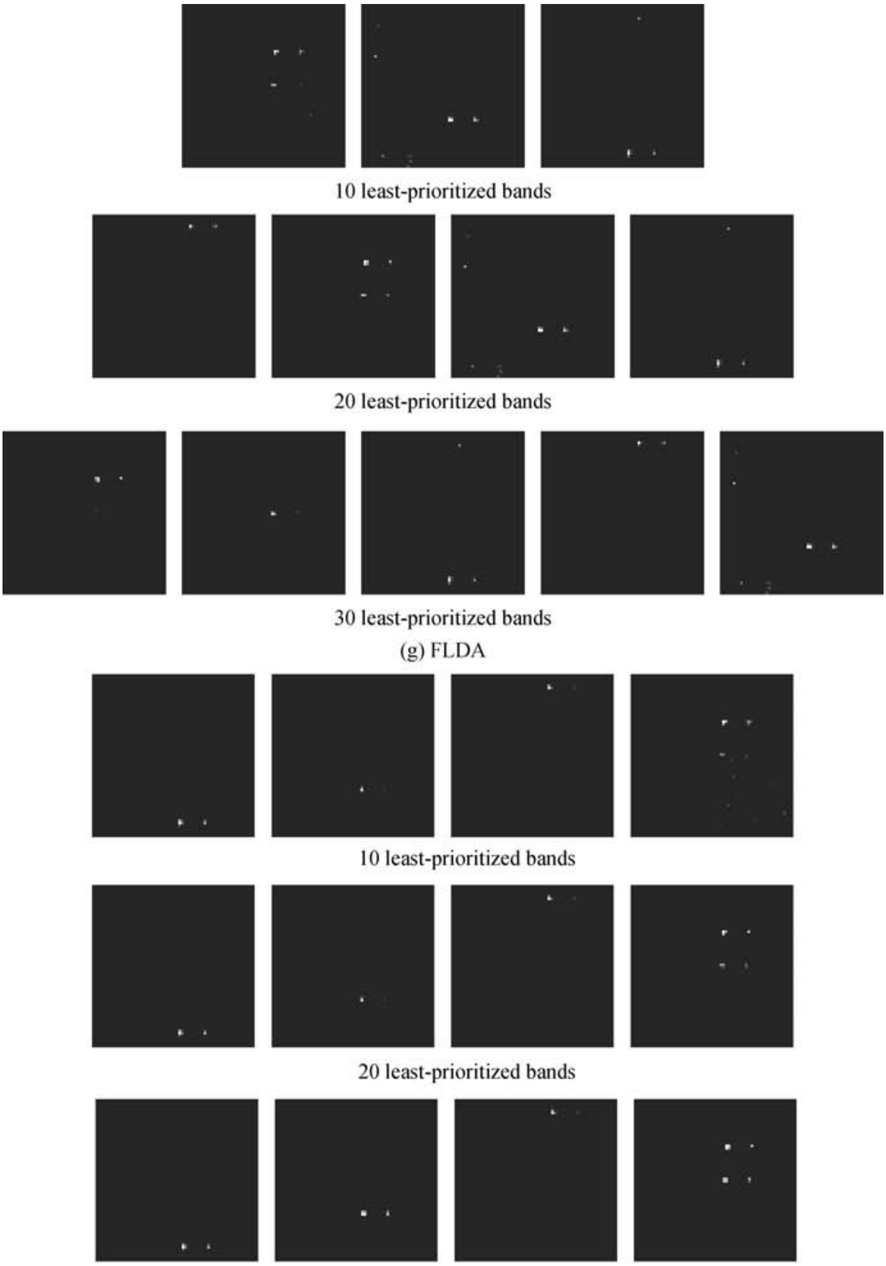
10 least-prioritized bands

20 least-prioritized bands

30 least-prioritized bands

(f) ID

Figure 21.4 (*Continued*)



10 least-prioritized bands

20 least-prioritized bands

30 least-prioritized bands

(g) FLDA

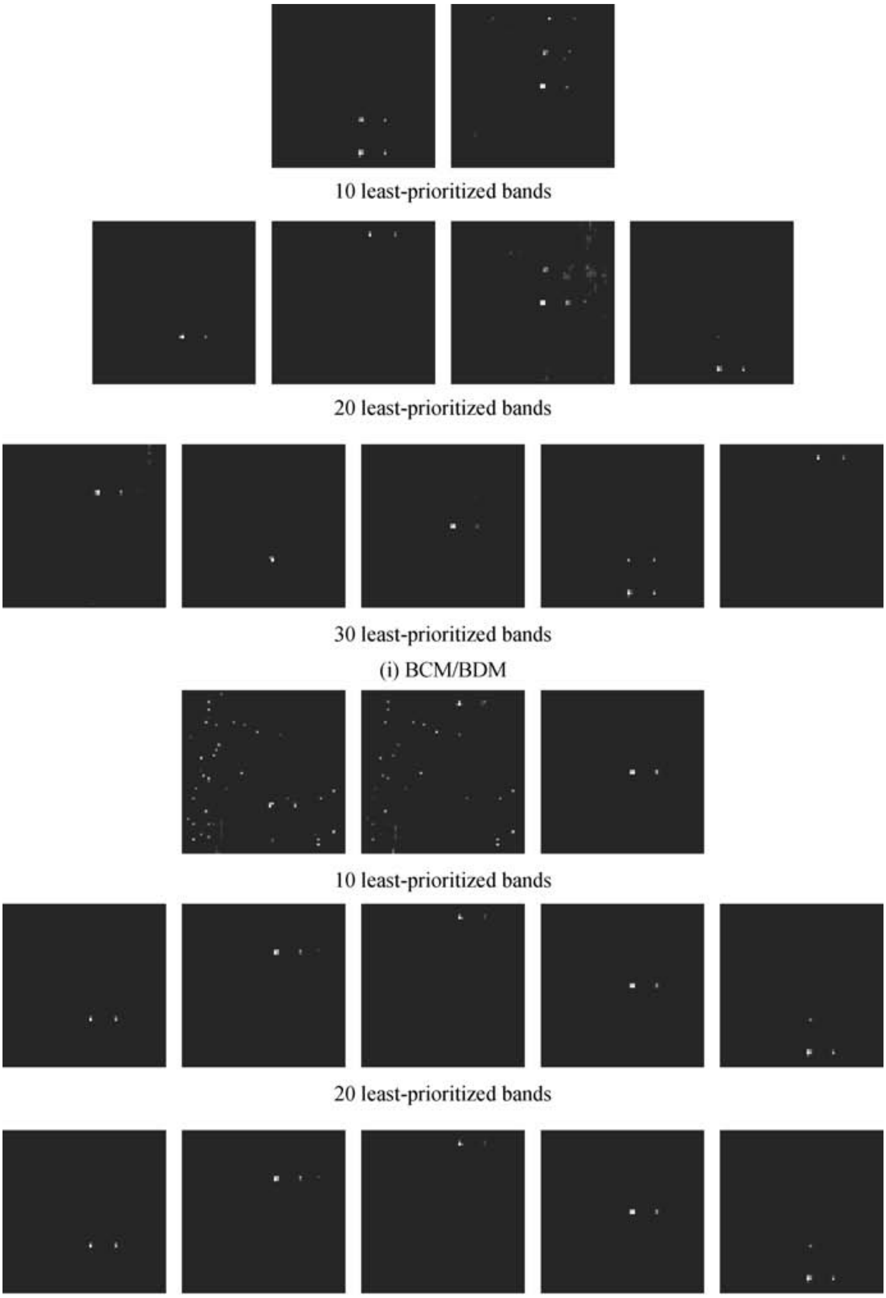
10 least-prioritized bands

20 least-prioritized bands

30 least-prioritized bands

(h) OSP

Figure 21.4 (*Continued*)



10 least-prioritized bands

20 least-prioritized bands

30 least-prioritized bands

(i) BCM/BDM

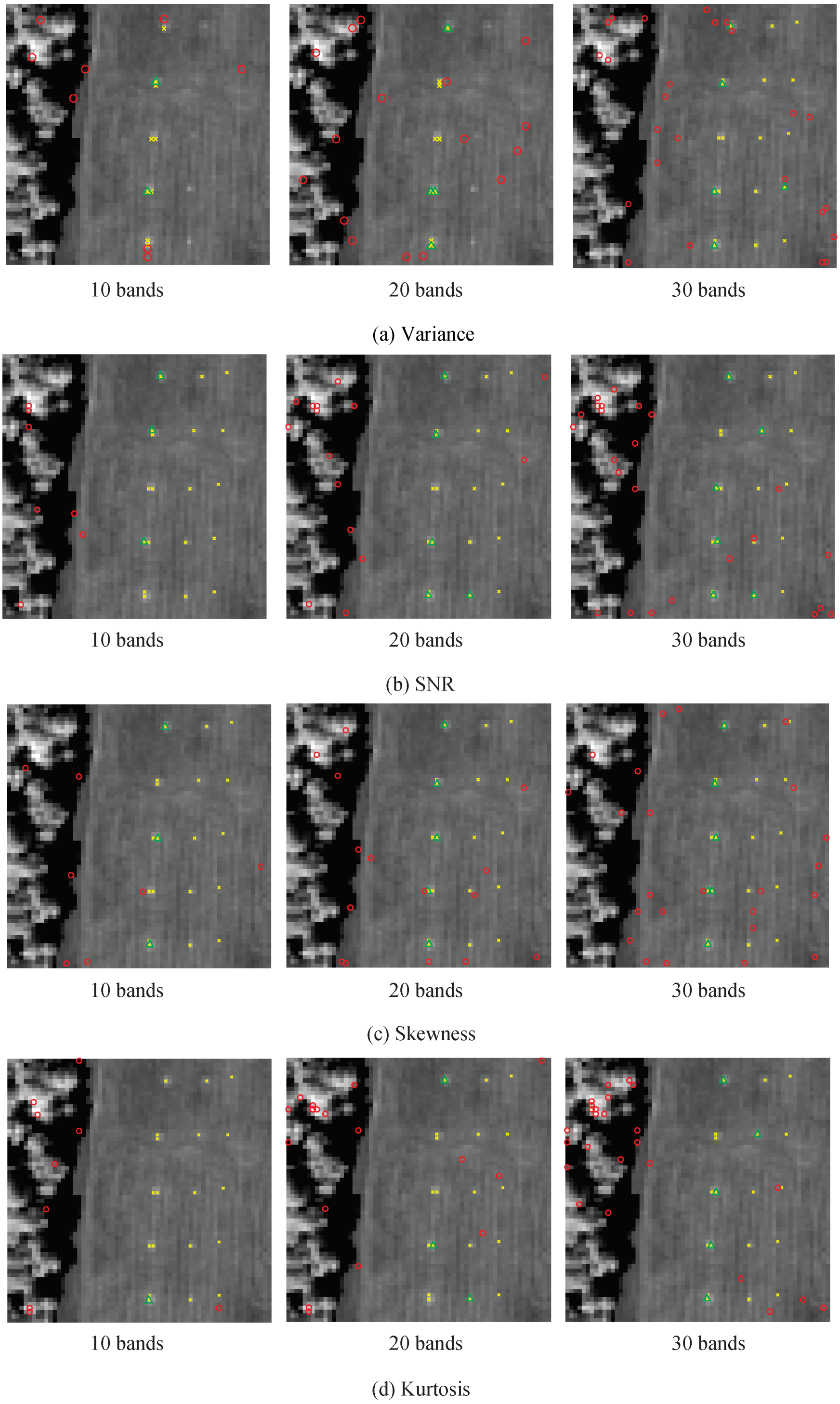
10 least-prioritized bands

20 least-prioritized bands

30 least-prioritized bands

(j) BCC/BDC

Figure 21.4 (*Continued*)



10 bands 20 bands 30 bands

(a) Variance

10 bands 20 bands 30 bands

(b) SNR

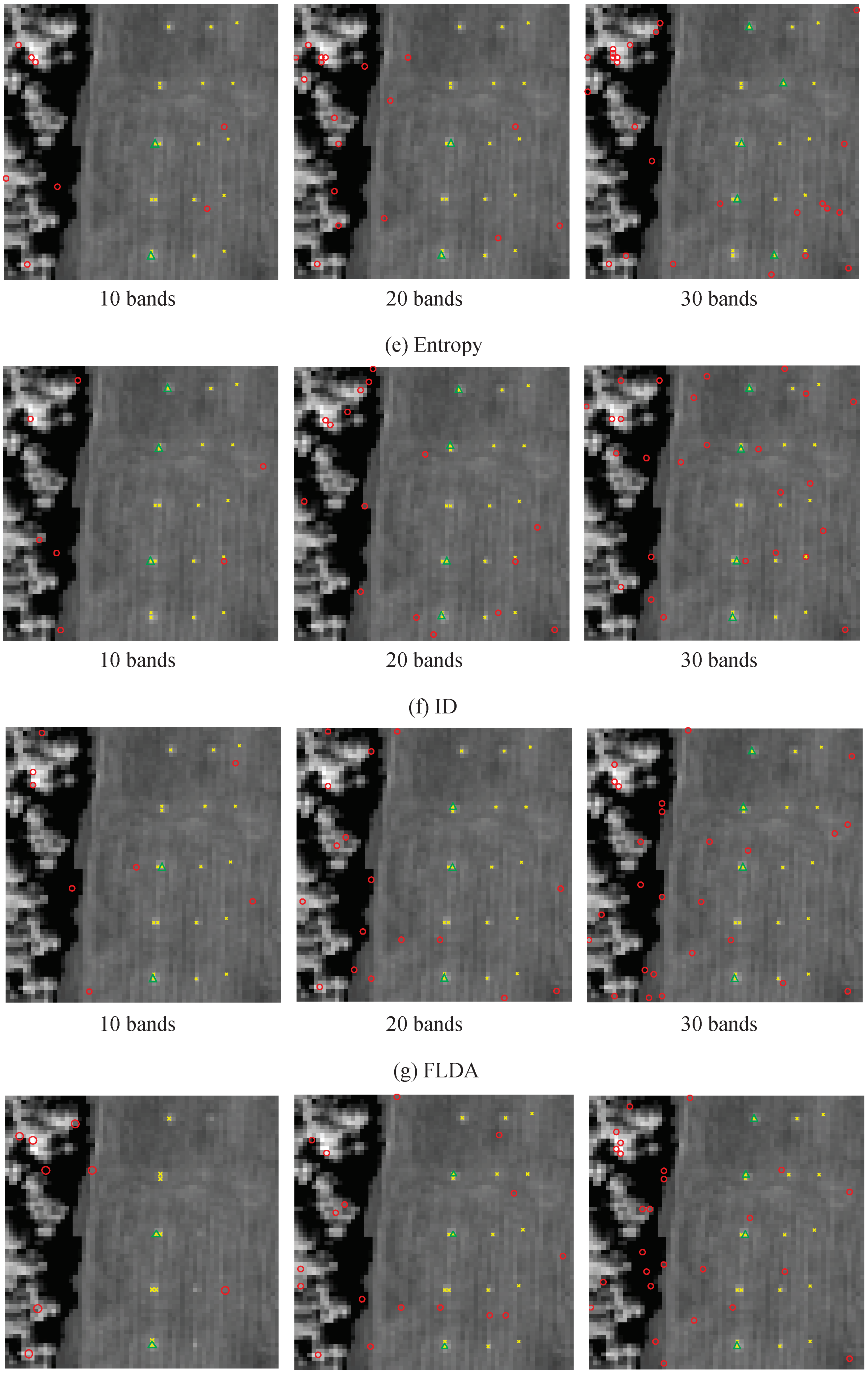
10 bands 20 bands 30 bands

(c) Skewness

10 bands 20 bands 30 bands

(d) Kurtosis

Figure 21.5 Endmembers extracted by IN-FINDR with $\tilde{q} = 10$, 20, and 30.



10 bands 20 bands 30 bands

(e) Entropy

10 bands 20 bands 30 bands

(f) ID

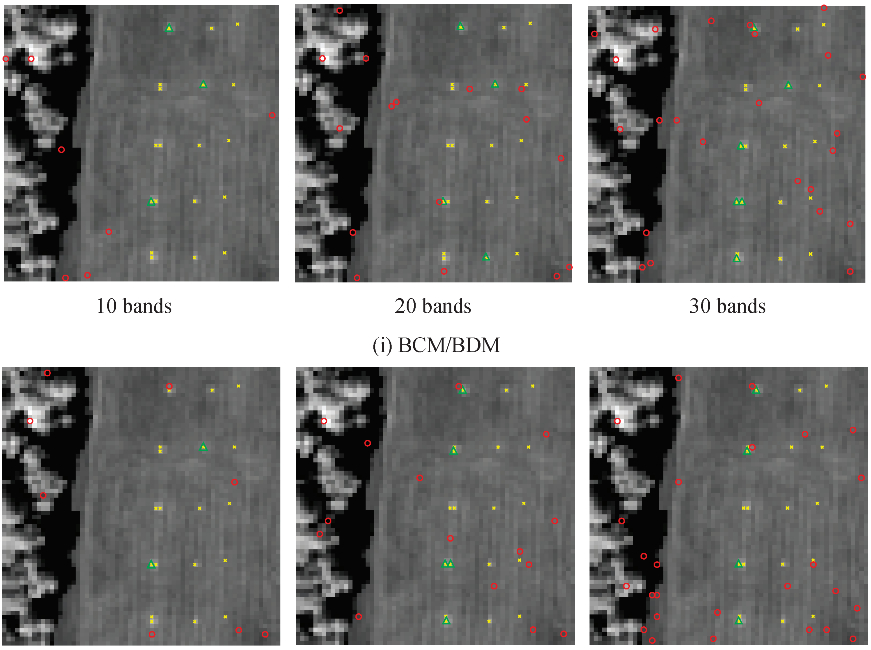
10 bands 20 bands 30 bands

(g) FLDA

10 bands 20 bands 30 bands

(h) OSP

Figure 21.5 (*Continued*)



10 bands 20 bands 30 bands

(i) BCM/BDM

10 bands 20 bands 30 bands

(j) BCC/BDC

Figure 21.5 (*Continued*)

Table 21.5 Endmembers extracted in Figure 21.5 that correspond to panel pixels

BP criteria	Endmembers corresponding to panel pixels		
	$\tilde{q} = 10$	$\tilde{q} = 20$	$\tilde{q} = 30$
Variance	p_{211}, p_{411}	$p_{11}, p_{411}, p_{412}, p_{521}$	$p_{11}, p_{221}, p_{411}, p_{43}, p_{521}$
SNR	p_{221}	p_{221}, p_{521}	$p_{211}, p_{411}, p_{521}$
Skewness	p_{11}, p_{312}	p_{11}, p_{311}, p_{411}	$p_{11}, p_{221}, p_{312}, p_{411}, p_{412}, p_{521}$
Kurtosis	p_{211}, p_{312}	$p_{211}, p_{312}, p_{521}$	$p_{11}, p_{211}, p_{311}, p_{521}$
Entropy	p_{11}, p_{411}, p_{52}	p_{11}, p_{411}, p_{521}	$p_{11}, p_{211}, p_{411}, p_{511}, p_{521}$
ID	$p_{211}, p_{311}, p_{312}, p_{521}$	$p_{11}, p_{221}, p_{311}, p_{312}, p_{521}$	$p_{11}, p_{221}, p_{311}, p_{312}, p_{521}$
FLDA	p_{211}, p_{412}	$p_{11}, p_{211}, p_{412}, p_{521}$	$p_{11}, p_{221}, p_{311}, p_{411}, p_{521}$
OSP	$p_{11}, p_{211}, p_{411}, p_{521}$	$p_{11}, p_{22}, p_{411}, p_{412}, p_{521}$	$p_{11}, p_{221}, p_{312}, p_{411}, p_{42}, p_{521}$
BCM/BDM	p_{11}, p_{521}	p_{11}, p_{412}, p_{521}	$p_{11}, p_{311}, p_{412}, p_{521}$
BCC/BDC	p_{11}, p_{312}, p_{411}	$p_{11}, p_{211}, p_{312}, p_{411}, p_{521}$	$p_{11}, p_{211}, p_{312}, p_{411}, p_{521}$

the results in Figure 21.2 and Table 21.3 where the skewness using $\tilde{q} = 20$ bands with highest-priority scores was the only one to extract the five distinct panel signatures.

21.4.3 Applications Using Mixing Highest-Prioritized and Least-Prioritized Bands

As shown in Sections 21.4.1 and 21.4.2 using highest-prioritized and least-prioritized spectral bands had different advantages. In this section it will be interesting to see if using a combined set of highest- and least-prioritized spectral bands can take both advantages as demonstrated by using mixed component analysis for spectral/spatial compression in Example 19.5 in Chapter 19.

21.4.3.1 Unsupervised Linear Spectral Mixture Analysis

According to experiments in Figures 21.1 and 21.4, 20 bands seemed to be an appropriate estimate for the value of $\tilde{q}$ used by BS. So, in this section, only the experiments using 20 bands were conducted for a comparative analysis. In this case, we mixed the 10 highest-prioritized and the 10 least-prioritized bands to make $\tilde{q} = 20$ spectral bands to implement UFCLS for spectral unmixing. Figure 21.6(a)–(j) shows the unmixed results produced by using the 10 BP criteria listed in Tables 21.1.

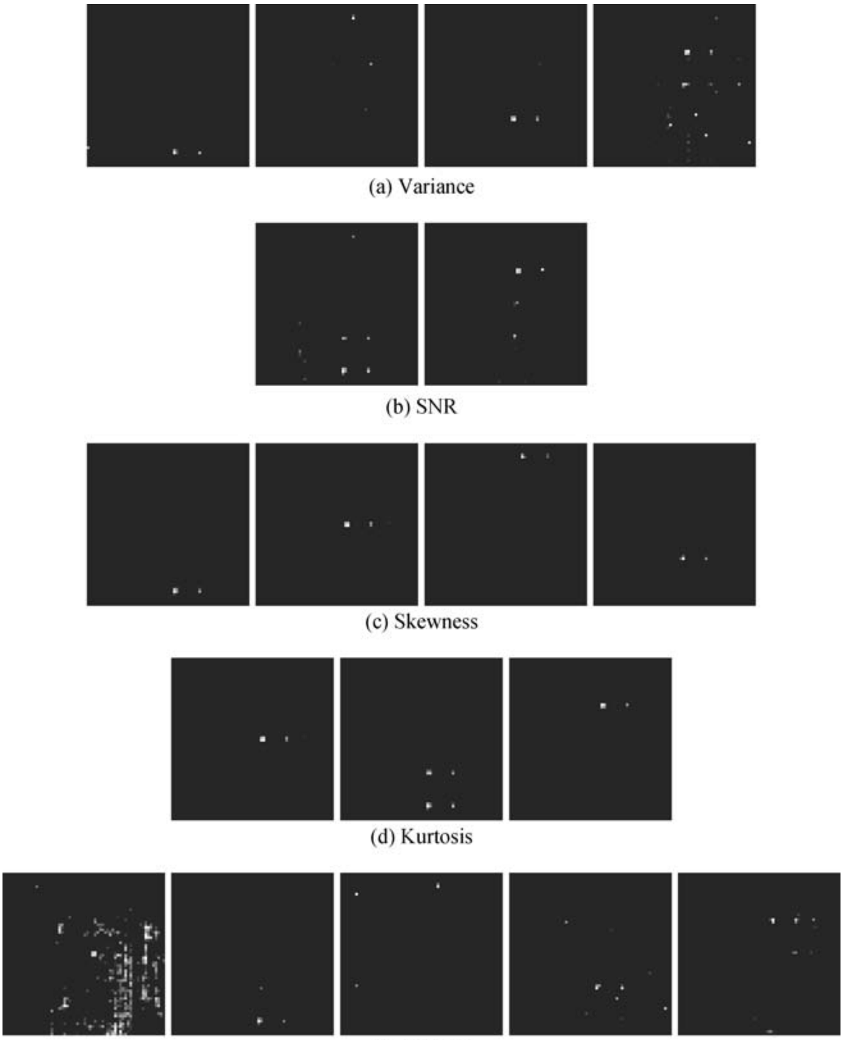
As noted in Sections 21.4.1 and 21.4.2, if only 20 bands were allowed to implement UFCLS, only the skewness and BCM/BDM in Figure 21.1 and the BCC/BDC in Figure 21.4 could unmix panel pixels in their corresponding five rows. However, if these 20 bands were selected by mixing 10 highest-prioritized and 10 least-prioritized bands, Figure 21.6 shows that there were four BP criteria, entropy, ID, FLDA, and BCC/BDC that could indeed achieve better unmixed results for the 19 panel pixels. These experiment demonstrated that using 20 highest-prioritized bands or 20 least-prioritized bands alone may not be as effective as mixing 10 highest-prioritized bands and 10 least-prioritized bands. However, this conclusion was not true in endmember extraction as will be demonstrated in the following section.

21.4.3.2 Endmember Extraction

Interestingly, if we implemented the same 10 BP criteria by selecting their first 10 bands with highest-priority scores in Table 21.2 and 10 bands with least-priority scores in Table 21.4 to make up $\tilde{q} = 20$ bands, the 20 endmembers extracted by IN-FINDR are shown in Figure 21.7 where the endmembers marked by triangles and open circles represent panel and non-panel pixels, respectively.

Table 21.6 also tabulates the panel pixels extracted in Figure 21.7 along with the results from Table 21.3 and Table 21.5 for $\tilde{q} = 20$ for comparison.

Unlike the results produced by unsupervised spectral unmixing in Figure 21.6, Table 21.6 demonstrated that endmember extraction using a combination of the first 10 highest-prioritized and the first 10 least-prioritized bands did not necessarily perform better than that using only the first 20 highest-prioritized bands or the first 20 least-prioritized bands alone. As a matter of fact, none of the 10 BP criteria could extract the five distinct panel signatures compared to the skewness using the first 20 highest-prioritized bands and the BCC/BDC using the first 20 least-prioritized bands that could pull out all the five panel signatures. Nevertheless, endmember extraction was more effective if the first 20 least-prioritized bands were used for most of the 10 BP criteria. This is mainly due to the fact that endmembers are usually considered as rare signatures and insignificant targets in which case only least-prioritized bands can effectively preserve their information.



(a) Variance

(b) SNR

(c) Skewness

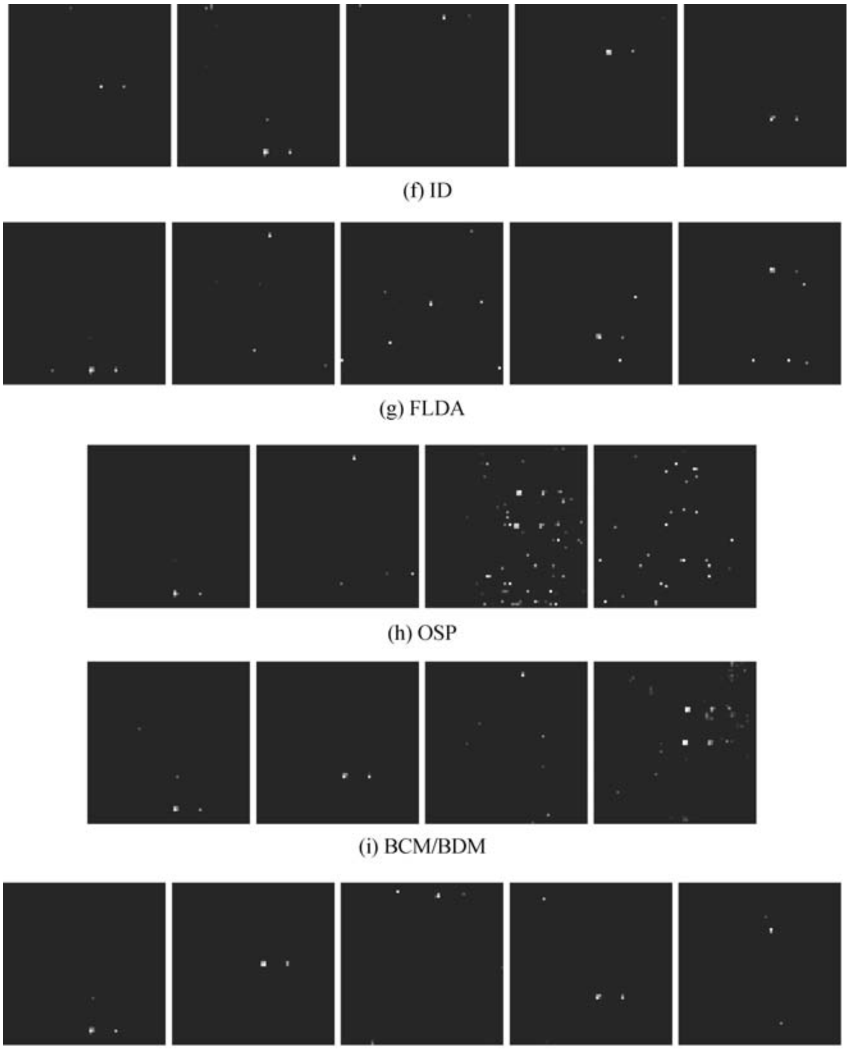
(d) Kurtosis

(e) Entropy

Figure 21.6 UFCLS-mixed pixel results produced by 10 BP criteria with $\tilde{q} = 20$ using the 10 highest-prioritized and the 10 least-prioritized bands.

As a final remark, all the results in Sections 21.4.6, 21.4.7 and 21.4.3 provided solid evidence that $\tilde{q} = 20$ was in fact a very good estimate for the number of spectral bands to be selected for spectral unmixing and endmember extraction. The number of 20 was very close to 18, which is twice VD-estimated value, $n_{\mathrm{VD}} = 9$ with $P_F = 10^{-4}$ which was further confirmed by the experiments conducted in Chapter 17.

As another example for endmember extraction, the real Cuprite data in Figure 1.12(a) is also used to conduct similar experiments for comparison. It should be noted that unlike the HYDICE, which provides complete knowledge of all 19 R panel pixels, the ground truth provided by the Cuprite image scene does not include complete knowledge of all mineral signatures. In this case, the two supervised BP criteria, OSP and FLDA, were not used for a comparative study.



(f) ID

(g) FLDA

(h) OSP

(i) BCM/BDM

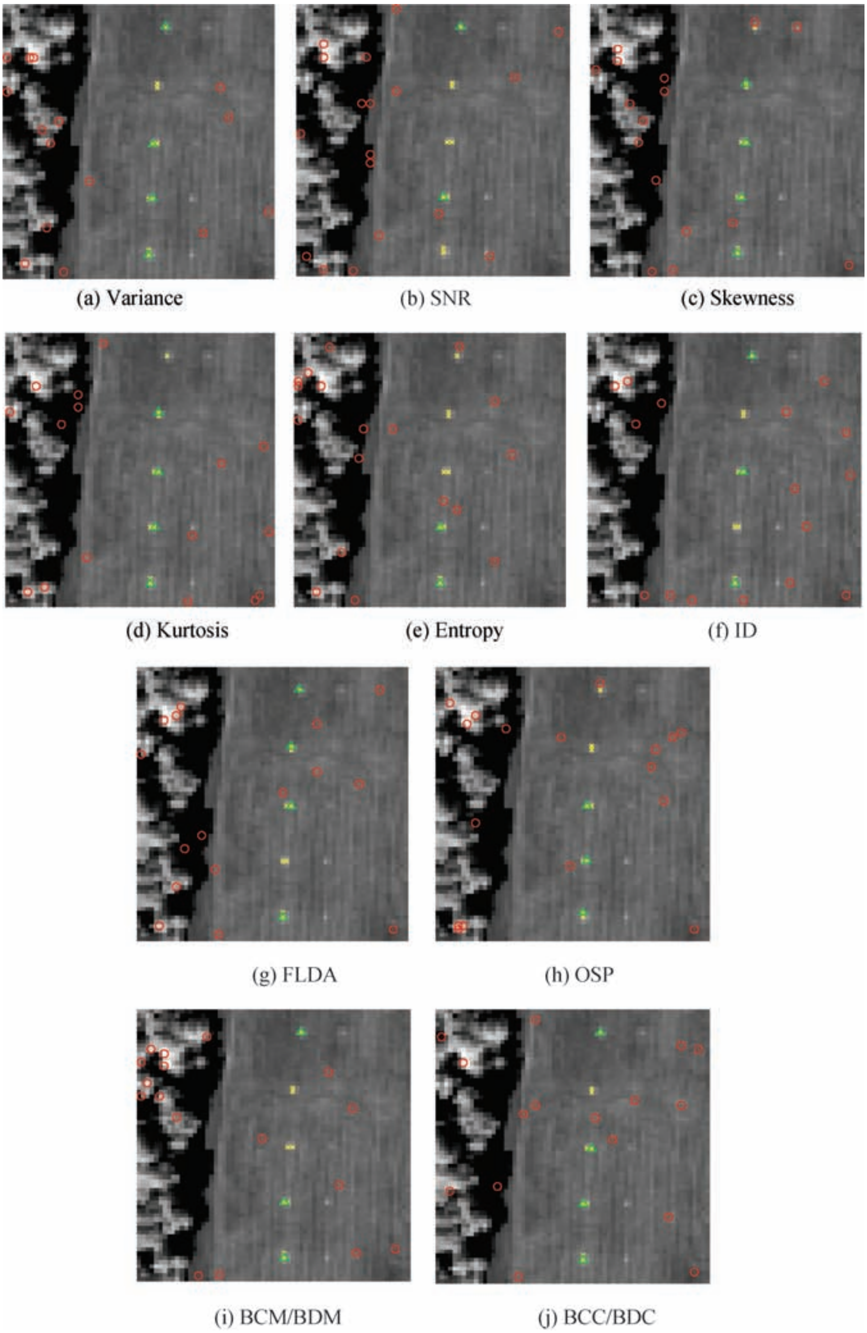
(j) BCC/BDC

Figure 21.6 (*Continued*)

VD for this image scene was estimated with various false alarm probabilities in Table 5.6. For our experiments, $n_{\mathrm{VD}} = 22$ is chosen with $P_F = 10^{-4}$ with HFC method used for VD estimation. This number was also used for the same image scene for endmember extraction in Chang et al. (2006).

Following the same experiments conducted for endmember extraction in Sections 21.4.1.2 and 21.4.2.2, Tables 21.7 and 21.8 list the first 40 highest-prioritized bands and first 40 least-prioritized bands, respectively, where 40 was approximately twice the value estimated by VD that is $44 = 2n_{\mathrm{VD}}$.

Tables 21.9 and 21.10 tabulate the IN-FINDR extracted endmembers that correspond to the five mineral signatures using $\tilde{q}$ highest-prioritized bands and $\tilde{q}$ least-prioritized bands for $\tilde{q} = 20$, 30 and 40, respectively, where extracted endmembers, denoted by lowercase letters, "a, b, c, k, m,"


(a) Variance

(b) SNR

(c) Skewness

(d) Kurtosis

(e) Entropy

(f) ID

(g) FLDA

(h) OSP

(i) BCM/BDM

(j) BCC/BDC

Figure 21.7 Endmembers extracted by IN-FINDR using the first 10 highest-prioritized and 10 least-prioritized bands.

Table 21.6 Comparison between 10 BP criteria by using three different band selection methods, which are 20 highest- 20 least-, and 10 highest + 10 least prioritized bands

	20 highest-prioritized bands	20 least-prioritized bands	10 highest + 10 least prioritized bands
Variance	p_{521}	$p_{11}, p_{411}, p_{412}, p_{521},$	$p_{11}, p_{311}, p_{412}, p_{521},$
SNR	$p_{11}, p_{221}, p_{412}, p_{521}, p_{52},$	p_{221}, p_{521}	$p_{11}, p_{411},$
Skewness	$p_{11}, p_{221}, p_{312}, p_{411}, p_{521},$	p_{11}, p_{311}, p_{411}	$p_{211}, p_{312}, p_{412}, p_{521},$
Kurtosis	$p_{11}, p_{412}, p_{52},$	$p_{211}, p_{312}, p_{521},$	$p_{211}, p_{312}, p_{412}, p_{521},$
Entropy	$p_{312}, p_{521},$	$p_{11}, p_{411}, p_{521},$	$p_{411}, p_{521},$
ID	$p_{11}, p_{211}, p_{412}, p_{521}$	$p_{11}, p_{221}, p_{311}, p_{312}, p_{521},$	$p_{11}, p_{312}, p_{521},$
FLDA	$p_{211}, p_{312}, p_{521},$	$p_{11}, p_{211}, p_{412}, p_{521},$	$p_{11}, p_{211}, p_{312}, p_{521},$
OSP	$p_{211}, p_{312}, p_{521},$	$p_{11}, p_{22}, p_{411}, p_{412}, p_{521}$	$p_{311}, p_{412}, p_{511}$
BCM/BDM	$p_{11}, p_{22}, p_{411}, p_{52}$	$p_{11}, p_{412}, p_{521},$	p_{11}, p_{411}, p_{521}
BCC/BDC	$p_{11}, p_{221}, p_{411}, p_{412}, p_{521}$	$p_{11}, p_{211}, p_{312}, p_{411}, p_{521}$	$p_{11}, p_{312}, p_{411}, p_{521},$

are identified by comparing their signatures against the signatures of the ground truth endmember pixels, upper case letters, "A, B, C, K, M" in Figure 1.12(b) by spectral similarity distance, SAM.

It should be noted that the IN-FINDR-found pixels were generally not the same ground truth pixels in Figure 1.12(b). Instead, they were located in different areas, but have very close signatures in the sense of SAM to those of the five mineral pixels identified in Figure 1.12(b)

As shown in Tables 21.9 and 21.10, the greater the value of $\tilde{q}$ was, the better the endmember extraction for BP criteria were. Nevertheless, $\tilde{q} = 20$ provided a very good estimate for the value of $\tilde{q}$ for IN-FINDR to extract all the five mineral signatures where the three BP criteria, variance, and BCM/BDM, and BCC/BDC in Table 21.9 and the four BP criteria, kurtosis, ID, BCM/BDM, and BCC/BDC in Table 21.10 were able to pull out pixels that corresponded to all the five distinct mineral signatures. In this case, it seemed that the $n_{\text{VD}} = 22$ was a good estimate for value of $\tilde{q}$ instead of twice VD-estimated value, $2n_{\text{VD}} = 18$ used in previous HYDICE experiments. If we further compare Table 21.10 to Table 21.9, it is found that for this particular cuprite image scene using the bands with the first 20 least priority scores to perform endmember extraction was more effective than using the bands with the first 20 highest priority scores.

Finally, we implemented a mixture of the first 10 highest-prioritized bands and first 10 least-prioritized bands for endmember extraction in the same fashion as that conducted in Section 21.4.3.2. Table 21.11 tabulates the IN-FINDR extracted endmembers that correspond to the five mineral signatures.

Comparing Table 21.11 to Tables 21.9 and 21.10, it was obvious that mixing the first 10 highest-prioritized bands and first 10 least-prioritized bands did not perform as well as the first 20 highest-prioritized bands or the first 20 least-prioritized bands alone did, a similar conclusion that was also observed for HYDICE experiments in Section 21.4.3.2.

Several new findings from the above experimental results are particularly interesting. One is that according to our extensive experience, BCM/BDM and BCC/BDC were generally better criteria than any other BP criteria in most applications. Another is that bands with the lower-priority scores had been shown to be more important than bands with the higher-priority scores in some applications such as endmember extraction, detection, and classification of small targets or anomalies. A third one is that as demonstrated in Chang (2006) and Chang and Wang (2006), no matter what BP criterion

Table 21.7 40 Highest-prioritized bands progressively selected by various 10 BP criteria

	40 highest priority scores
Variance	(First 10 bands) 87/85/88/86/89/84/91/80/78/90
	(Second 10 bands) 92/83/82/79/93/81/98/99/97/189
	(Third 10 bands) 77/76/75/94/100/74/96/95/73/72
	(Fourth 10 bands) 101/71/70/69/68/67/123/124/122/121
SNR	(First 10 bands) 68/71/67/69/70/72/66/73/65/26
	(Second 10 bands) 74/25/27/23/24/28/22/64/89/21
	(Third 10 bands) 75/63/87/88/85/86/20/91/19/18
	(Fourth 10 bands) 83/84/42/29/53/90/76/52/39/17
Skewness	(First 10 bands) 2/1/3/4/5/6/7/8/9/10
	(Second 10 bands) 11/12/13/14/184/15/185/183/181/16
	(Third 10 bands) 179/178/180/182/17/177/170/171/18/19
	(Fourth 10 bands) 159/169/20/158/176/157/21/22/23/186
Kurtosis	(First 10 bands) 2/3/1/4/5/6/7/8/9/185
	(Second 10 bands) 181/183/184/10/179/178/180/170/182/177
	(Third 10 bands) 11/169/171/12/176/13/172/14/159/158/
	(Fourth 10 bands) 168/157/156/175/155/15/154/160/167/173
Entropy	(First 10 bands) 87/82/88/85/76/83/89/77/80/101
	(Second 10 bands) 86/100/74/98/81/99/71/84/75/91
	(Third 10 bands) 78/70/79/69/73/92/72/68/90/110
	(Fourth 10 bands) 62/63/64/130/111/61/129/58/109/57
ID	(First 10 bands) 6/2/124/117/11/123/12/96/125/9
	(Second 10 bands) 13/15/10/126/122/14/7/16/5/165
	(Third 10 bands) 112/1/8/4/167/3/164/166/114/163
	(Fourth 10 bands) 162/19/17/18/94/95/25/161/119/113
BCM/BDM	(First 10 bands) 26/117/48/37/189/64/1/185/10/172
	(Second 10 bands) 47/4/60/28/165/17/5/2/151/158
	(Third 10 bands) 3/94/29/9/7/6/8/58/13/11
	(Fourth 10 bands) 91/137/14/153/123/174/12/177/160/167
BCC/BDC	(First 10 bands) 185/37/2/3/5/64/8/9/6/7
	(Second 10 bands) 10/165/4/11/12/14/151/13/28/15
	(Third 10 bands) 16/153/17/19/18/29/184/20/177/47
	(Fourth 10 bands) 21/22/188/25/24/23/167/186/30/31

was used the number of spectral bands, $\tilde{q}$, required for BS was generally higher than the number of spectral dimensions required to be retained after DR in order for BS to accomplish the same task as DR did. In this case, VD produces a reliable estimate for DR as shown in Wang and Chang (2006b) and Chang et al. (2006), while providing a lower bound on BS as witnessed in our experiments.

21.5 Progressive Band Dimensionality Process

The introduction of BP is to set a stage for progressive band dimensionality process (PBDP) to perform PBDE and PBDR. Following the treatment given Section 20.4 similar definitions to (20.12)–(20.14) can be also derived as follows.

Assume that $\{\rho_l^{\text{BPC}}\}_{l=1}^{L}$ is a set of priority scores calculated by a BPC where ρ_j^{BPC} is a priority score assigned to the jth band, $\mathbf{B}_j$. Then for $j, k \in \{1, 2, \ldots, L\}$

$$\rho_j^{\text{BPC}} > \rho_k^{\text{BPC}} \Leftrightarrow \text{priority}_{\text{BPC}}(\mathbf{B}_j) > \text{priority}_{\text{BPC}}(\mathbf{B}_k) \tag{21.40}$$

Table 21.8 40 Least-prioritized bands progressively selected by various 10 BP criteria

	40 Least priority scores
Variance	(First 10 bands) 2/1/3/4/5/6/185/184/7/183
	(Second 10 bands) 8/181/182/180/186/179/9/178/10/177
	(Third 10 bands) 176/169/11/160/170/171/175/159/168/167
	(Fourth 10 bands) 161/158/172/12/174/173/166/165/162/164
SNR	(First 10 bands) 187/189/188/186/185/183/182/184/180/170
	(Second 10 bands) 1/181/139/177/179/176/175/178/102/173
	(Third 10 bands) 138/174/140/172/171/159/169/168/2/158/
	(Fourth 10 bands) 167/164/165/136/161/160/166/156/162/157
Skewness	(First 10 bands) 135/129/128/130/127/131/132/126/125/124
	(Second 10 bands) 133/134/123/122/121/120/119/118/117/116
	(Third 10 bands) 115/114/113/139/112/111/110/109/138/108
	(Fourth 10 bands) 140/107/136/137/141/102/103/189/106/104
Kurtosis	(First 10 bands) 189/123/122/121/124/120/119/125/118/101
	(Second 10 bands) 126/117/116/127/115/100/99/114/98/128/
	(Third 10 bands) 113/97/112/96/95/129/187/92/94/91/
	(Fourth 10 bands) 111/93/90/89/110/88/130/87/86/85
Entropy	(First 10 bands) 2/1/4/3/164/165/5/163/6/166
	(Second 10 bands) 7/8/9/167/10/123/112/11/12/162
	(Third 10 bands) 13/14/16/96/15/161/19/168/185/18
	(Fourth 10 bands) 124/17/95/183/184/125/117/126/177/180
ID	(First 10 bands) 100/99/87/109/88/108/110/86/85/98
	(Second 10 bands) 84/83/92/130/89/131/132/133/134/107
	(Third 10 bands) 135/76/82/75/74/129/101/77/91/174
	(Fourth 10 bands) 111/138/175/141/80/136/140/176/173/71
BCM/BDM	(First 10 bands) 122/125/121/127/129/118/120/134/132/128
	(Second 10 bands) 131/130/101/115/110/92/126/99/93/100
	(Third 10 bands) 98/109/82/88/108/97/116/144/86/84
	(Fourth 10 bands) 96/90/83/106/102/89/103/107/105/80
BCC/BDC	(First 10 bands) 26/189/48/117/158/80/124/1/122/125
	(Second 10 bands) 127/183/121/128/129/120/98/70/134/119
	(Third 10 bands) 131/126/118/140/130/132/115/111/79/135
	(Fourth 10 bands) 116/141/113/100/123/101/96/110/95/99

where $\{\text{priority}_{\text{BPC}}(\mathbf{B}_l)\}_{l=1}^{L}$ determines how the L spectral bands, $\{\mathbf{B}_l\}_{l=1}^{L}$ are selected by PBDP progressively. If $\{\text{priority}_{\text{BPC}}(\mathbf{B}_l)\}_{l=1}^{L}$ is arranged in descending order according to their priorities, $\text{priority}_{\text{BPC}}(\mathbf{B}_{l_1}) \geq \text{priority}_{\text{BPC}}(\mathbf{B}_{l_2}) \geq \cdots \geq \text{prioirty}_{\text{BPC}}(\mathbf{B}_{l_L})$ where

$$l_1 = \arg\{\max\nolimits_l \rho_l^{\text{BPC}}\} l_2 = \arg\{\max\nolimits_{l \neq l_1} \rho_l^{\text{BPC}}\}, \ldots, \text{ and } l_L = \arg\{\min\nolimits_l \rho_l^{\text{BPC}}\} \tag{21.41}$$

and $\{l_1, l_2, \ldots, l_L\}$ is a simply permutation of $\{1, 2, \ldots, L\}$. Then the rank of all the L spectral bands, $\{\mathbf{B}_l\}_{l=1}^{L}$ is arranged by $\text{rank}(\mathbf{B}_{l_1}) = 1, \text{rank}(\mathbf{B}_{l_2}) = 2, \ldots, \text{rank}(\mathbf{B}_{l_L}) = L$. That is, $\text{rank}_{\text{BPC}}(\mathbf{B}_l) \in \{1, 2, \ldots, L\}$ and

$$\text{priority}_{\text{BPC}}(\mathbf{B}_j) > \text{priority}_{\text{BPC}}(\mathbf{B}_k) \Leftrightarrow \text{rank}_{\text{BPC}}(\mathbf{B}_j) < \text{rank}_{\text{BPC}}(\mathbf{B}_k) \tag{21.42}$$

where the smaller number the rank($\mathbf{B}_l$), the higher priority the $\mathbf{B}_l$. It is worth noting that for a generic representation there is no particular BPC specified in ρ_l^{BPC} in (21.40). However, for example, if BPC is specified by skewness, then $\rho_l^{\text{BPC}} = \rho_l^{\text{skewness}}$.

Table 21.9 Endmembers extracted by IN-FINDR that corresponded to one of the five ground truth minerals using the first 20, 30, and 40 highest-prioritized bands

BP criteria	Endmembers corresponding to five minerals		
	$\tilde{q} = 20$	$\tilde{q} = 30$	$\tilde{q} = 40$
Variance	a, b, c, k, m	b, c, k, m	a, b, c, k
SNR	a, b, c	a, c, m, k	a, b, c, k, m
Skewness	a, b, c, m	a, c, m	a, c, m
Kurtosis	a, b, c, k	b, c, k, m	a, b, c, k, m
Entropy	a, b, c, k	b, c, k, m	a, b, c, k, m
ID	a, c	a, b, c, k, m	a, b, c, k, m
BCM/BDM	a, b, c, k, m	a, b, c, k, m	a, b, c, k, m
BCC/BDC	a, b, c, k, m	a, b, c, k, m	a, b, c, k, m

By means of $\{\rho_l^{\text{BPC}}\}_{l=1}^{L}$ or $\{\text{priority}_{\text{BPC}}(\mathbf{B}_l)\}_{l=1}^{L}$ PBDP can be performed by starting any band number and then progressively adding or removing spectral bands specified by their corresponding priority scores $\{\text{priority}_{\text{BPC}}(\mathbf{B}_l)\}_{l=1}^{L}$ in accordance with their assigned ranks by $\{\text{rank}_{\text{BPC}}(\mathbf{B}_l)\}_{l=1}^{L}$ specified by (21.42).

21.6 Hyperspectral Compresssion by PBDP

In analogy with PSDP used to design progressive spectral dimensionality reduction (PSDR) via DP and progressive spectral dimensionality expansion (PSDE) via DP in Chapter 20 PBDP is also a backbone of two major dual processes, progressive band dimensionality reduction (PBDR) via BP, and progressive band dimensionality expansion (PBDE) via BP. It has been shown in Chang et al. (2010) that the number of target signature substances of interest in hyperspectral imagery was generally estimated between n_{VD} and $2n_{\text{VD}}$. So, if we use one spectral band to accommodate a specific material substance, then the number of bands required to be selected must be equal to or greater than the number of signature substances, which is determined by VD. So, these two numbers, n_{VD} and $2n_{\text{VD}}$ can be used to provide a reasonable lower bound and an upper bound on the value of $\tilde{q}$ for PBDE and PBDR, respectively.

Table 21.10 Endmembers extracted by IN-FINDR that corresponded to one of the five ground truth minerals using the first 20, 30, and 40 least-prioritized bands

BP criteria	Endmembers corresponding to five minerals		
	$\tilde{q} = 20$	$\tilde{q} = 30$	$\tilde{q} = 40$
Variance	a, c, k, m	a, b, c, k, m	a, c, m, k
SNR	a, b, c, k	a, c, m, k	a, b, c, m
Skewness	a, b, c	a, b, k, m	a, b, c, k, m
Kurtosis	a, b, c, k, m	a, c, k, m	a, b, c, k, m
Entropy	a, b, c, m	a, b, c, k, m	a, b, c, k, m
ID	a, b, c, k, m	a, b, c, k, m	a, b, c, k, m
BCM/BDM	a, b, c, k, m	a, b, c, k, m	a, b, c, k, m
BCC/BDC	a, b, c, k, m	a, b, c, k, m	a, b, c, k, m

Table 21.11 Endmember extracted by the mixture of the first 10 highest-prioritized and 10 least-prioritized bands

BP criteria	Endmembers corresponding to five minerals
Variance	a, c, m
SNR	a, b, c
Skewness	a, b, c, m
Kurtosis	a, c, k, m
Entropy	c, k, m
ID	a, b, c, k
BCM/BDM	a, b, c, k, m
BCC/BDC	a, b, c, k, m

21.6.1 Progressive Band Dimensionality Reduction Via BP

PBDR is a process that allows users to reduce a large number of spectral bands by removing certain spectral bands with low priorities. It starts with a maximal number of spectral bands and begins to eliminate a number of spectral bands with lowest priorities from the currently processed band set until the performance of data processing is not satisfied or it reaches the minimal number of spectral bands, which can be determined by VD. By implementing PBDR hyperspectral information compression can be achieved by gradually losing information of those spectral bands being removed from consideration. In what follows, we describe its implementation in detail.

PBDR:

1. Prioritize all spectral bands via a BP criterion.
2. *Initialization*: Use VD to determine the minimal number of spectral bands required to be retained, denoted by n_{VD}. Let n_{initial} be the number of spectral bands to begin with for the process and n_Δ be the step size of bands to reduce. Set $k = 0$.
3. Start off spectral bands with $\{\text{priority}_{\text{BPC}}(\mathbf{B}_l)\}_{l=1}^{n_{\text{initial}}}$ where BPC is a generic notation for a BP criterion to be specified.
4. Evaluate the performance of data processing to see if it is satisfied. If it is not, the algorithm is terminated. Otherwise, let $k \leftarrow k + 1$ and continue.
5. At the kth reduction, eliminate the next least-prioritized spectral bands from currently being processed spectral bands so that the resultant number of spectral bands to be processed is reduced to $n_{\text{initial}} - k \cdot n_\Delta$. In this case, the total number of spectral components to be processed is reduced from $n_{\text{initial}} - (k-1)n_\Delta$ to $n_{\text{initial}} - k \cdot n_\Delta$.
6. If $(n_{\text{initial}} - k \cdot n_\Delta) > n_{\text{VD}}$, go to step 5. Otherwise, the algorithm is terminated.

Three notes on PBDR can be summarized below.

a. One is the n_{inital} used to start PBDR. The upper bound on n_{initial} is the total number of spectral bands. However, in many applications, it is usually sufficient by setting $n_{\text{initial}} = 2n_{\text{VD}}$.

b. In step 5 of PBDR, the number of spectral bands eliminated at each iteration is n_Δ, which can be arbitrary. For a simple case, we can set $n_\Delta = 1$. So, the number of added spectral bands will be increased from $n_{\text{initial}} - k$ to $n_{\text{initial}} - (k+1)$.

c. The stopping rule provided in step 6 is only for a general guideline and is not necessarily optimal. It can be replaced with other rules when there is a better one.

21.6.2 Progressive Band Dimensionality Expansion Via BP

As a complete opposite to PBDR, PBDE performs reversely in the way that PBDR does. The process is terminated only if the performance of data processing is satisfied or it reaches the maximal number of spectral components. With PBDE the information available to data processing is gradually increased band-by-band by including new information provided by added spectral bands in descending priorities, in which case less hyperspectral information compression is achieved. A detailed implementation of PBDE is summarized as follows.

PBDE:

1. Prioritize all the spectral bands via a BP criterion.
2. *Initialization*: Use VD to determine an initial number of spectral bands needed to be started with, denoted by n_{VD}. Let n_{final} be the final number of spectral bands to terminate the process and n_{Δ} be the step size of bands to expand. Set $k = 0$.
3. Start off the first n_{VD} prioritized spectral bands with $\{\mathrm{priority}_{\mathrm{BPC}}(\mathbf{B}_l)\}_{l=1}^{n_{\mathrm{VD}}}$ where BPC is a generic notation for a BP criterion to be specified.
4. Evaluate the performance of data processing to see if it is satisfied. If it is, the algorithm is terminated. Otherwise, let $k \leftarrow k + 1$ and continue.
5. At the kth expansion, add the next highest-prioritized spectral bands. In this case, the total number of spectral bands to be processed is increased from $n_{\mathrm{VD}} + (k-1) \cdot n_{\Delta}$ to $n_{\mathrm{VD}} + k \cdot n_{\Delta}$.
6. If $(n_{\mathrm{VD}} + k \cdot n_{\Delta}) < n_{\mathrm{final}}$, go to step 5. Otherwise, the algorithm is terminated.

Three notes on PBDE similar to those of PBDR described at the end of Section 21.4.1 are also worth mentioning.

a. One is the n_{final} used to terminate to PBDE. The upper bound on n_{final} is the total number of spectral bands. However, in many applications, it is usually sufficient by setting $n_{\mathrm{T}} = 2n_{\mathrm{VD}}$.
b. In step 2 of PBDE, n_{Δ} is the step size of bands, which denotes the number of spectral bands to be added at a time in each iteration in step 5. Generally, n_{Δ} can be set 1 or 2.
c. The stopping rule provided in step 6 is only for a general guideline and is not necessarily optimal. It can be replaced with other rules when there is a better one.

It should be pointed out that the range estimated by VD, $[n_{\mathrm{VD}}, 2n_{\mathrm{VD}}]$ only provides a feasible stopping rule for both PBDR and PBDE. In order to obtain the optimal solution, the stopping rule must be determined by separate and more specific applications. For instance, experiments conducted in Section 21.4.3.1 provided good examples where the stopping rule can be determined by a specific application performance, LSMA.

A similar comment made at the end of Section 20.5 in Chapter 20 on the pair of (PSDR, PSDE) in comparison with the pair of the sequential forward selection (SFS) and sequential backward selection (SBS) developed for feature selection in (Serpico and Bruzzone, 2001) is also applied to the pair (PBDR, PBDE) where the information contained in a spectral band can be considered as a feature vector. So, PBDE can be carried out as a sequential forward process that increases spectral information by adding more spectral bands, while PBDR can be implemented as a sequential backward process, which is a reverse process of PBDE and reduces spectral information by eliminating more spectral bands. The only difference between the pair of (PBDR, PBDE) and the pair of (SBS, SFS) is that the latter pair does not prioritize features for selection.

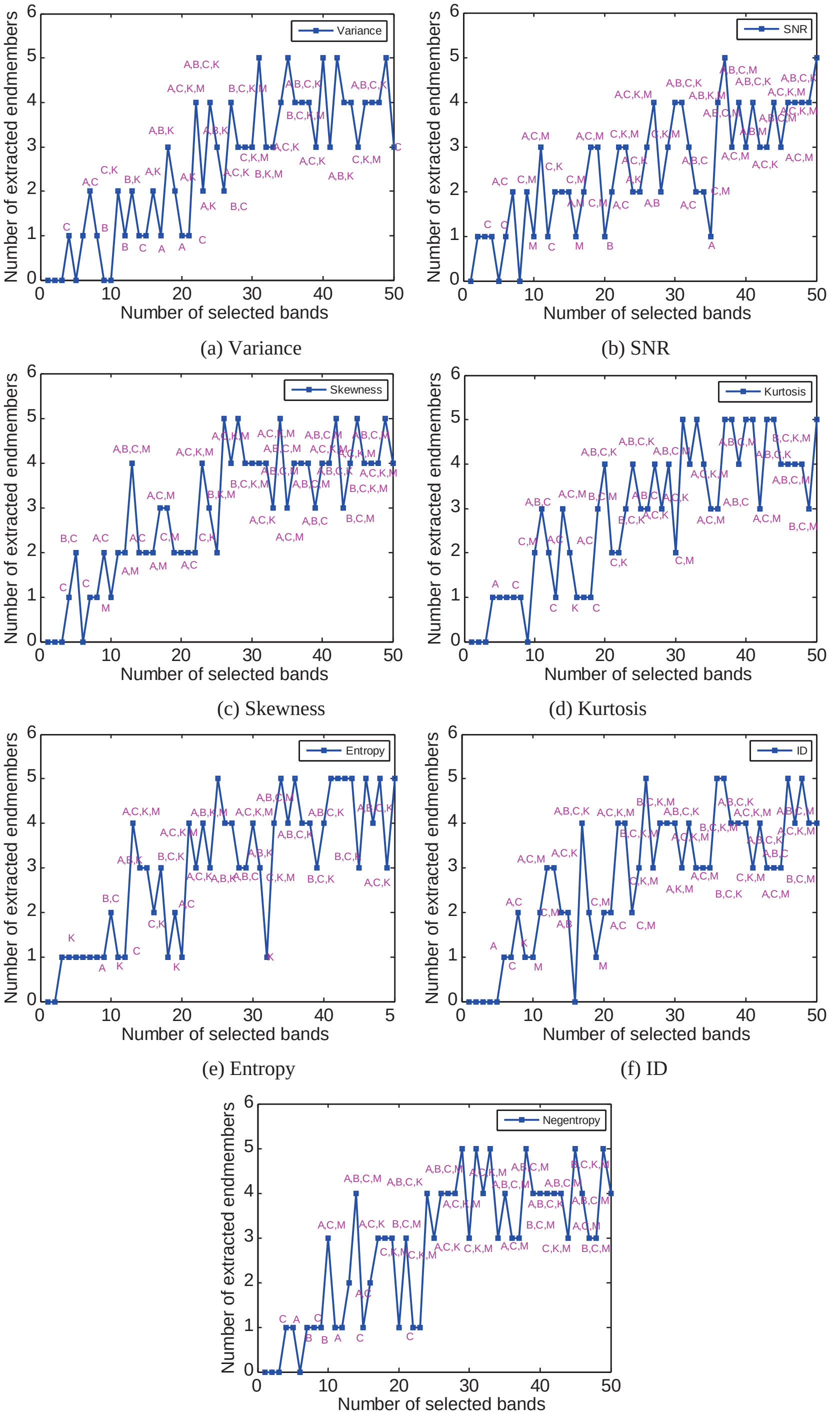
Finally, in order to make a further distinction between the pair of (PBDR, PBDE) and the pair of (SBS, SFS), one remark on the difference between "progressive" process and "sequential" process is noteworthy. A progressive process is a process that uses previous results to gradually improve or reduce performance. A sequential process is a process that carries out data sample vectors one after another in sequence where each data sample vector is fully processed. A simple example should serve the purpose of illustration. Assume that there is an 8-bit image. This image can be represented by using progressive image processing via the bit-plane coding. It is first encoded by the most significant bit and then more bits are added to improve image quality progressively until the least bit is added to completely represent the image. So, in this case, there are eight stages (one stage represented by one bit) to represent images progressively with previous images included as part of images generated in subsequent stages. This progressive process is terminated when it completes eight progressive stages. On the other hand, the image can be represented by using sequential image processing where each image pixel is encoded by using the full bit rate, 8 bits. This sequential process begins with the first pixel, then second pixel, etc. until it reaches the last pixel. So, this process is carried out pixel by pixel and line by line with 8-bit coding. This sequential image process is completed until it reaches the last image pixel. So, in progressive image processing the progressive nature is determined by eight stages where the entire image is improved from low to high resolution. How many stages required to be completed is determined by how fine resolution is needed. In sequential image processing the sequential nature is determined by the number of pixels to be processed. The entire image cannot be completed until all the image pixels are processed. The web image is represented by such a sequential image representation. Using a similar interpretation, PBDP belongs to progressive image processing, which performs band dimensionality expansion and reduction, while SFS and SBS are actually sequential image processing, which finds features sequentially. So, one fundamental difference between PBDE and sequential band dimensionality expansion (SBDE) is that the bands selected by PBDE using a smaller number of bands are always part of the bands selected by PBDE using a larger number of bands, while the bands selected by the SBDE using a smaller number of bands are not necessarily part of bands selected by SBDE using a larger number of bands. This is because SBDE must re-select new bands again once the number of bands is increased. Another fundamental difference is that PBDE adds bands according to priority scores calculated by a band prioritization criterion, while the SBDE must solve a new optimization problem as the number of bands changes and no band prioritization is involved. Similar arguments are also applied to PBDR and sequential band dimensionality reduction (SBDR). The SFS and SBS are considered as such sequential image processes that must re-solve optimization problems every time a feature is removed or added. These selection processes can be made in a progressive manner once these features are found and prioritized as the way it is done by a band prioritization criterion.

21.7 Experiments for PBDP

As a parallel section to Section 20.6 the same experiments conducted for PSDP were also performed for PBDP in this section for comparison where three applications, endmember extraction, land cover/use classification, and spectral unmixing using three different types of hyperspectral image data sets are considered.

21.7.1 Endmember Extraction

The Cuprite data in Figure 1.12(a) were used for endmember extraction where IN-FINDR was selected to extract the five mineral signatures, A, B, C, K, and M of major interest in the scene. Figure 21.8(a)–(g) plots extracted endmembers by IN-FINDR where seven BP criteria, variance,



(a) Variance

(b) SNR

(c) Skewness

(d) Kurtosis

(e) Entropy

(f) ID

(f) Negentropy

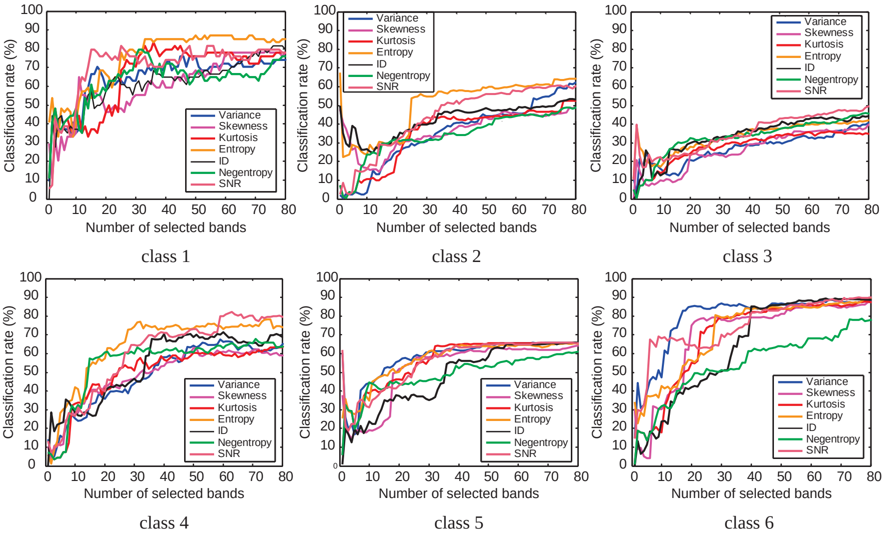
Figure 21.8 Endmember extraction results of PBDP prioritized cuprite scene using different BP criteria.

SNR, skewness, kurtosis, entropy, ID, and neg-entropy were used to prioritize bands to be used by PBDP. VD-estimated value for this scene was $n_{\mathrm{VD}} = 22$ that was the same used for experiments in Chapter 20. Its twice value $2n_{\mathrm{VD}} = 44$ provided an upper bound used by PBDP. Therefore, the x-axis represents the number of prioritized bands used by PBDP starting from 0 to 50, and the y-axis is the number of extracted endmembers. When the number of extracted endmembers is less than 5, the extracted endmembers are specified by particular mineral signatures. As shown in Figure 21.8 the high-order statistics-based BP criteria generally performed better than second-order statistics-based BP criteria such as variance and SNR in the sense that fewer band numbers were required to extract five endmembers. The smallest and largest numbers of bands required to extract all the five mineral signatures were between 25 and 31. Compared to the results produced by PSDP in Figure 20.8, it is obvious that the number of spectral bands required by PBDP was greater than that by PSDP, which was between 18 and 28. This made sense because PSDP performed data compaction as opposed to PBDP, which essentially performed data reduction.

21.7.2 *Land Cover/Use Classification*

The Purdue Indiana Indian Pine Test Site in Figure 1.13 was also used for land cover/use classification. Figure 21.9 shows classification rates of MLC using PSDP based on seven BP criteria: variance colored by dark blue, SNR by light brown, skewness by purple, kurtosis by red, entropy by deep yellow, ID by black, and negentropy by green where the x-axis denotes the number of prioritized bands to be used to form image cubes for MLC classification, while the y-axis indicates MLC classification rates. In order to see how the number of bands affects the classification performance, the number of prioritized dimensions is run from 1 to the entire number of bands, 202.

As shown in Figure 21.9, it is very clear that different classes required different number of spectral bands to achieve their best performance in classification. For instance, class 16 only required a



class 1

class 2

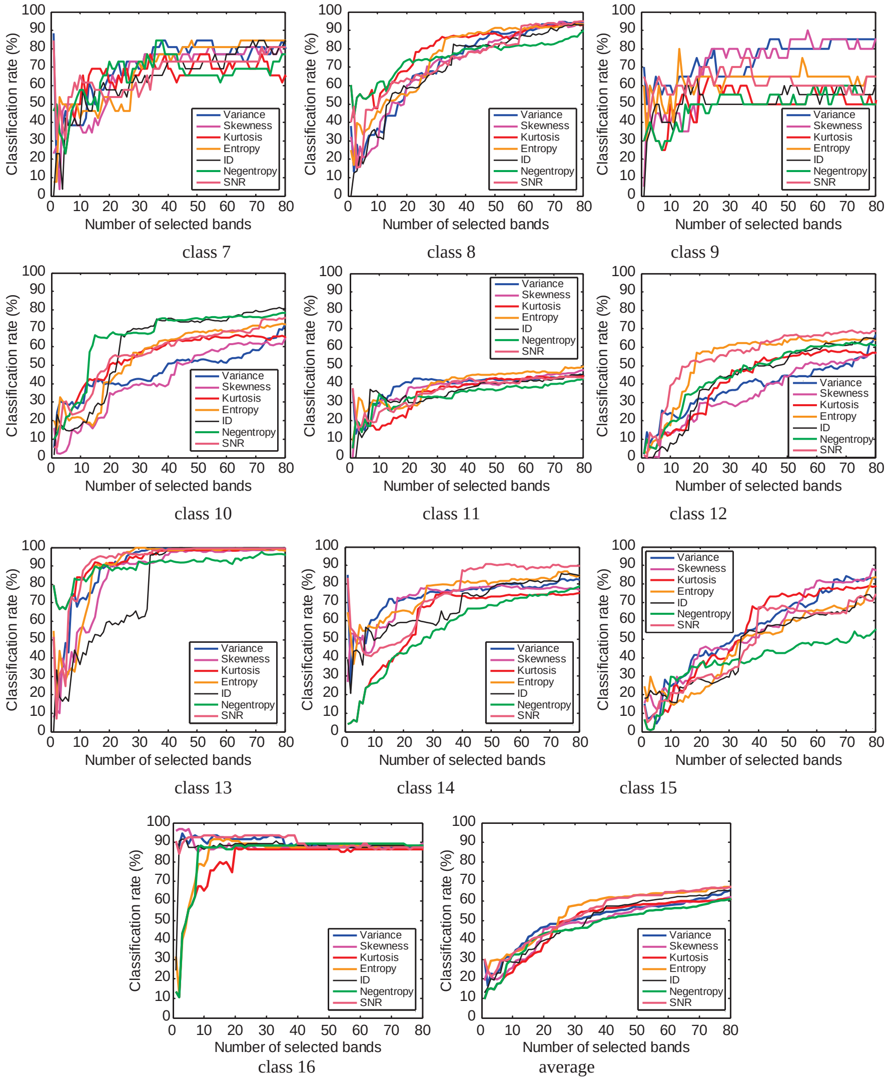
class 3

class 4

class 5

class 6

Figure 21.9 MLC-classification rate of 16 classes versus the number of prioritized bands.



class 7

class 8

class 9

class 10

class 11

class 12

class 13

class 14

class 15

class 16

average

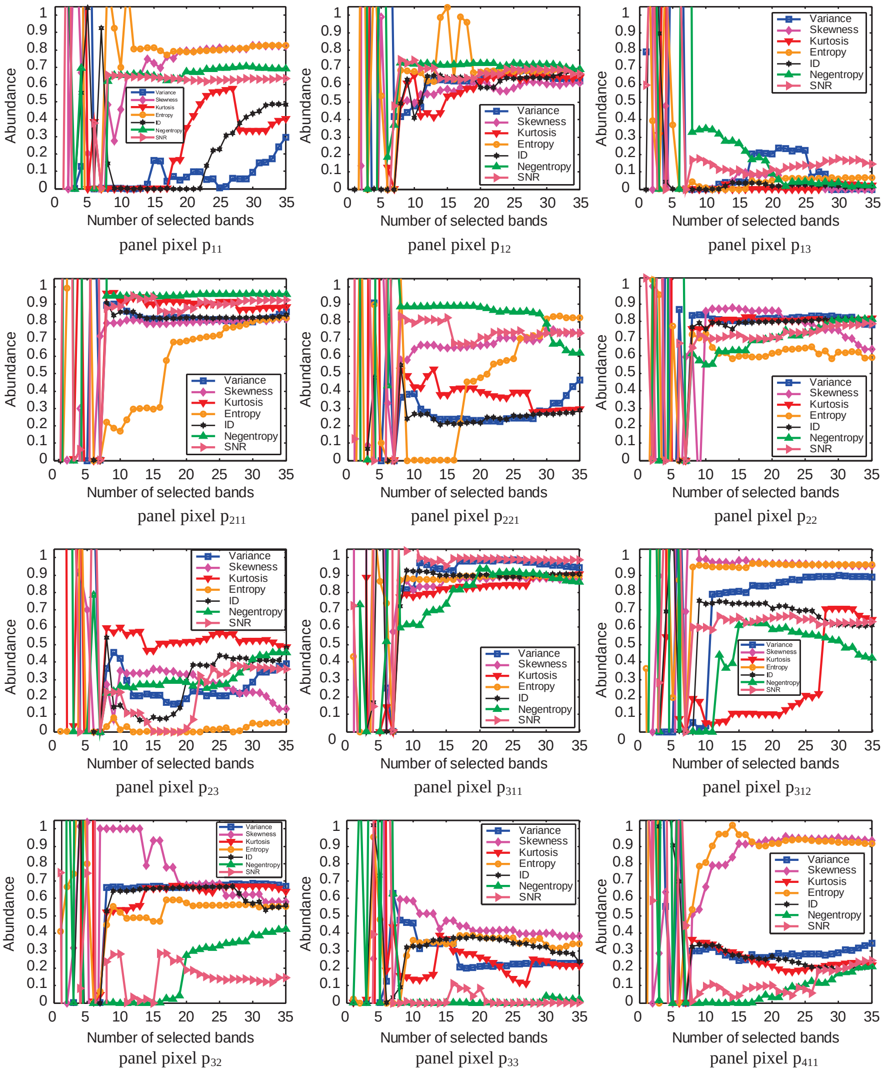
Figure 21.9 (*Continued*)

few bands to achieve very high classification rate. Classes 1, 8, and 13 could also reach very high classification performance as long as $\tilde{q}$ went beyond 50. On the contrary, some of classes were very difficult to achieve such as high classification performance even if the entire 202 spectral bands were fully used. For example, classes 3 and 11 are considered as most difficult cases. This is probably due to the fact that the size of these two classes is too small or the ground truth information is not very accurate for these classes. As a consequence, the sample vectors in these two classes cannot provide reliable statistics to be used by MLC. From Figure 21.9 another interesting observation is worth noting where the classification of classes 7 and 9 started to slightly deteriorate when $\tilde{q}$ was greater than a certain number for some BP criteria. In particular, as the number of prioritized bands was increased the classification performance of these two classes was actually degraded. Other than these extreme cases we categorize classification performance into two groups. One is the cases that the classification rates never improved and saturated after a certain number of used prioritized bands, for example, classes 5, 13, and 16. Another is that the classification kept improving as the number of used prioritized bands was increased. It is particularly true for most of remaining classes, such as classes 1, 2, 3, 6, 8, 11, 12, and 15.

21.7.3 Linear Spectral Mixture Analysis

The 15-panel HYDICE images scene in Figure 1.15(a) and (b) was used to demonstrate the utility of PBDP in spectral unmxing. According to the ground truth in Figure 1.15(b) and (c) and four identified background signatures in Figure 1.17 there are at least nine signatures present in the scene that can be used to form a nine-signature matrix for a linear mixing model to perform spectral unmixing where FCLS was used to unmix all data sample vectors into their corresponding abundance fractions. Six BP criteria: variance, skewness, kurtosis, entropy, ID, and neg-entropy were used to evaluate the unmixing performance of PBDR and PBDE with the step size of band numbers set to one, that is, step size $n_\Delta = 1$. Since VD provides a good estimate for a lower bound to n_{initial} for PBDE and an upper bound to n_{final} for PBDR, n_{initial} is set to $n_{\text{VD}} = 9$ for PBDE and $2n_{\text{VD}} = 18$ for PBDR, respectively. From the scene there are five distinct panel signatures present in the field. If one band is required to accommodate one signature, it needs at least five bands to perform spectral unmixing in which case 5 provides a lower bound for $n_{\text{initial}} = 9$ used by PBDE. On the other hand, it has been shown in Chang (2003a, Chapter 5) and Heinz and Chang (2001) that 34 target signatures were sufficient for FCLS to work well in spectral unmxing. In this case, a sufficient number of bands to accommodate 34 target signatures was 34 that was used as an upper bound on $n_{\text{final}} = 18$ used by PBDR. Figure 21.10 plots six curves to represent the FCLS-unmixed abundance fractions of the 19 R pixels in the five rows in the range of [0,1] using the number of prioritized bands, $\tilde{q}$ ranging from 8 to 35, respectively, with the x-axis representing the number of prioritized bands to be used to form image cubes for FCLS versus y-axis indicating the FCLS-unmixed abundance fractions of 19 R panel pixels where six BP criteria: variance colored by dark blue-square, skewness by green-diamond, kurtosis by red-upside down triangle, entropy by cyan-open circle, ID by black-asterisk, and neg-entropy by purple-star were implemented. It is worth noting that when the number of bands, $\tilde{q}$ was smaller than 8, the results were fluctuated and not reliable as shown in the plots.

As shown in Figure 21.10, each panel pixel required a different value of $\tilde{q}$ to achieve its optimal performance in terms of its FCLS-unmixed abundance fraction compared to its true abundance in Figure 1.15(b). Since in the plot of the panel pixel p_{521} the five curves by generated by variance (blue), skewness (green), kurtosis (red), ID (black), and negentropy (magenta) were identical, only two curves are shown in the plot, one for the five identical curves and the other for entropy (cyan). The quantification results in Figure 21.10 provided evidence that an optimal set of selected prioritized bands for one panel pixel was not necessarily also optimal for another panel pixel. For example, from visual inspection of the plots in Figure 21.10 the 19 panel pixels can be divided into three groups according to unmixing performance using various values of $\tilde{q}$ to select bands. The first group including panel pixels, p_{12} (except entropy), p_{211}, p_{22}, p_{311}, p_{43}, p_{511}, p_{521}, p_{52}, p_{53} required fairly stable values of $\tilde{q}$ to produce good FCLS-unmixed abundance fractions. The second group including panel pixels, p_{13}, p_{32}, p_{33}, p_{411}, p_{42} needed moderately stable values of $\tilde{q}$ to produce good FCLS-unmixed abundance fractions. The third group including panel pixels, p_{11}, p_{221}, p_{23}, p_{312}, p_{412} used fluctuated values of $\tilde{q}$ to produce good FCLS-unmixed abundance fractions. Interestingly, the three panel pixels p_{11}, p_{12}, and p_{13} in row 1 happened to the case that covers all the three groups while all the four panel pixels, p_{511}, p_{521}, p_{52}, p_{53} in row 5 falling in the same first



panel pixel p_{11}

panel pixel p_{12}

panel pixel p_{13}

panel pixel p_{211}

panel pixel p_{221}

panel pixel p_{22}

panel pixel p_{23}

panel pixel p_{311}

panel pixel p_{312}

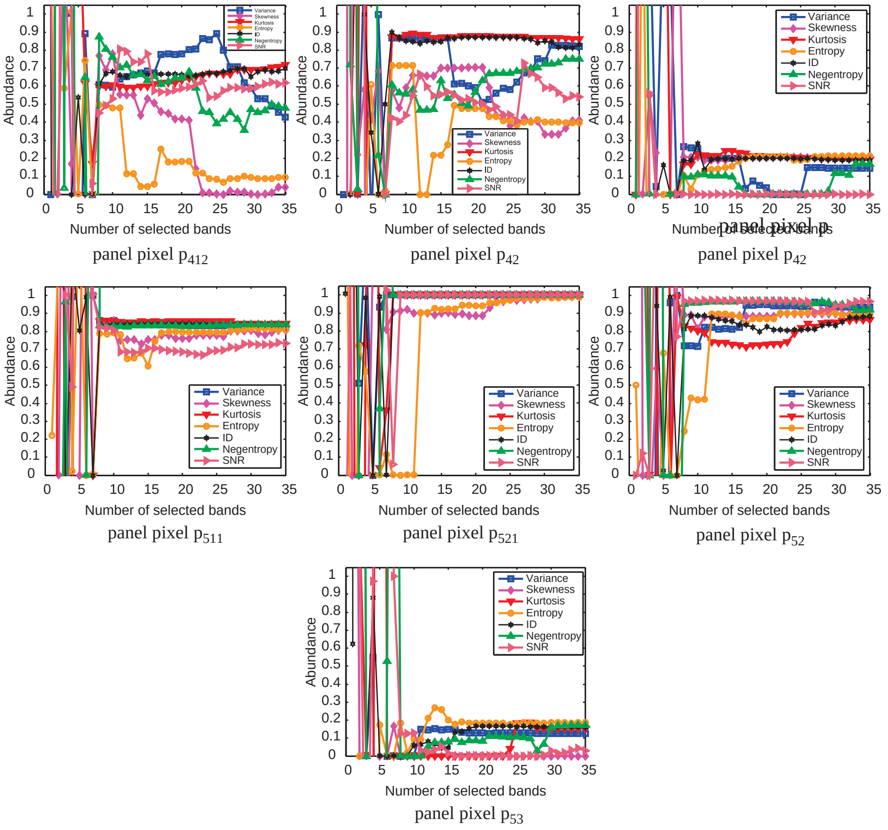
panel pixel p_{32}

panel pixel p_{33}

panel pixel p_{411}

Figure 21.10 Plots of FCLS-unmixed abundance fractions versus number of prioritized bands for 19 red pixels in five row panels in HYDICE image by variance, skewness, kurtosis, entropy, ID, and negentropy.

group. Accordingly, using a fixed number of spectral bands to perform the conventional BS could not accomplish what PBDP could as demonstrated in Figure 21.10. However, a stopping rule for PBDP could be further determined by FCLS performance by comparing FCLS-unmixed results from using two consecutive band sets. Using the panel pixel p_{511} as an example, PBDE would be terminated at $\tilde{q} = 8$ if $n_{\text{inital}} = 5$ or 9 if $n_{\text{inital}} = 9$ when both results were very close. By contrast, PBDR would be terminated at $\tilde{q} = 18$ if $n_{\text{final}} = 18$ or 35 if $n_{\text{final}} = 35$. So, choosing the values of n_{inital} and n_{final} was important. Nevertheless, the conducted experiments seemed to demonstrate that the range provided by VD, $[n_{\text{VD}}, 2n_{\text{VD}}]$ may not be optimal but indeed offered a feasible region for users by setting $[n_{\text{initial}}, n_{\text{final}}]$ to $[n_{\text{VD}}, 2n_{\text{VD}}]$. This issue will be further discussed in Chapter 22 by introducing a new concept of DDA.



panel pixel p_{412}

panel pixel p_{42}

panel pixel p_{42}

panel pixel p_{511}

panel pixel p_{521}

panel pixel p_{52}

panel pixel p_{53}

Figure 21.10 (*Continued*)

21.8 Conclusions

The concept of BP was previously investigated by Chang et al. (1999) to be part of band selection that prioritizes spectral bands in accordance with their priority scores calculated by a specific criterion designed for a particular application. The term of BP was coined and further explored for PBDP in Chang et al. (2010) where the two dual processes, forward PBDP and backward PBDP, were re-named as PBDE and PBDR, respectively, both of which can perform band selection without actually solving an optimization problem required by the conventional BS. Several potentials of PBDP in various applications have been explored in this chapter. (1) It extends second-order statistics band prioritization criteria to including high-order statistics band prioritization criteria where four categories of criteria are derived for BP based on second-order statistics, high-order statistics, classification, and band correlation/dependence minimization, respectively. Interestingly, such categorization has not been studied in the literature. (2) By virtue of PBDP two dual processes can be developed, namely, PBDR via BP and PBDE via BP that can be used for applications in data compression, storage, transmission, and communication. (3) PBDP takes advantage of a recently developed VD to provide lower or upper bounds on the number of bands to be selected. (4) In some applications where the small targets can only show up in bands with least priorities. In this case, PBDP enables users to perform band selection from bands with least priorities instead of highest priorities where this concept was never investigated in the past. (5) PBDP can be further extended to progressive band selection by including a process , called band de-correlation (BD), which can remove interband correlation among prioritized bands. (6) Finally, PBDP does not require knowing the number of bands needed to be selected as required by the conventional BS. The bands can be selected according to their calculated priority scores progressively to improve performance where VD can be used as an initial guess. This cannot be done by the conventional BS if the number of bands is changed in which case BS needs to be re-processed again. This advantage is particularly useful for high computing performance in hyperspectral data compression and communication in space-borne platform where reduction of computational complexity becomes imperative. (7) The progressive nature provided by PBDP does not exist in the conventional BS. It allows users to perform progressive band dimensionality reduction, expansion, selection, transmission in data communication, etc. for further image analysis. Obviously, this task cannot be accomplished by the conventional BS since it does not rank all the spectral bands. (8) PBDP allows progressive band selection (PBS) to perform band selection in either way in terms of higher priorities and lower priorities depending on applications as illustrated in our experiments. Despite the fact that only two applications, that is, endmember extraction and linear spectral unmixing are investigated in this chapter for utility of PBDP, other applications such as target detection, anomaly detection, and data compression can also be conducted. It is our belief that PBDP developed in this chapter will find its way in many more applications yet to be explored (Chang, 2013). Since PBDP does not perform inter-band redundancy compression, BD is usually included in PBDP to expand its capability in band selection. So, when PBDP is implemented in conjunction with BD, it is called PBS that will be discussed in Chapter 23.

22

Dynamic Dimensionality Allocation

The progressive spectral dimensionality process (PSDP) in Chapter 20 and progressive band dimensionality process (PBDP) in Chapter 21 are developed to mitigate three major issues arising in both dimensionality reduction (DR) and band selection (BS). One is that the number of dimensions to be retained after DR, q, and the number of bands for BS to select, $\tilde{q}$, must be known *a priori*. Another is that the values of q and $\tilde{q}$ must be fixed once they are determined and cannot be changed during the process. The third one is that when the values of q and $\tilde{q}$ are changed, the entire process of DR or BS must be reimplemented over again and cannot take advantage of results obtained with previous smaller values of q and $\tilde{q}$. Both PSDP and PBDP address the second and third issues by introducing dimensionality prioritization (DP) and band prioritization (BP) and the second issue by bounding q and $\tilde{q}$ from below by n_{VD} and above by $2n_{\mathrm{VD}}$ with n_{VD} being the value determined by virtual dimensionality (VD) defined in Chapter 5, that is, $n_{\mathrm{VD}} \le q,\ \tilde{q} \le 2n_{\mathrm{VD}}$. However, in real world problems the values of q and $\tilde{q}$ generally vary with different applications even when the same data set is used. As a consequence, fixing q and $\tilde{q}$ at constant values may not be practical or realistic. This chapter introduces a new concept, to be called dynamic dimensionality allocation (DDA), that allows users to dynamically adjust the values of q and $\tilde{q}$ to meet various applications.

22.1 Introduction

When DR and BS are performed, it assumes that for a given data set the values of q and $\tilde{q}$ are known and fixed during the process regardless of applications. However, in many practical applications this may not be true. Using linear spectral mixture analysis (LSMA) as an example, let $\{\mathbf{m}_j\}_{j=1}^{p}$ be a set of p signatures used by LSMA to perform spectral unmixing. Due to their own spectral characteristics the spectral discriminatory powers of these p signatures should be determined by their spectral distinctions (see Chapter 2 in Chang (2003a)). Accordingly, each individual signature should also require different values of q and $\tilde{q}$ for its unmixing. This evidence has been witnessed in Figures 20.1, 20.2, 20.6, and 21.8–21.10. To address and resolve this issue, the parameters q and $\tilde{q}$ must be made variables to adapt dynamically instead of being fixed at constants. While such a thought is very desirable, how can it be really done? It is shown in source coding that to achieve optimal coding performance, variable-length coding instead of fixed-length coding must be used due to variable probabilities of source alphabets used for information transmission (Cover and Thomas, 1991). The rationale of designing variable-length optimal codes seems to provide a clue to solving the issue of variable dimensionality encountered in DR and BS. However, two issues

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

must be addressed first, "how do we define source alphabets in hyperspectral data?" and "how do we define their associated probabilities?"

Interestingly, the above two issues can be restated in the context of hyperspectral imaging using the following example. First, we can interpret a hyperspectral data as an information source. Then the endmembers discussed in PART II (Chapters 7–11) can be considered as hyperspectral signatures that correspond to source alphabets. The next task is how to define the probabilities of these endmembers that are similar to source alphabet probabilities. In source coding, the source alphabet probabilities are described by relative frequencies of occurrence among source alphabets. Such source probabilities can be interpreted by the concept of relative spectral discriminatory probabilities recently introduced by Chang (2000) and Chang (2003a, Chapter 2). In other words, the probability of each endmember can be characterized by the power of its signature discriminability. In light of this interpretation the well-established coding theory is readily applied to hyperspectral imaging where finding bit allocations for source alphabets is equivalent to finding dimensionality allocation which specifies optimal values of q and $\tilde{q}$ required for each endmember to perform DR and BS, respectively. This simple exemplary example offers the basic idea of a new concept dynamic dimensionality allocation (DDA) introduced in the following section.

22.2 Dynamic Dimensionality Allocaction

The DDA presented in this section is designed to dynamically determine the values of q and $\tilde{q}$. It originates from the pigeon-hole principle described in Section 1.3.2 as well as variable-length coding from the information theory. According to the pigeon-hole principle each signature is assumed as a pigeon to be accommodated by a particular spectral dimension/band which can be considered as a pigeon-hole. Therefore, the number of pigeons should determine at least how many pigeon-holes required for accommodation. This is equivalent to saying that the number of spectrally distinct signatures determines the minimal number of spectral dimensions/bands required for signature discrimination, which is exactly the original idea of VD. To materialize DDA, we first interpret the use of a pigeon-hole to accommodate a pigeon by a binary bit "1" and "0" otherwise. This implies that a spectral dimension/band being used to specify a particular signature will be encoded by "1." Otherwise, "0" will be assigned to an unused spectral dimension/band. To fit the profile of source coding, the first task is to determine what type of signatures that can be considered as source alphabets. If the signature knowledge is provided *a priori*, this known signatures can be used as desired source alphabets. If there is no prior knowledge available about the data, the signatures should be found in an unsupervised means. In this case, the VD developed in Chapter 5 can be used to estimate the number of spectrally distinct signatures, denoted by n_{VD}. To find n_{VD} unknown signatures, the automatic target generation process (ATGP) developed by Ren and Chang (2003) and discussed in Section 8.5.1 can be used to produce a set of signatures, $\{\mathbf{s}_j\}_{j=1}^{n_{\mathrm{VD}}}$, that correspond to desired source alphabets. Or alternatively, the virtual signatures found in the supervised LSMA in Chapter 17 can also serve as the same purpose. Once these signatures of interest are found, the next task is to calculate their discriminatory probabilities according to the relative spectral discriminatory probability (RSDPB) in Chang (2000) and Chapter 2 in Chang (2003a) that will determine DDA. In doing so, we briefly review the basic concept of source coding that will be used to define DDA.

Assume that an information source S is emitted by a set of source alphabets $\{a_j\}_{j=1}^{J}$ with a given probability distribution $\{p_j\}_{j=1}^{J}$ where p_j is the probability of the occurrence of the source alphabet a_j. To encode the source S these source alphabets must be represented by a set of code words, called code book. Two coding schemes are generally used for finding a code book for source encoding. One is called fixed-length coding which assigns code words with equal length to all the source alphabets $\{a_j\}_{j=1}^{J}$. The other is variable-length coding that assigns code words with variable coding lengths to individual source alphabets according to their occurrence probabilities. Let the coding length used to encode a_j be denoted by l_j. Shannon showed that the optimal coding scheme must be variable-length coding with the mean coding length $\bar{l} = \sum_{j=1}^{J} p_j l_j$ determined and approximated by the source entropy, $-\sum_{j=1}^{J} p_j \log p_j$. The Huffman coding is proved to be the one that achieves the optimal coding performance in terms of Shannon entropy (see details in Section 31.2.1). The only case that a fixed-length coding also achieves the same performance as the Huffman coding does is when the occurrence probabilities of all the source alphabets are equally likely.

Now how can we borrow the idea of the variable-length coding described above to apply to hyperspectral imaging? First, let $\{\mathbf{s}_j\}_{j=1}^{n_S}$ denote the signatures of interest where n_S can be either the number of known signatures or n_{VD} if no prior knowledge is given. Furthermore, let n_j denote the number of spectral dimensions/bands required to represent the jth signature, $\mathbf{s}_j$. Then $\mathbf{s}_j$, n_S, and n_j shall play the same role as a_j, J, and l_j do in source coding to represent the jth source alphabet, the number of source alphabets and number of bits required for a code book to encode the set of source alphabets $\{a_j\}_{j=1}^{J}$ along with their corresponding probabilities $\{p_j\}_{j=1}^{J}$. In other words, the source occurrence probability p_j is now interpreted by n_j that reflects how difficult the $\mathbf{s}_j$ is discriminated from other signatures in terms of spectral similarity in the same way that how frequently the a_j occurs in terms of probabilities relative to other source alphabets used in source coding. As a result, the higher the probability p_j is, the shorter coding length, l_j is. This implies that the easier to be discriminated the signature $\mathbf{s}_j$ is, the smaller number the n_j is. In particular, to best represent the $\mathbf{s}_j$ in terms of spectral dimensions/bands the n_j must vary with discriminatory power possessed by the signature $\mathbf{s}_j$. The traditional DR/BS makes a simple and natural assumption that all the signatures are equally discriminable by setting $n_j = n_S$ for all $\{\mathbf{s}_j\}_{j=1}^{n_S}$ in which case it performs so-called static dimensionality allocation (SDA), that is, fixed-size band dimensionality. Apparently, it is generally not true in hyperspectral data where each material substance signature has its own spectral characteristics and has a different level of signature discriminability. So, what DDA is to the SDA in DR/BS is the same as what is the variable-length coding to fixed-length coding in source coding. Recently, Wang and Chang (2007) have introduced a new concept of variable number variable band selection (VNVBS) to be discussed in Chapter 27 and showed that using variable numbers of spectral bands was more effective than using fixed number of spectral bands. It provided evidence in advantages of using DDA over SDA.

The next remaining issue is how to come up a technique to determine DDA similar to variable-length coding used in source coding where the bit allocation is determined by the coding lengths $\{l_j\}_{j=1}^{J}$ which can be calculated by their associated probabilities. By interpreting signature discriminatory probabilities as source alphabet probabilities we are able to do the same for DDA in such a way that variable numbers of spectral dimensions/bands are assigned to represent various discriminatory powers of signatures. Three commonly used coding schemes, Shannon coding, Huffman coding, and Hamming coding can be developed to find DDA.

22.3 Signature Discriminatory Probabilties

As noted above, the key to materialize the concept of DDA is to find a means of interpreting source alphabet probabilities used in source coding in terms of hyperspectral signatures. Let the entire hyperspectral data be considered as an information source with a set of hyperspectral signatures $\{\mathbf{s}_j\}_{j=1}^{n_S}$ that correspond to source alphabets $\{a_j\}_{j=1}^{J}$. We now interpret relative occurrence frequencies among J source alphabets, $\{a_j\}_{j=1,}^{J}$ as relative spectral discriminatory powers among the n_S signatures, $\{\mathbf{s}_j\}_{j=1}^{n_S}$, then the source alphabet probabilities $\left\{p_j\right\}_{j=1}^{J}$ can be interpreted as signature discriminatory probabilities among $\{\mathbf{s}_j\}_{j=1}^{n_S}$, denoted by $\{\pi_j\}_{j=1}^{n_S}$ which can be obtained as follows.

To begin with, we select a spectral similarity measure, denoted by $m(\cdot,\cdot)$ such as spectral angle mapper (SAM), and spectral information divergence (SID) (Chang, 2003a). Technically, SID may be a better candidate than SAM since it is a criterion designed to measure discrepancy between two probability distributions. However, if SAM is used, the cosine value, $\cos\theta$, will be used instead of the values of angle, θ. Next, we choose a reference signature $\mathbf{s}$ as a benchmark against which each signature of $\{\mathbf{s}_j\}_{j=1}^{n_S}$ will be compared and computed for finding their relative spectral discriminatory probabilities, $m(\mathbf{s}_j, \mathbf{s})$ for all $1 \leq j \leq n_S$. Normalizing by the constant of $\sum_{j=1}^{n_S} m(\mathbf{s}_j, \mathbf{s})$ a probability vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_{n_S})$ can be obtained by

$$\pi_j = \frac{m(\mathbf{s}_j, \mathbf{s})}{\sum_{j=1}^{n_S} m(\mathbf{s}_j, \mathbf{s})} \tag{22.1}$$

that is a probability of difficulty level of discriminating the jth signature $\mathbf{s}_j$ with respect to the reference signature $\mathbf{s}$.

There are three candidates can be used for selection of the reference signature $\mathbf{s}$, data sample mean $\boldsymbol{\mu}$, signature mean $\bar{\boldsymbol{\mu}} = (1/n_S)\sum_{j=1}^{n_S} \mathbf{s}_j$ and any signature from $\{\mathbf{s}_j\}_{j=1}^{n_S}$. Which one is a better choice depends upon applications (Chang et al., 2010; Wang et al., 2010; Wang and Chang, 2007).

22.4 Coding Techniques for Determining DDA

Finding signature discriminatory probabilities only accomplishes half the task. The other half task is to design a technique to allocate DDA required for each of signatures, $\{\mathbf{s}_j\}_{j=1,}^{n_S}$ based on their signature discriminatory probabilities. Suppose that each spectral dimension (i.e., spectral component) or spectral band can be only used to accommodate one and only one signature, then a binary value "1" can be used to indicate whether a spectral dimension or spectral band is being used for signature accommodation, "1" for being used and "0" for remaining "unused". Consequently, DDA can be addressed by bit allocation where the number of spectral dimensions, q and the number of spectral bands, $\tilde{q}$ required for each signature corresponds to the coding length used to encode a source alphabet in source coding. This implies that finding variable-length code words using bit allocation is equivalent to finding variable spectral dimensions of q and variable spectral bands of $\tilde{q}$ for $\{\mathbf{s}_j\}_{j=1}^{n_S}$ using DDA. The following three well-established coding schemes are readily applied to determine DDA.

22.4.1 Shannon Coding-Based DDA

An earliest and well-known coding scheme was developed by Shannon who introduced self-information to account for information provided by each source alphabet. For each source alphabet

a_j with probability p_j, the self-information of a_j, $I(a_j)$ is defined in Fano (1961) as

$$I(a_j) = -\log p_j \tag{22.2}$$

Using (22.2) a Shannon coding-based scheme to determine DDA can be described as follows.

Shannon coding for dynamic dimensionality allocation

1. Find the self-information of $\mathbf{s}_j$, $I(\mathbf{s}_j) = -\log \pi_j$ for all $1 \le j \le n_\mathrm{S}$.
2. Find q_j for the signature $\mathbf{s}_j$ by

$$q_j = \begin{cases} \lceil -\log \pi_j \rceil; & \text{if } \lceil -\log \pi_j \rceil \le L \\ L; & \text{otherwise} \end{cases} \tag{22.3}$$

   where $\lceil x \rceil$ is defined by the smallest integer $\ge x$.
3. Define the *j*th dimensionality allocation $b_j = n_\mathrm{S} + q_j$ assigned to the signature $\mathbf{s}_j$ where $n_\mathrm{S} \le b_j \le n_\mathrm{S} + L$.

It should be noted that in step 3 the dimensionality allocation $b_{\mathbf{s}_j}$ is broken down into two values. The first value n_S is the number of spectral dimensions/bands required to specify the n_S signatures $\{\mathbf{s}_j\}_{j=1,}^{n_\mathrm{S}}$ where one spectral dimension/band is required to accommodate each signature. When one spectral dimension/band is being used for signature accommodation, it is encoded by "1" and "0" otherwise. According to the definition of VD in Chapter 5, n_S is the number of spectrally distinct signatures if there is no prior signature knowledge available. So, there are at least n_VD bits required to accommodate all the $n_\mathrm{S} = n_\mathrm{VD}$ signatures, one bit for an individual signature. Such bits can be considered as information bits. The second value q_j is the number of spectral dimensions/bands required for $\mathbf{s}_j$ to distinguish itself from other signatures, $\{\mathbf{s}_i\}_{i=1,i\neq j}^{\mathrm{ns}}$. In this case, the self-information $I(\mathbf{s}_j)$ defined by (22.2) is used to calculate additional q_j bits that represent q_j spectral dimensions/bands required to discriminate the signature $\mathbf{s}_j$ from other signatures, $\{\mathbf{s}_i\}_{i=1,i\neq j}^{\mathrm{ns}}$. Since the $I(\mathbf{s}_j) = -\log \pi_j$ may be very large when π_j is very small in which case, it is set by the total number of spectral bands, L. To address this issue the Shannon coding is replaced by the Huffman coding as described in the following.

22.4.2 Huffman Coding-Based DDA

From a theoretical point of view the Shannon code is an asymptotic optimal code. But practically, it is not an optimal code. It is well-known that the only practical optimal code is the one developed by Huffman (Cover and Thomas, 1991), referred to as Huffman coding. It is a variable-length coding technique which is based on a simple fact that the smaller probability the source alphabet is, the longer its coding length. By replacing Shannon code with the Huffman code, the Shannon coding for DDA becomes the following Huffman coding for DDA.

Huffman coding for dynamic dimensionality allocation

1. Find the Huffman code words for $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_{n_\mathrm{S}})$ and let l_j be the coding length of $\mathbf{s}_j$ using the probability π_j.
2. Define dimensionality allocation $b_j = n_\mathrm{S} + l_j$ assigned to the signature $\mathbf{s}_j$ where $n_\mathrm{S} \le b_j \le n_\mathrm{S} + \max_{1 \le j \le n_\mathrm{S}} l_j$.

22.4.3 Hamming Coding-Based DDA

Both the Shannon coding and Huffman coding described above are variable-length coding. This section presents an interesting fixed-length coding for finding DDA. It is derived from the Hamming code which uses 7 bits comprising of four information bits and three parity check bits to correct a single encoding error. As before, each of the n_S signatures requires one spectral dimension/band for its accommodation. In this case, the spectral dimensions/bands to specify these n_S signatures are referred to as information spectral dimensions/bands. Now the variable lengths, $\{l_j\}_{j=1}^{\mathrm{ns}}$ used by the Huffman coding can be replaced with a fixed length coding using n_S as information spectral dimensions/bands and $\lceil -\log n_\mathrm{S} \rceil$ as parity check spectral dimensions/bands for $\mathbf{s}_j$ for all $1 \le j \le n_\mathrm{S}$. Then steps 1 and 2 in the above Huffman coding can be merged into one step as follows. This resulting coding is called Hamming coding in which case DDA becomes static dimensionality allocation (SDA).

Hamming coding for static dimensionality allocation

Define dimensionality allocation $b_j = n_\mathrm{S} + \lceil -\log n_\mathrm{S} \rceil$ assigned to the signature $\mathbf{s}_j$ for all $1 \le j \le n_\mathrm{S}$.

It should be noted that the Hamming coding for SDA assigns the same dimensionality allocation to all signatures $\{\mathbf{s}_j\}_{j=1}^{\mathrm{ns}}$ with $b_j = n_\mathrm{S} + \lceil -\log n_\mathrm{S} \rceil$ which is bounded above from $2n_\mathrm{VD}$. Since the Hamming coding involves no signature discriminatory probabilities of $\{\mathbf{s}_j\}_{j=1}^{n_\mathrm{S}}$, it does not require prior knowledge of signatures of interest, $\{\mathbf{s}_j\}_{j=1}^{n_\mathrm{S}}$ as the Shannon coding and Huffman coding do. Therefore, its determined DDA is a constant regardless of what applications are. In this case, DDA is reduced to SDA.

22.4.4 Notes on DDA

In the above three coding schemes for finding DDA, the generic notation of n_S is used to indicate the number of signatures of interest that indeed varies with different applications. For example, if the considered application is endmember extraction or linear spectral mixture analysis (LSMA), the n_S can be set to n_VD. On the other hand, if the unsupervised target detection is considered as an application, the desired signature of interest are those virtual signatures generated by the unsupervised virtual signature finding algorithm (UVSFA) derived in Chapter 17. In this case, the n_S is the number of virtual signatures and is shown to be in the range of $[n_\mathrm{VD}, 2n_\mathrm{VD}]$, that is, $n_\mathrm{VD} \le n_\mathrm{S} \le 2n_\mathrm{VD}$. Finally, a very important note on DDA is worthwhile. The effectiveness of DDA is more visible and evidential in applications of BS than that in DR. One is that BS is derived from the pigeon-hole principle where one bit represents one spectral band being used for signature accommodation. However, this may not quite fit DR because each spectral dimension produced by DR is a transformed dimension not the original band dimension. Another is mainly due to the fact the projection index components (PICs) produced by a DR transform are generally orthogonal. However, this is not true for BS that uses spectral bands that are generally highly correlated and not necessarily orthogonal. So, a spectral band is highly prioritized, so are its neighboring spectral bands. That explains why DR can gain little benefit from DDA as shown in Safavi (2010).

22.5 Experiments for Dynamic Dimensionality Allocation

As demonstrated in PSDP and PBDP different signatures required various spectral dimensions/bands to effectively perform linear spectral unmixing due to their different spectral characteristics. These experiments provided evidence that a signature should have its own set of spectral dimensions/bands to characterize its spectral profile. In order for PSDP and PBDP to demonstrate progressive performance, three applications, endmember extraction, land cover/use classification, and spectral unmixing, were considered in Chapters 20 and 21 as the parameters, q and $\tilde{q}$, varied ranging from n_{VD} to $2n_{\mathrm{VD}}$ without knowing exactly what the values of q and $\tilde{q}$ were. DDA presented in Section 22.2 is developed to make an attempt of narrowing down the range of $[n_{\mathrm{VD}},2n_{\mathrm{VD}}]$ to best possible values that can be specified by DDA for dimensionality allocations for various signatures. In the following sections we calculate DDA for the three different image data sets which have been used for experiments conducted in Chapters 20 and 21 where the reference signatures were chosen to be the signature means of signatures of interest. Of course selection of an appropriate reference signature is also an interesting issue which will be discussed in Chapters 25, 27 and 28.

22.5.1 Reflectance Cuprite Data

The reflectance Cuprite data in Figure 1.12 has been used for experiments in PART II to evaluate the effectiveness of an endmember extraction algorithm. According to the five known mineral signatures, $\mathbf{m}_1$ = alunite, $\mathbf{m}_2$ = buddingtonite, $\mathbf{m}_3$ = calcite, $\mathbf{m}_4$ = kaolinite, and $\mathbf{m}_5$ = muscovite provided by the ground truth, DDA for these five signatures can be calculated by three coding methods, Shannon coding, Huffman coding, and Hamming coding and their results are tabulated in the last column of Table 22.1 where the signature discriminatory probabilities along with their corresponding lengths are calculated by two spectral measures, SAM and SID and provided in the

Table 22.1 DDA results calculated by the three coding methods for reflectance cuprite data

	Signature $\mathbf{m}_j$	n_{VD}		π_{j}	q_{j} or l_{j}	b_{j} (DDA)
Shannon coding	$\mathbf{m}_1$ (Alunite)	22	SID	0.1727	$\lceil 2.5334 \rceil = 3$	**25**
			SAM	0.1790	$\lceil 2.4823 \rceil = 3$	**25**
	$\mathbf{m}_2$ (Buddingtonite)	22	SID	0.0764	$\lceil 3.7112 \rceil = 4$	**26**
			SAM	0.1150	$\lceil 3.1205 \rceil = 4$	**26**
	$\mathbf{m}_3$ (Calcite)	22	SID	0.1455	$\lceil 2.7806 \rceil = 3$	**25**
			SAM	0.1728	$\lceil 2.5327 \rceil = 3$	**25**
	$\mathbf{m}_4$ (Kaolinite)	22	SID	0.2529	$\lceil 1.9835 \rceil = 2$	**24**
			SAM	0.1922	$\lceil 2.3795 \rceil = 3$	**25**
	$\mathbf{m}_5$ (Muscovite)	22	SID	0.0372	$\lceil 4.4784 \rceil = 5$	**27**
			SAM	0.0888	$\lceil 3.4929 \rceil = 4$	**26**
	$\mathbf{m}_6$ (BKG)	22	SID	0.3153	$\lceil 1.6652 \rceil = 2$	**24**
			SAM	0.2522	$\lceil 1.9872 \rceil = 2$	**24**
Huffman coding	$\mathbf{m}_1$ (Alunite)	22	SID	0.1727	2	**24**
			SAM	0.1790	3	**25**
	$\mathbf{m}_2$ (Buddingtonite)	22	SID	0.0764	4	**26**
			SAM	0.1150	3	**25**
	$\mathbf{m}_3$ (Calcite)	22	SID	0.1455	3	**25**
			SAM	0.1728	3	**25**
	$\mathbf{m}_4$ (Kaolinite)	22	SID	0.2529	2	**24**
			SAM	0.1922	2	**24**
	$\mathbf{m}_5$ (Muscovite)	22	SID	0.0372	4	**26**
			SAM	0.0888	3	**25**
	$\mathbf{m}_6$ (BKG)	22	SID	0.3153	2	**24**
			SAM	0.2522	2	**24**
Hamming coding	$\mathbf{m}_1$ to $\mathbf{m}_6$	22	n/a	n/a	3	**25**

table for reference. In addition, the value of VD, $n_{\mathrm{VD}} = 22$ is also included in the table for comparison. As we can see from the table the values of DDA is in the range between 24 and 26.

To substantiate the fact that the values provided by DDA are good estimates of q and $\tilde{q}$ IN-FINDR was used to extract endmembers. Table 22.2 tabulates extracted endmembers where SID was used to measure spectral similarity with a "1" indicating a success in extraction of one mineral signature and a "0" indicating a failure in missing extraction of one mineral signature.

Table 22.2 Endmember extraction using DDA for reflectance cuprite data

		n_{VD}	Shannon coding	Huffman coding	Hamming coding	$2n_{\mathrm{VD}}$	Total
$\tilde{q}$ = No. of selected bands		22	25	24	25	44	189
A	Variance	1	1	1	1	1	1
	Skewness	0	0	0	0	1	1
	Kurtosis	0	1	1	1	1	1
	Entropy	1	1	1	1	1	1
	ID	1	0	0	0	1	1
	Negentropy	0	1	1	1	0	1
$\tilde{q}$ = No. of selected bands		22	26	26	25	44	189
B	Variance	0	1	1	1	1	1
	Skewness	1	1	1	0	0	1
	Kurtosis	1	1	1	0	1	1
	Entropy	1	0	0	1	1	1
	ID	1	1	1	0	1	1
	Negentropy	1	1	1	0	0	1
$\tilde{q}$ = No. of selected bands		22	25	25	25	44	189
C	Variance	1	0	0	0	1	1
	Skewness	1	1	1	1	1	1
	Kurtosis	1	1	1	1	1	1
	Entropy	1	1	1	1	1	1
	ID	1	1	1	1	1	1
	Negentropy	1	1	1	1	1	1
$\tilde{q}$ = No. of selected bands		22	24	24	25	44	189
K	Variance	1	1	1	1	1	1
	Skewness	1	1	1	0	1	1
	Kurtosis	1	1	1	1	1	1
	Entropy	1	1	1	1	1	1
	ID	1	0	0	1	0	1
	Negentropy	0	1	1	1	1	1
$\tilde{q}$ = No. of selected bands		22	27	26	25	44	189
M	Variance	1	1	0	0	0	1
	Skewness	0	1	1	1	1	1
	Kurtosis	0	1	0	0	1	1
	Entropy	0	1	1	1	1	1
	ID	1	1	1	1	0	1
	Negentropy	0	1	1	0	1	1

An interesting finding in Table 22.2 is that using more bands did not guarantee better endmember extraction. For example, when bands were prioritized by skewness, IN-FINDR successfully extracted the mineral signature "B" if the value of $\hat{q}$ was 22, but it failed at $\hat{q} = 25$. Then it succeeded again when the value of $\hat{q}$ was increased to 26. A similar phenomenon was also observed when bands were prioritized by variance to extract the mineral signature M where IN-FINDR was able to extract "M" when $\hat{q} = 22$, but failed when $\hat{q} = 25, 26$. It then succeeded again when $\hat{q} = 27$, but also failed again when $\hat{q} = 44$. Also interesting is that $n_{\text{VD}} = 22$ turned out to be a very good estimate. On many occasions IN-FINDR successfully extracted mineral signatures at $\hat{q} = n_{\text{VD}} = 22$, but failed at $\hat{q} > 22$, such as ID at $\hat{q} = 24, 25$ failing to extract mineral signature "A," entropy at $\hat{q} = 26$ failing to extract mineral signature "B", variance at $\hat{q} = 25$ failing to extract mineral signature "C," skewness at $\hat{q} = 25$ failing to extract mineral signature "K," variance at $\hat{q} = 25, 26$ failing to extract mineral signature "M." Nevertheless, according to Table 22.2 the best result was the one using negentropy as a BP criterion to prioritize bands and DDA determined by Shannon coding or Huffman coding in which case all the five mineral signatures could be successfully extracted by IN-FINDR.

22.5.2 Purdue's Data

The Purdue Indiana Indian Pine test site in Figure 1.13 has been studied extensively in the literature for land cover/use classification. There are 16 classes of interest plus a background class. So, a total of 17 classes are present in the data. Each of 17 class signatures was found by the sample mean of 50% sample vectors randomly selected from its class and the other 50% will be used as test sample vectors for maximum likelihood classification (MLC). Table 22.3 tabulates DDA results using Shannon coding, Huffman coding, and Hamming coding where the signature discriminatory probabilities along with their corresponding lengths were calculated by two spectral

Table 22.3 DDA results calculated by the three coding methods for the Purdue data

	Signature $\mathbf{m}_j$	n_{VD}		π_{j}	q_{j} or l_{j}	b_{j} (DDA)
Shannon coding	$\mathbf{m}_1$ (class 1)	29	SID	0.0128	$\lceil 6.2933 \rceil = 7$	**36**
			SAM	0.0304	$\lceil 5.0394 \rceil = 6$	**35**
	$\mathbf{m}_2$ (class 2)	29	SID	0.0437	$\lceil 4.5159 \rceil = 5$	**34**
			SAM	0.0587	$\lceil 4.0908 \rceil = 5$	**34**
	$\mathbf{m}_3$ (class 3)	29	SID	0.0392	$\lceil 4.6748 \rceil = 5$	**34**
			SAM	0.0551	$\lceil 4.1824 \rceil = 5$	**34**
	$\mathbf{m}_4$ (class 4)	29	SID	0.0140	$\lceil 6.1622 \rceil = 7$	**36**
			SAM	0.0322	$\lceil 4.9577 \rceil = 5$	**34**
	$\mathbf{m}_5$ (class 5)	29	SID	0.1341	$\lceil 2.8986 \rceil = 3$	**32**
			SAM	0.1002	$\lceil 3.3194 \rceil = 4$	**33**
	$\mathbf{m}_6$ (class 6)	29	SID	0.0316	$\lceil 4.9826 \rceil = 5$	**34**
			SAM	0.0487	$\lceil 4.3591 \rceil = 5$	**34**
	$\mathbf{m}_7$ (class 7)	29	SID	0.0042	$\lceil 7.9007 \rceil = 8$	**37**
			SAM	0.0174	$\lceil 5.8465 \rceil = 6$	**35**
	$\mathbf{m}_8$ (class 8)	29	SID	0.0104	$\lceil 6.5866 \rceil = 7$	**36**
			SAM	0.0273	$\lceil 5.1946 \rceil = 6$	**35**
	$\mathbf{m}_9$ (class 9)	29	SID	0.0154	$\lceil 6.0226 \rceil = 7$	**36**
			SAM	0.0344	$\lceil 4.8631 \rceil = 5$	**34**
	$\mathbf{m}_{10}$ (class 10)	29	SID	0.0543	$\lceil 4.2035 \rceil = 5$	**34**
			SAM	0.0653	$\lceil 3.9372 \rceil = 4$	**33**

	Signature $\mathbf{m}_j$	n_{VD}		π_{j}	q_{j} or l_{j}	b_{j} (DDA)
	$\mathbf{m}_{11}$ (class 11)	29	SID	0.0480	$\lceil 4.3798 \rceil = 5$	**34**
			SAM	0.0609	$\lceil 4.0380 \rceil = 5$	**34**
	$\mathbf{m}_{12}$ (class 12)	29	SID	0.0430	$\lceil 4.5388 \rceil = 5$	**34**
			SAM	0.0584	$\lceil 4.0984 \rceil = 5$	**34**
	$\mathbf{m}_{13}$ (class 13)	29	SID	0.0277	$\lceil 5.1716 \rceil = 6$	**35**
			SAM	0.0428	$\lceil 4.5446 \rceil = 5$	**34**
	$\mathbf{m}_{14}$ (class 14)	29	SID	0.2293	$\lceil 2.1248 \rceil = 3$	**32**
			SAM	0.1281	$\lceil 2.9650 \rceil = 3$	**32**
	$\mathbf{m}_{15}$ (class 15)	29	SID	0.0580	$\lceil 4.1082 \rceil = 5$	**34**
			SAM	0.0650	$\lceil 3.9429 \rceil = 4$	**33**
	$\mathbf{m}_{16}$ (class 16)	29	SID	0.2105	$\lceil 2.2483 \rceil = 3$	**32**
			SAM	0.1326	$\lceil 2.9149 \rceil = 3$	**32**
	$\mathbf{m}_{17}$ (BKG)	29	SID	0.0239	$\lceil 5.3868 \rceil = 6$	**35**
			SAM	0.0426	$\lceil 4.5521 \rceil = 5$	**34**
Huffman coding	$\mathbf{m}_1$ (class 1)	29	SID	0.0128	7	**36**
			SAM	0.0304	5	**34**
	$\mathbf{m}_2$ (class 2)	29	SID	0.0437	5	**34**
			SAM	0.0587	4	**33**
	$\mathbf{m}_3$ (class 3)	29	SID	0.0392	5	**34**
			SAM	0.0551	4	**33**
	$\mathbf{m}_4$ (class 4)	29	SID	0.0140	7	**36**
			SAM	0.0322	5	**34**
	$\mathbf{m}_5$ (class 5)	29	SID	0.1341	3	**32**
			SAM	0.1002	3	**32**
	$\mathbf{m}_6$ (class 6)	29	SID	0.0316	5	**34**
			SAM	0.0487	4	**33**
	$\mathbf{m}_7$ (class 7)	29	SID	0.0042	7	**36**
			SAM	0.0174	6	**35**
	$\mathbf{m}_8$ (class 8)	29	SID	0.0104	7	**36**
			SAM	0.0273	6	**35**
	$\mathbf{m}_9$ (class 9)	29	SID	0.0154	6	**35**
			SAM	0.0344	5	**34**
	$\mathbf{m}_{10}$ (class 10)	29	SID	0.0543	4	**33**
			SAM	0.0653	4	**33**
	$\mathbf{m}_{11}$ (class 11)	29	SID	0.0480	5	**34**
			SAM	0.0609	4	**33**
	$\mathbf{m}_{12}$ (class 12)	29	SID	0.0430	5	**34**
			SAM	0.0584	4	**33**
	$\mathbf{m}_{13}$ (class 13)	29	SID	0.0277	5	**34**
			SAM	0.0428	5	**34**
	$\mathbf{m}_{14}$ (class 14)	29	SID	0.2293	2	**31**
			SAM	0.1281	3	**32**
	$\mathbf{m}_{15}$ (class 15)	29	SID	0.0580	4	**33**
			SAM	0.0650	4	**33**
	$\mathbf{m}_{16}$ (class 16)	29	SID	0.2105	2	**31**
			SAM	0.1326	3	**32**
	$\mathbf{m}_{17}$ (BKG)	29	SID	0.0239	6	**35**
			SAM	0.0426	5	**34**
Hamming coding	$\mathbf{m}_1$ to $\mathbf{m}_{17}$	29	n/a	n/a	5	**34**

measures, SAM and SID, and provided in the table for reference. In addition, the value of VD, $n_{\mathrm{VD}} = 29$, was also included in the table for comparison.

Table 22.4 tabulates the classification rates by MLC for 16 classes using DDA calculated in Table 22.3 where SID was used to measure spectral similarity. Also included in the table is the last column which gave the best classification rates with the optimal number of bands provided in parentheses.

It is noted that in most cases MLC only needed a fewer prioritized bands to perform well without using all spectral bands, such as classes 7, 9, and 16 that did not require many spectral bands to produce the best results. There are some observations noteworthy. One is that second order statistic BPC generally performed better than high order and infinite order BPC due to the fact that the land covers of this particular scene are large, and the data samples are heavily mixed because of low spatial resolution and their contributions to statistics are mainly second-order statistics. This is not true for high-resolution hyperspectral data, such as HYDICE as will be demonstrated in the following section, which searches for rare data samples with pure signatures. On the other hand, it can be noted that in most cases using fewer band dimensions MLC could perform as well its using all dimensions. For instance, classes 7, 9, and 16 did not require more bands to produce the best results. There are only classes, 2, 3, 4, 6, 10, and 11 that required almost full dimensions to produce the best MLC results. Nevertheless, in general, DDA did provide some guidelines in selecting an appropriate value of $\tilde{q}$ for MLC to perform reasonably well. Finally, similar to endmember extraction Table 22.4 also demonstrates that using more bands did not necessarily produce better classification for each of seven used BP criteria (i.e., variance, SNR, skewness, kurtosis, entropy, ID, and negentropy), for example, classification of class 1 with SNR used as a BP criterion at $\hat{q} = 29$ versus 34 and 36 as well as with kurtosis used as a BP criterion at $\hat{q} = 36$ versus 58; classification of class 6 with skewness used as BP criterion at $\hat{q} = 29$ versus 34; classification of class 7 with variance used as BP criterion at $\hat{q} = 36$ and 37 versus 58 as well as with negentropy used as a BP criterion at $\hat{q} = 36$ versus 58; classification of class 16 with ID used as BP criterion at $\hat{q} = 31$ versus 58 as well as with entropy used as a BP criterion at $\hat{q} = 29$ versus 31. Nevertheless, according to Table 22.4, if DDA was used to determined $\hat{q}$ the overall performance on MLC rates of 16 classes was generally improved over that produced by letting $\hat{q} = n_{\mathrm{VD}}$ regardless of which BP criterion was used.

22.5.3 HYDICE Data

The five panel signatures in Figure 1.16 was used to unmix the 19 R panel pixels, $p_{11}, p_{12}, p_{13}, p_{211}, p_{221}, p_{22}, p_{23}, p_{311}, p_{312}, p_{32}, p_{33}, p_{411}, p_{412}, p_{42}, p_{43}, p_{511}, p_{521}, p_{52}, p_{53}$ in the HYDICE 15-panel scene in Figure 1.15 for abundance fraction quantification. According to the ground truth in Figures 1.15–1.17, nine signatures, $\mathbf{m}_1 = \mathrm{p}_1$, $\mathbf{m}_2 = \mathbf{p}_2$, $\mathbf{m}_3$=$\mathbf{p}_3$, $\mathbf{m}_4 = \mathbf{p}_4$, and $\mathbf{m}_5 = \mathbf{p}_5$ in Figure 1.16 and figure background signatures, $\mathbf{m}_6$ = grass $\mathbf{m}_7$ = road $\mathbf{m}_8$ = tree and $\mathbf{m}_9$ = interferer in Figure 1.17 were used as a desired set of signatures of interest. Table 22.5 tabulates DDA calculated by Shannon coding, Huffman coding, and Hamming coding using these nine signatures where SAM and SID were used to measure spectral similarity.

Table 22.6 tabulates FCLS-unmixed abundance fractions of the 19 R panel pixels with the number of selected prioritized spectral bands determined by n_{VD} and DDA using the three coding methods, and optimal number of bands to produce the best results in the last column.

Several BP criteria were chosen to investigate DA. The results demonstrated that the skewness produced the best performance of FCLS abundance results because in most cases skewness criterion required less bands to achieve the optimal performance. On the other hand, for few panel pixels the best estimated p was around 9 that was the same as n_{VD}. For most of panel pixels $p = 9$

Table 22.4 Classification rates of 16 classes in the Purdue data by MLC

		n_{VD}	Shannon coding	Huffman coding	Hamming coding	$2n_{\text{VD}}$	Total	Optimal
Class 1	$\tilde{q}$	29	36	36	34	58	202	
	Variance	62.96	70.37	70.37	70.37	70.37	85.19	85.19 (129)
	SNR	75.93	70.37	70.37	70.37	75.93		85.19 (179)
	Skewness	53.7	61.11	61.11	61.11	68.52		87.04 (200)
	Kurtosis	72.22	83.33	83.33	77.78	77.78		88.89 (169)
	Entropy	79.63	83.33	83.33	85.19	87.04		87.04 (50)
	ID	55.56	68.52	68.52	62.96	66.67		88.89 (163)
	Negentropy	68.52	70.37	70.37	77.78	62.96		88.89 (193)
Class 2	$\tilde{q}$	29	34	34	34	58	202	
	Variance	30.26	35.91	35.91	35.91	44.77	67.5	68.48 (187)
	SNR	42.68	46.93	46.93	46.93	55.93		68.48 (174)
	Skewness	33.4	33.61	33.61	33.61	45.96		67.71 (177)
	Kurtosis	41.14	43.38	43.38	43.38	44.49		67.5 (202)
	Entropy	54.95	57.46	57.46	57.46	60.04		69.94 (190)
	ID	39.75	45.75	45.75	45.75	48.19		67.64 (189)
	Negentropy	30.47	31.66	31.66	31.66	43.58		68.62 (194)
Class 3	$\tilde{q}$	29	34	34	34	58	202	
	Variance	24.46	26.86	26.86	26.86	33.93	54.8	54.92 (189)
	SNR	42.68	46.93	46.93	46.93	55.93		68.48 (174)
	Skewness	23.62	23.62	23.62	23.62	35.85		54.8 (202)
	Kurtosis	27.22	30.1	30.1	30.1	34.65		54.8 (202)
	Entropy	30.58	32.85	32.85	32.85	39.69		55.04 (195)
	ID	33.09	34.17	34.17	34.17	41.73		54.8 (198)
	Negentropy	32.13	33.69	33.69	33.69	39.09		54.8 (201)
Class 4	$\tilde{q}$	29	36	36	34	58	202	
	Variance	42.74	49.57	49.57	46.58	64.96	78.63	82.05 (159)
	SNR	64.53	69.66	69.66	67.95	71.37		82.05 (63)
	Skewness	46.15	50.43	50.43	49.57	61.54		82.91 (146)
	Kurtosis	52.56	53.42	53.42	55.13	59.83		79.91 (201)
	Entropy	73.08	73.93	73.93	75.21	74.79		82.91 (101)
	ID	45.3	67.95	67.95	58.55	70.51		81.62 (185)
	Negentropy	61.11	60.26	60.26	62.39	64.1		81.62 (167)
Class 5	$\tilde{q}$	29	32	32	34	58	202	
	Variance	60.56	62.17	62.17	61.17	65.39	65.79	66 (70)
	SNR	57.34	61.77	61.77	59.96	65.39		66 (70)
	Skewness	57.95	59.56	59.56	58.55	62.37		65.79 (121)
	Kurtosis	54.53	64.19	64.19	63.98	65.39		66 (180)
	Entropy	59.56	61.97	61.97	58.95	64.19		65.79 (89)
	ID	35.81	54.53	54.53	43.86	64.99		65.79 (77)
	Negentropy	45.27	48.49	48.49	45.67	54.12		65.79 (183)
Class 6	$\tilde{q}$	29	34	34	34	58	202	
	Variance	86.21	85.81	85.81	85.81	86.35	91.97	92.24 (199)
	SNR	64.26	71.49	71.49	71.49	85.68		92.37 (197)
	Skewness	79.12	78.05	78.05	78.05	85.14		92.24 (193)
	Kurtosis	78.45	79.65	79.65	79.65	85.68		92.5 (179)
	Entropy	79.79	80.05	80.05	80.05	84.74		92.9 (188)

(*continued*)

Table 22.4 (*Continued*)

		n_{VD}	Shannon coding	Huffman coding	Hamming coding	$2n_{\mathrm{VD}}$	Total	Optimal
	ID	50.07	64.26	64.26	64.26	88.22		91.97 (196)
	Negentropy	50.2	51.54	51.54	51.54	65.19		92.1 (200)
Class 7	$\tilde{q}$	29	37	36	34	58	202	
	Variance	73.08	80.77	80.77	73.08	76.92	73.08	84.62 (38)
	SNR	61.54	69.23	69.23	57.69	73.08		88.46 (101)
	Skewness	73.08	76.92	76.92	73.08	76.92		80.77 (54)
	Kurtosis	65.38	69.23	69.23	76.92	69.23		80.77 (143)
	Entropy	69.23	76.92	73.08	73.08	80.77		88.46 (94)
	ID	65.38	65.38	65.38	65.38	69.23		84.62 (69)
	Negentropy	69.23	84.62	84.62	73.08	61.54		84.62 (36)
Class 8	$\tilde{q}$	29	36	36	34	58	202	
	Variance	67.69	76.89	76.89	75.87	88.14	97.96	98.36 (186)
	SNR	67.08	75.87	75.87	71.17	82.41		98.36 (180)
	Skewness	72.19	74.64	74.64	73.42	87.73		98.57 (190)
	Kurtosis	81.39	87.12	87.12	86.09	88.96		98.16 (162)
	Entropy	76.89	86.3	86.3	85.89	90.8		98.16 (190)
	ID	66.05	82.62	82.62	71.37	85.28		97.96 (177)
	Negentropy	74.23	77.51	77.51	76.07	81.8		98.57 (188)
Class 9	$\tilde{q}$	29	36	35	34	58	202	
	Variance	65	70	65	65	80	65	90 (84)
	SNR	65	75	75	70	60		75 (35)
	Skewness	75	80	75	75	85		90 (57)
	Kurtosis	60	60	60	60	55		75 (137)
	Entropy	65	65	65	65	65		75 (55)
	ID	50	50	50	50	55		70 (195)
	Negentropy	55	50	50	55	55		70 (191)
Class 10	$\tilde{q}$	29	34	33	34	58	202	
	Variance	42.46	41.63	41.63	41.63	52.58	86.16	87.71 (181)
	SNR	54.55	56.92	56.61	56.92	67.25		87.09 (176)
	Skewness	39.46	39.26	39.36	39.26	58.68		86.78 (180)
	Kurtosis	55.37	59.61	59.09	59.61	64.98		86.16 (202)
	Entropy	56.82	59.81	59.4	59.81	69.11		87.09 (196)
	ID	69.52	71.28	70.04	71.28	73.86		86.98 (195)
	Negentropy	67.56	67.25	66.84	67.25	75.62		87.09 (172)
Class 11	$\tilde{q}$	29	34	34	34	58	202	
	Variance	42.34	40.84	40.84	40.84	41.29	51.09	51.58 (199)
	SNR	35.86	39.22	39.22	39.22	40.52		51.58 (199)
	Skewness	39.71	40.15	40.15	40.15	40.88		51.5 (185)
	Kurtosis	36.87	38.7	38.7	38.7	42.67		52.23 (163)
	Entropy	40.32	42.3	42.3	42.3	45.87		52.8 (178)
	ID	32.9	35.53	35.53	35.53	41.57		51.09 (202)
	Negentropy	33.31	32.25	32.25	32.25	38.41		51.26 (197)
Class 12	$\tilde{q}$	29	34	34	34	58	202	
	Variance	35.18	39.41	39.41	39.41	44.63	77.85	79.64 (151)
	SNR	55.21	58.47	58.47	58.47	65.64		78.83 (166)
	Skewness	30.13	29.8	29.8	29.8	51.79		78.83 (165)
	Kurtosis	40.72	48.05	48.05	48.05	55.37		79.15 (181)

		n_{VD}	Shannon coding	Huffman coding	Hamming coding	$2n_{\mathrm{VD}}$	Total	Optimal
	Entropy	60.26	62.38	62.38	62.38	63.84		79.15 (191)
	ID	43.49	47.72	47.72	47.72	58.31		78.5 (199)
	Negentropy	42.67	46.25	46.25	46.25	60.59		80.29 (170)
Class 13	$\tilde{q}$	29	35	34	34	58	202	
	Variance	97.64	98.58	99.53	99.53	99.53	99.53	99.53 (34)
	SNR	96.7	98.58	96.7	96.7	99.53		99.53 (48)
	Skewness	91.04	91.51	91.51	91.51	97.64		99.53 (100)
	Kurtosis	95.28	96.7	97.17	97.17	98.58		99.53 (61)
	Entropy	100	99.06	99.06	99.06	98.58		100 (29)
	ID	65.57	95.75	95.75	95.75	99.53		100 (44)
	Negentropy	87.74	92.45	91.98	91.98	92.45		100 (131)
Class 14	$\tilde{q}$	29	32	31	34	58	202	
	Variance	75.66	76.82	75.73	75.73	79.83	86.71	87.4 (168)
	SNR	71.02	74.73	74.65	74.65	90.49		90.8 (49)
	Skewness	76.97	75.81	75.58	75.58	78.21		87.71 (167)
	Kurtosis	68.01	75.04	75.19	75.19	73.88		86.71 (201)
	Entropy	79.06	79.52	79.68	79.68	82.46		87.02 (175)
	ID	58.73	60.43	60.97	60.97	80.6		86.79 (201)
	Negentropy	50.7	58.5	55.33	55.33	72.1		91.65 (156)
Class 15	$\tilde{q}$	29	34	33	34	58	202	
	Variance	47.63	53.16	52.11	53.16	69.21	88.95	90.79 (124)
	SNR	31.58	40	35	40	63.42		89.47 (178)
	Skewness	44.74	48.68	48.42	48.68	77.89		89.47 (194)
	Kurtosis	39.74	49.74	49.47	49.74	73.68		92.63 (140)
	Entropy	40.26	48.95	48.68	48.95	65.53		90.26 (125)
	ID	29.74	51.32	32.89	51.32	62.63		91.84 (173)
	Negentropy	36.84	37.63	39.21	37.63	46.84		89.74 (194)
Class 16	$\tilde{q}$	29	32	31	34	58	202	
	Variance	92.63	92.63	92.63	92.63	87.37	89.47	93.68 (31)
	SNR	93.68	93.68	93.68	93.68	88.42		93.68 (20)
	Skewness	86.32	86.32	86.32	86.32	88.42		90.53 (91)
	Kurtosis	86.32	86.32	86.32	86.32	86.32		90.53 (186)
	Entropy	88.42	87.37	87.37	87.37	88.42		91.58 (17)
	ID	89.47	89.47	90.53	89.47	87.37		90.53 (33)
	Negentropy	88.42	88.42	88.42	88.42	89.47		90.53 (192)

seemed to be insufficient but they still achieved good performance within $p = 18 = 2\ n_{\mathrm{VD}}$. Similar to AVIRIS data experiments conducted for endmember extraction in Section 22.5.1 and land cover/use classification in Section 22.5.2 the same conclusion that using more bands did not necessarily improve performance was also applied to HYDICE data. For example, when the three BP criteria, SNR, skewness and entropy were used to prioritize bands, the FCLS-unmixed abundance fraction of p_{511} at $\hat{q} = 9$ was more accurate than that obtained at $\hat{q} = 12$. Similarly, the same conclusions were also applied to the unmixed abundance fraction of the panel pixel p_{211} with variance used as a BP criterion at $\hat{q} = 9$ against 13–15, 18; unmixed abundance fraction of the panel pixel p_{221} with kurtosis used as a BP criterion at $\hat{q} = 9$ against 14–15, 18, FCLS-unmixed abundance fraction of the panel pixel p_{412} with negentropy used as a BP criterion at $\hat{q} = 9$ versus 13–15,18; unmixed abundance fraction of the panel pixel p_{12} with ID used as a BP criterion at $\hat{q} = 9$ against 13–15,

Table 22.5 DDA results calculated by the three coding methods for HYDICE data

	Signature $\mathbf{m}_j$	n_{VD}		π_{j}	q_{j} or l_{j}	b_{j} (DDA)
Shannon coding	$\mathbf{m}_1 = \mathbf{p}_1$	9	SID	0.0172	$\lceil 5.8591 \rceil = 6$	**15**
			SAM	0.0462	$\lceil 4.4350 \rceil = 5$	**14**
	$\mathbf{m}_2 = \mathbf{p}_2$	9	SID	0.0295	$\lceil 5.0832 \rceil = 6$	**15**
			SAM	0.0779	$\lceil 3.6822 \rceil = 4$	**13**
	$\mathbf{m}_3 = \mathbf{p}_3$	9	SID	0.0358	$\lceil 4.8055 \rceil = 5$	**14**
			SAM	0.0897	$\lceil 3.4794 \rceil = 4$	**13**
	$\mathbf{m}_4 = \mathbf{p}_4$	9	SID	0.0695	$\lceil 3.8466 \rceil = 4$	**13**
			SAM	0.1004	$\lceil 3.3163 \rceil = 4$	**13**
	$\mathbf{m}_5 = \mathbf{p}_5$	9	SID	0.1070	$\lceil 3.2246 \rceil = 4$	**13**
			SAM	0.1140	$\lceil 3.1331 \rceil = 4$	**13**
	$\mathbf{m}_6$ (grass)	9	SID	0.1007	$\lceil 3.3114 \rceil = 4$	**13**
			SAM	0.1396	$\lceil 2.8403 \rceil = 3$	**12**
	$\mathbf{m}_7$ (road)	9	SID	0.0565	$\lceil 4.1463 \rceil = 5$	**14**
			SAM	0.1035	$\lceil 3.2727 \rceil = 4$	**13**
	$\mathbf{m}_8$ (tree)	9	SID	0.2869	$\lceil 1.8014 \rceil = 2$	**12**
			SAM	0.1479	$\lceil 2.7571 \rceil = 3$	**12**
	$\mathbf{m}_9$ (interferer)	9	SID	0.2969	$\lceil 1.7518 \rceil = 2$	**11**
			SAM	0.1808	$\lceil 2.4674 \rceil = 3$	**12**
Huffman coding	$\mathbf{m}_1 = \mathbf{p}_1$	9	SID	0.0172	5	**14**
			SAM	0.0462	4	**13**
	$\mathbf{m}_2 = \mathbf{p}_2$	9	SID	0.0295	5	**14**
			SAM	0.0779	4	**13**
	$\mathbf{m}_3 = \mathbf{p}_3$	9	SID	0.0358	4	**13**
			SAM	0.0897	3	**12**
	$\mathbf{m}_4 = \mathbf{p}_4$	9	SID	0.0695	4	**13**
			SAM	0.1004	3	**12**
	$\mathbf{m}_5 = \mathbf{p}_5$	9	SID	0.1070	3	**12**
			SAM	0.1140	3	**12**
	$\mathbf{m}_6$ (grass)	9	SID	0.1007	3	**12**
			SAM	0.1396	3	**12**
	$\mathbf{m}_7$ (road)	9	SID	0.0565	4	**13**
			SAM	0.1035	3	**12**
	$\mathbf{m}_8$ (tree)	9	SID	0.2869	2	**11**
			SAM	0.1479	3	**12**
	$\mathbf{m}_9$ (interferer)	9	SID	0.2969	2	**11**
			SAM	0.1808	3	**12**
Hamming coding	$\mathbf{m}_1$ to $\mathbf{m}_9$	9		n/a	4	**13**

18. However, according to Table 22.6, if DDA was used to determine $\hat{q}$, the overall performance on FCLS-unmixing 19 R panel pixels was generally better than that produced by letting $\hat{q} = n_{\mathrm{VD}}$ regardless of which BP criterion was used.

Finally, several conclusions can be drawn from Tables 22.2, 22.4 and 22.6. First, a BP criterion has a significant impact on performance. Its selection must be determined by a specific application. Second, it is generally not true that using more BP-ranked bands will yield better performance. This may be due to the fact that high prioritized bands will also be highly

Table 22.6 Unmixed abundance fractions of 19 panel pixels in the HYDICE data by FCLS

			Shannon	Huffman	Hamming			
		n_{VD}	coding	coding	coding	$2n_{\text{VD}}$	total	Optimal
$\tilde{q} =$ Number of selected bands		9	15	14	13	18	169	
p_{11}	Variance	0	0.16	0.02	0	0.07	0.76	0.76 (169)
	SNR	0.65	0.64	0.64	0.65	0.63		0.77 (146)
	Skewness	0.28	0.72	0.75	0.65	0.75		0.90 (59)
	Kurtosis	0	0	0	0	0.16		0.8 (167)
	Entropy	0.92	0.81	0.81	0.81	0.79		1.16 (11)
	ID	0	0	0	0	0		0.8 (167)
	Negentropy	0.65	0.65	0.66	0.66	0.63		0.8 (146)
p_{12}	Variance	0.44	0.63	0.63	0.62	0.62	0.54	0.69 (27)
	SNR	0.63	0.64	0.65	0.65	0.64		0.74 (10)
	Skewness	0.51	0.57	0.57	0.54	0.58		0.66 (69)
	Kurtosis	0.44	0.63	0.63	0.62	0.62		0.67 (42)
	Entropy	0.51	0.57	0.57	0.54	0.58		1.04 (15)
	ID	0.61	0.53	0.44	0.41	0.58		0.67 (33)
	Negentropy	0.68	1.04	0.99	0.62	0.96		0.72 (9)
p_{13}	Variance	0.72	0.72	0.72	0.72	0.72	0.01	0.24 (21)
	SNR	0	0.04	0.04	0.02	0.04		0.17 (9)
	Skewness	0.72	0.64	0.64	0.7	0.64		0.1 (15)
	Kurtosis	0	0.04	0.04	0.04	0.21		0.2 (108)
	Entropy	0	0.1	0.05	0	0.08		0.1 (58)
	ID	0	0	0	0.03	0		0.18 (81)
	Negentropy	0.02	0	0	0	0.04		0.35 (10)
$\tilde{q} =$ Number of selected bands		9	15	14	13	18	169	
p_{211}	Variance	0.17	0.11	0.1	0.11	0.09	0.91	0.91 (68)
	SNR	0.89	0.94	0.93	0.93	0.86		0.95 (37)
	Skewness	0.79	0.79	0.79	0.81	0.79		1 (59)
	Kurtosis	0.96	0.9	0.9	0.94	0.92		0.96 (9)
	Entropy	0.19	0.3	0.3	0.3	0.68		0.97 (137)
	ID	0.84	0.82	0.82	0.83	0.82		0.91 (167)
	Negentropy	0.95	0.95	0.95	0.95	0.95		0.96 (30)
p_{221}	Variance	0.38	0.24	0.24	0.25	0.24	0.79	0.79 (169)
	SNR	0.81	0.82	0.8	0.81	0.68		0.82 (15)
	Skewness	0.58	0.65	0.65	0.66	0.66		1 (68)
	Kurtosis	0.49	0.38	0.38	0.53	0.41		0.8 (168)
	Entropy	0	0	0	0	0.45		1 (60)
	ID	0.24	0.22	0.21	0.24	0.22		0.8 (167)
	Negentropy	0.89	0.89	0.89	0.89	0.89		0.89 (19)
p_{22}	Variance	0.84	0.8	0.8	0.8	0.81	0.82	0.84 (67)
	SNR	0.72	0.72	0.72	0.71	0.7		0.83 (120)
	Skewness	-0.23	0.87	0.88	0.87	0.86		0.88 (14)
	Kurtosis	0.76	0.81	0.81	0.81	0.8		0.83 (134)
	Entropy	0.7	0.6	0.59	0.62	0.6		0.87 (139)
	ID	0.79	0.79	0.75	0.78	0.79		0.83 (131)
	Negentropy	0.58	0.63	0.63	0.63	0.69		0.83 (112)

(*continued*)

Table 22.6 (*Continued*)

		n_{VD}	Shannon coding	Huffman coding	Hamming coding	$2n_{\mathrm{VD}}$	total	Optimal
p_{23}	Variance	0.45	0.21	0.22	0.21	0.16	0.42	0.45 (9)
	SNR	0.23	0	0.05	0.11	0		0.46 (59)
	Skewness	0.07	0.36	0.34	0.33	0.34		0.44 (146)
	Kurtosis	0.57	0.47	0.46	0.58	0.51		0.6 (10)
	Entropy	0.08	0	0	0	0		0.43 (140)
	ID	0.14	0.08	0.06	0.07	0.1		0.48 (38)
	Negentropy	0.24	0.27	0.27	0.27	0.29		0.46 (36)
$\tilde{q}=$ Number of selected bands		9	14	13	13	18	169	
p_{311}	Variance	0.82	0.93	0.94	0.94	0.98	0.91	0.99 (25)
	SNR	1.04	0.98	0.97	0.97	1		1.13 (10)
	Skewness	0.87	0.83	0.83	0.83	0.88		0.91 (164)
	Kurtosis	0.79	0.81	0.79	0.79	0.82		0.99 (45)
	Entropy	0.88	0.88	0.87	0.87	0.89		0.92 (161)
	ID	0.93	0.9	0.91	0.91	0.9		0.99 (77)
	Negentropy	0.62	0.7	0.7	0.7	0.84		0.99 (66)
p_{312}	Variance	0.02	0.8	0.8	0.8	0.84	0.90	0.90 (94)
	SNR	0.6	0.64	0.64	0.64	0.64		0.92 (62)
	Skewness	0.99	0.96	0.97	0.97	0.97		0.99 (9)
	Kurtosis	0.17	0.1	0.06	0.06	0.1		0.9 (154)
	Entropy	0.96	0.95	0.95	0.95	0.97		0.98 (140)
	ID	0.75	0.73	0.74	0.74	0.74		0.91 (154)
	Negentropy	0	0.39	0.36	0.36	0.62		0.9 (151)
p_{32}	Variance	0.67	0.67	0.67	0.67	0.66	0.45	0.69 (30)
	SNR	0.28	0.02	0.01	0.01	0.24		0.69 (61)
	Skewness	1	0.79	1	1	0.78		1 (9)
	Kurtosis	0.53	0.66	0.56	0.56	0.67		0.67 (18)
	Entropy	0.56	0.49	0.49	0.49	0.59		0.59 (17)
	ID	0.64	0.66	0.65	0.65	0.66		0.67 (27)
	Negentropy	0	0	0	0	0.02		0.61 (47)
p_{33}	Variance	0.46	0.34	0.34	0.34	0.2	0.21	0.46 (9)
	SNR	0.01	0	0	0	0.08		0.3 (39)
	Skewness	0.59	0.43	0.51	0.51	0.44		0.59 (9)
	Kurtosis	0.15	0.39	0.16	0.16	0.28		0.39 (14)
	Entropy	0.27	0.35	0.35	0.35	0.39		0.39 (17)
	ID	0.32	0.36	0.36	0.36	0.37		0.38 (19)
	Negentropy	0	0	0	0	0		0.32 (42)
$\tilde{q}=$ Number of selected bands		9	13	13	13	18	169	
p_{411}	Variance	0.3	0.28	0.28	0.28	0.26	0.51	0.51 (169)
	SNR	0.05	0.06	0.06	0.06	0.09		0.51 (146)
	Skewness	0.54	0.79	0.79	0.79	0.92		0.96 (22)
	Kurtosis	0.35	0.29	0.29	0.29	0.23		0.51 (168)
	Entropy	0.79	0.97	0.97	0.97	0.9		1.02 (14)
	ID	0.34	0.3	0.3	0.3	0.25		0.51 (169)
	Negentropy	0	0	0	0	0.02		0.51 (169)
p_{412}	Variance	0.61	0.66	0.66	0.66	0.78	0.44	0.89 (25)
	SNR	0.49	0.74	0.74	0.74	0.57		0.81 (11)

		n_{VD}	Shannon coding	Huffman coding	Hamming coding	$2n_{\text{VD}}$	total	Optimal
	Skewness	0.69	0.55	0.55	0.55	0.45		0.78 (10)
	Kurtosis	0.6	0.59	0.59	0.59	0.62		0.87 (123)
	Entropy	0.49	0.11	0.11	0.11	0.18		0.49 (9)
	ID	0.67	0.66	0.66	0.66	0.67		0.81 (110)
	Negentropy	0.81	0.66	0.66	0.66	0.63		0.85 (66)
p_{42}	Variance	0.87	0.87	0.87	0.87	0.62	0.76	0.87 (9)
	SNR	0.4	0.6	0.6	0.6	0.52		0.86 (51)
	Skewness	0.48	0.66	0.66	0.66	0.7		0.79 (128)
	Kurtosis	0.87	0.88	0.88	0.88	0.88		0.89 (11)
	Entropy	0.71	0	0	0	0.48		0.81 (140)
	ID	0.87	0.85	0.85	0.85	0.87		0.87 (22)
	Negentropy	0.56	0.47	0.47	0.47	0.5		0.8 (40)
p_{43}	Variance	0.26	0.2	0.2	0.2	0.07	0.16	0.26 (9)
	SNR	0	0	0	0	0		0.16 (169)
	Skewness	0.21	0.19	0.19	0.19	0.2		0.21 (22)
	Kurtosis	0.21	0.21	0.21	0.21	0.21		0.24 (14)
	Entropy	0.03	0.14	0.14	0.14	0.21		0.24 (93)
	ID	0.19	0.2	0.2	0.2	0.2		0.28 (10)
	Negentropy	0.1	0.1	0.1	0.1	0		0.18 (40)
$\tilde{q} =$ Number of selected bands		9	13	12	13	18	169	
p_{511}	Variance	0.86	0.84	0.84	0.84	0.84	0.83	0.86 (10)
	SNR	0.82	0.68	0.69	0.68	0.69		0.83 (50)
	Skewness	0.85	0.75	0.75	0.75	0.78		0.86 (10)
	Kurtosis	0.86	0.85	0.85	0.85	0.86		0.86 (9)
	Entropy	0.79	0.65	0.65	0.65	0.8		0.85 (131)
	ID	0.84	0.84	0.84	0.84	0.84		0.84 (14)
	Negentropy	0.83	0.83	0.83	0.83	0.83		0.84 (40)
p_{521}	Variance	1	1	1	1	1	1	1 (13)
	SNR	1	1	1	1	1		1 (10)
	Skewness	0.91	0.9	0.9	0.9	0.89		1 (44)
	Kurtosis	1	1	1	1	1		1 (10)
	Entropy	0	0.9	0.9	0.9	0.94		1 (168)
	ID	1	1	1	1	1		1 (10)
	Negentropy	1	1	1	1	1		1 (21)
p_{52}	Variance	0.72	0.81	0.82	0.81	0.95	0.94	0.96 (21)
	SNR	0.97	0.97	0.97	0.97	0.97		0.97 (12)
	Skewness	0.89	0.88	0.88	0.88	0.88		0.93 (169)
	Kurtosis	0.82	0.74	0.74	0.74	0.72		0.96 (54)
	Entropy	0.43	0.9	0.9	0.9	0.87		0.94 (169)
	ID	0.89	0.87	0.88	0.87	0.82		0.96 (56)
	Negentropy	0.96	0.97	0.97	0.97	0.97		0.97 (19)
p_{53}	Variance	0	0.15	0.15	0.15	0.13	0.14	0.15 (13)
	SNR	0.13	0.04	0.02	0.04	0		0.14 (168)
	Skewness	0	0	0	0	0		0.16 (68)
	Kurtosis	0	0	0	0	0		0.19 (27)
	Entropy	0.02	0.27	0.21	0.27	0.18		0.27 (13)
	ID	0	0.07	0.08	0.07	0.15		0.17 (19)
	Negentropy	0	0.08	0.05	0.08	0.09		0.17 (36)

correlated. As a result, redundant bands are also very likely selected and may not provide any benefit but rather conflicting information. This issue can be resolved by band de-correlation (BD) to be discussed in Chapter 23. Last but not least, as shown in Chapter 20, VD can appropriately estimate the number of dimensions to be retained after dimensionality reduction, q. But when it comes to band selection, VD seems to under-estimate the value of $\tilde{q}$. In this case, DDA provides a better estimate of $\tilde{q}$.

22.6 Conclusions

The concept of DDA revolutionizes the traditional fixed-size band allocation in the sense that the former can vary band numbers according to various applications as opposed to the latter which uses a fixed number of bands for BS. A similar idea was also previously explored for a single hyperspectral signature in hyperspectral data Wang and Chang (2007). Its idea is derived from variable-length coding widely used in source coding where different source alphabets require different coding lengths for their own code words in accordance with their occurrence frequencies. This same idea can be applied to hyperspectral band selection provided that each source alphabet and its coding length are interpreted as a hyperspectral signature such as an endmember and the number of bands used to characterize the signature. Within this context various hyperspectral signatures will require their own variable numbers of bands as well as different bands to better describe their spectral characteristics (see Chapter 27). This is particularly true for LSMA where the signatures used to form a signature matrix to perform spectral unmixing certainly exhibit different spectral profiles for their own characterization. As a result, each signature used by LSMA has a different degree of difficulty with discrimination for its own identity and thus, requires a different set of bands to characterize its own spectral profile. The development of DDA arises from this need and allows users to allocate band dimensionality for individual hyperspectral signatures as demonstrated in the experiments conducted in this chapter.

23

Progressive Band Selection

Both progressive spectral dimensionality process (PSDP) in Chapter 20 and progressive band dimensionality process (PBDP) in Chapter 21 are developed to prioritize spectral dimensions/bands and process spectral dimensions/bands in the context of progressive spectral dimension/band dimensionality expansion and reduction via dimensionality prioritization (DP)/band prioritization (BP). However, there is a key difference between PSDP and PBDP. In PSDP, "*data compaction*" is performed via a transformation of the original data space into a spectral-transformed component space where spectral components are of major interest and each spectral component is specified by a projection vector as a spectral component dimension. Such projection vectors are obtained by linearly combining all spectral band dimensions across the entire range of wavelengths. In contrast to PSDP, PBDP performs "*data reduction*" by retaining only those bands that are of interest and discarding the rest. As a result, there is no data processing involved in PBDP as it is in PSDP that processes the entire data cube by a transformation. This is why PSDP requires projection vectors to specify spectral components while PBDP does not, since the bands in PBDP can be considered a counterpart of projection vectors in PSDP. However, it is worth noting that projection vectors are completely different from spectral bands because the bands are acquired with individual and separate wavelengths with interband correlation yet to be explored, whereas projection vectors are obtained by a transformation using interband correlation. This crucial distinction leads to a new concept of progressive band selection (PBS), which implements PBDP in conjunction with band de-correlation (BD) whose counterpart is not found in PSDP (refer to Chapter 20). The need of BD arises from the fact that some highly prioritized bands may also share much information in common if their acquired wavelengths are very close in range. Such interband correlation among bands needs to be addressed by BD in PBDP, but this is not an issue for DR used in PSDP. So, despite that a concept similar to PBS can also be derived for progressive dimensionality reduction (PDR), Safavi (2010) has shown that not much gain could be benefited from PDR since much spectral dimensionality correlation has already been removed by mutual original projection vectors implemented by PSDP. Accordingly, this chapter focuses only on PBS; readers may refer to Safavi (2010) for the treatment of PDR.

23.1 Introduction

The PBDP (Chapter 21) coupled with the dynamic dimensionality allocation (DDA) (Chapter 22) paves the way for a new approach to band selection (BS), referred to as progressive band selection (PBS) to be studied in this chapter. The idea of PBS emerged from the issue of implementing

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

PBDP for BS where high prioritized bands may also have high correlation. If PBDP is directly used for BS, it is very much likely that if a band is selected for its high priority, other bands highly correlated with this particular band such as its adjacent bands may also have high priorities. As a result, they will also be selected according to their high priority scores. In order to avoid such a situation, these bands must be removed prior to BS. In other words, once a band is selected, all other bands having high correlation with this particular band should not be selected. Using BD to resolve this issue for BS was investigated in Chang et al. (1999), where an information divergence-based band de-correlation approach was developed for this purpose. The PBS proposed in this chapter basically takes advantage of this approach by including BD as a preprocessing step prior to PBDP in which case the latter will not repeatedly select bands that have high interband correlation with bands already selected. Two approaches to BD are developed to effectively remove the interband correlation, spectral measure-based band de-correlation and orthogonalization-based band de-correlation. Since BP prioritizes each of the bands without taking into account interband correlation, it can be considered as an intraband criterion. On the other hand, BD is used to measure only the correlation between two bands and, thus, it can be viewed as an interband criterion. Accordingly, when PBDP that uses BP as an intraband criterion is implemented in conjunction with BD that is an interband criterion to perform BS, the resulting BS is called PBS. The only remaining issue is how many bands are required to be selected for PBS. The DDA developed in Chapter 22 provides a solution to this issue.

23.2 Band De-Corrleation

The PBDP in Chapter 21 is designed to prioritize all bands in accordance with their contained information measured by a custom-designed band prioritization criterion. It does not consider the case that bands with high priority scores may also be highly correlated. As a consequence, when PBDP is directly applied to BS, some of highly correlated bands may be also selected simply based on their priority scores. Such a dilemma can be avoided by BD that allows PBS to select bands not only with high priorities but also with interband correlation as least as possible. The idea of BD is to consider a band image with a size of $M \times N$ pixels as an MN-dimensional band vector by concatenating pixels line by line from top to bottom. With this interpretation, two band images can be represented by two vectors with same dimensionality. The correlation between two band images can now be measured by discrepancy between their corresponding vectors. Two BD criteria, spectral measure-based BD and orthogonalization-based BD, are developed and described in the following sections.

23.2.1 Spectral Measure-Based BD

Since a band image can be represented as a band vector, the correlation of two band images can be interpreted as mutual discriminability of their representing band vectors. Accordingly, all the signature-based hyperspectral measures developed in Section 16.2 can be used for this purpose. Two measures are of particular interest: spectral angle mapper (SAM) (Section 16.2.2) and spectral information divergence (SID) (Section 16.2.4). It is worth noting that the information divergence (ID) for BD introduced by Chang et al. (1999) is actually SID. A general approach to spectral measure-based BD is described in the following sections.

Algorithm for Spectral Measure-Based BD

1. Let $\{\mathbf{B}_l\}_{l=1}^{L}$ be a total set of bands to be de-correlated. Let $\mathbf{b}_l$ be the corresponding band vector representing the lth band image, and $\mathbf{B}_l$ and the resulting set of band images be denoted by $\Omega_L = \{\mathbf{b}_l\}_{l=1}^{L}$.

2. Set $k = 1$ and $\Omega_k = \{\mathbf{b}_l\}_{l=1}^{n_1}$ with $n_1 = 1$.
3. At the kth iteration, select a band $\mathbf{b}_l$ in $\Omega_L - \Omega_k$ and measure the band correlation

$$\max_{\mathbf{b}_k \in \Omega_k} \mathrm{BD}(\mathbf{b}_l, \mathbf{b}_k) \tag{23.1}$$

   where $\mathrm{BD}(\mathbf{b}_l, \mathbf{b}_k)$ is a criterion used to measure the band correlation between their representing band images $\mathbf{B}_j$ and $\mathbf{B}_k$. The BD measure between $\mathbf{b}_l$ and all $\mathbf{b}_k$ in Ω_{BD} defined in (23.1) can be any hyperspectral measure. For example, if a particular measure is used for BD such as SID, then BD($\mathbf{B}_l$, $\mathbf{B}_k$) in (23.1) is defined as $\mathrm{BD}(\mathbf{b}_l, \mathbf{b}_k) = \mathrm{SID}(\mathbf{b}_l, \mathbf{b}_k)$ and BD using ID as a measure is referred to as ID-BD.
4. If $\mathrm{BD}(\mathbf{b}_l, \mathbf{b}_k) > \varepsilon$, then $\Omega_{k+1} = \Omega_k \cup \{\mathbf{b}_l\}$; check if $k = L$. If it is, go to step 6. Otherwise, let $\Omega_{k+1} = \Omega_k$ and $k \leftarrow k + 1$. Go to step 3.
5. If $\mathrm{BD}(\mathbf{b}_l, \mathbf{b}_k) < \varepsilon$, then $\Omega_{k+1} = \Omega_k$; check if $k = L$. If it is, go to step 6. Otherwise, let $k \leftarrow k + 1$. Go to step 3.
6. The resulting Ω_k, denoted by Ω_{BD}, is the final set of selected bands where the interband correlation among all bands in Ω_{BD} must be greater than ε.

23.2.2 Orthogonalization-Based BD

As an alternative to the spectral measure-based BD described above, the principle of orthogonality can be used as a measure of BD. In other words, two bands have the best possible de-correlation when they are mutually orthogonal. So, if we follow the same treatment in Section 23.2.1 by considering a band image as a band vector, orthogonalizing of two band images can be carried out by orthogonalizing their corresponding band vectors. In light of this interpretation, the well-known Gram-Schmidt orthogonalization procedure can be used for the purpose of orthogonalization, referred to as Gram-Schmidt orthogonalization-based BD (GSO-BD), which is described in the following section.

Algorithm for Gram-Schmidt Orthogonalization-Based BD

1. Let $\Omega_L = \{\mathbf{B}_l\}_{l=1}^{L}$ be a total set of bands to be de-correlated. Convert band images $\{\mathbf{B}_l\}_{l=1}^{L}$ to band vectors $\{\mathbf{b}_l\}_{l=1}^{L}$. Let ε be a prescribed threshold.
2. Apply the following Gram-Schmidt orthogonalization procedure to $\{\mathbf{b}_j\}_{l=1}^{L}$ by

$$\begin{aligned}
\boldsymbol{\beta}_1 &= \mathbf{b}_1 \\
\boldsymbol{\beta}_2 &= \mathbf{b}_2 - \mathrm{proj}_{\boldsymbol{\beta}_1}(\mathbf{b}_2) \\
\boldsymbol{\beta}_3 &= \mathbf{b}_3 - \mathrm{proj}_{\boldsymbol{\beta}_1}(\mathbf{b}_3) - \mathrm{proj}_{\boldsymbol{\beta}_2}(\mathbf{b}_3) \\
&\vdots \\
\boldsymbol{\beta}_k &= \mathbf{b}_k - \textstyle\sum_{j=1}^{k-1} \mathrm{proj}_{\boldsymbol{\beta}_j}(\mathbf{b}_j) \quad \text{for} \quad k = 1, 2, \ldots, L
\end{aligned} \tag{23.2}$$

   where the projection operator is defined by

$$\mathrm{proj}_{\mathbf{u}_k}(\mathbf{b}_k) = \frac{\langle \mathbf{b}_k, \boldsymbol{\beta}_k \rangle}{\langle \boldsymbol{\beta}_k, \boldsymbol{\beta}_k \rangle} \boldsymbol{\beta}_k \tag{23.3}$$

   and $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T \mathbf{y}$ is an inner product of $\mathbf{x}$ and $\mathbf{y}$.
3. Normalize the orthogonal vectors $\{\boldsymbol{\beta}_l\}_{l=1}^{L}$ to $\mathbf{u}_k = 1$.
4. Convert all orthogonal band vectors $\{\mathbf{u}_l\}_{l=1}^{L}$ to band images $\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2, \ldots, \hat{\mathbf{B}}_L$, denoted by $\hat{\Omega}_L = \{\hat{\mathbf{B}}_l\}_{l=1}^{L}$. The $\hat{\Omega}_L$ is the final band-decorrelated set produced by the Gram-Schmidt orthogonalization procedure.

The following are the three noteworthy comments on GSO-BD:

1. Technically speaking, in order to properly orthogonalize two band images, we need to find the normal vectors that are perpendicular to their respective band images via solving a set of so-called normal equations. Using the band vectors for orthogonalization may not be optimal since the band vectors may miss spatial correlation when consecutive lines of pixels are concatenated. Nonetheless, it is a simplest way to accomplish the goal.
2. The principle of orthogonality is an optimal de-correlation criterion in the context of second-order statistics. It does not necessarily de-correlate high-order statistics as the spectral measure-based BD does. Consequently, the spectral measure-based BD generally performs better than the orthogonalization-based BD.
3. Unlike the spectral measure-based BD that removes redundant bands from the original band set to obtain a smaller band set Ω_{BD}, the GSO-BD does not remove bands but only orthogonalizes all the original bands to obtain a new orthogonal band set $\hat{\Omega}_L$ with the same total number of bands L.

23.3 Progressive Band Selection

The PBDP (Chapter 21) allows users to select bands progressively back and forth for band dimensionality reduction and expansion at discretion without actually determining the number of bands $\tilde{q}$ needed to be selected. In the traditional BS, the value of this p must be first determined and then fixed at a constant value during the whole data processing. When the value of $\tilde{q}$ is not desired and a new value is needed, the entire BS process must be reimplemented where the previous results obtained for old values of $\tilde{q}$ cannot be reused for the new value of $\tilde{q}$ because new bands will be selected by solving a new set of BS optimization problem. This type of BS is referred to as fixed-dimensionality BS (FDBS) to emphasize the role of p to be fixed at a constant value instead of using static band selection (SBS), discussed in Chapter 22. Most importantly, such FDBS may not be applicable or effective in hyperspectral data exploitation. For example, as discussed in Chapter 22, various material substances have their own spectral characteristics in signature profiles that may present different levels of difficulty for data analysis. In this case, these material substances may require different values of $\tilde{q}$ to respond to various applications so as to achieve better results. To resolve this issue, the number of bands to be selected, $\tilde{q}$, must be considered as a variable instead of a fixed value assumed by standard FDBS so that the parameter $\tilde{q}$ can adapt to different applications. However, to avoid formidable computational complexity resulting from repeated implementation of FDBS using various values of $\tilde{q}$, progressive band selection (PBS) offers a feasible strategy that enables users to perform BS progressively as they desire without reselecting bands every time the value of p is changed.

It should be noted that PBS differs from FDBS in many aspects. First and foremost, it is the way of selecting bands. FDBS is performed by fixing the number of bands to be selected, $\tilde{q}$ and then selecting a $\tilde{q}$-band subset from the entire set of bands by solving an optimization problem. In contrast, PBS does not solve a BS optimization problem. Instead, it makes use of PBDP to perform BS in a forward manner (i.e., band expansion) or in a backward manner (i.e., band reduction) via band prioritization (BP) measured by a specific criterion. As a result, its performance is determined by the nature in progression and not the value of $\tilde{q}$ that determines FDBS performance. Second, when the value of $\tilde{q}$ is changed, FDBS must be reimplemented because the bands to be selected are generally different. This drawback is remedied by PBS where the previously selected bands are always included as a part of band sets produced by subsequent band selections in a progressive process so as to perform band expansion or reduction according to the calculated band priorities. Last but not least, in order for PBS to work effectively, it requires band de-correlation (BD) to remove highly correlated bands prior to BS since highly prioritized adjacent bands may also share much information in common. If one band is highly prioritized, so are its adjacent bands. In this case, if one of them is selected, other bands may be considered as redundant due to significant overlapped interband information about them. However, BD is never an issue in FDBS due to the fact that it solves an optimization problem for a given value of $\tilde{q}$.

The PBS presented in this section can be implemented in conjunction with the dynamic dimensionality allocation (DDA) developed in Chapter 22 as a two-stage process in two different fashions. One is to implement BP in the first stage followed by BD in the second stage. In other words, PBS first implements BP to prioritize and rank all bands and then uses BD to remove bands with highly interband correlation with previously selected bands to perform BS. Another way is to reverse the two processes of BP and BD by first implementing BD and then BP. However, a major disadvantage resulting from this approach is that an initial band to be selected as the first band for BD is crucial since the original bands are not prioritized or ranked.

23.3.1 PBS: BP Followed by BD

The PBS approach presented in this section, referred to as BP/BD-PBS, is actually PBDP implemented in conjunction with BD and DDA where the DDA is used to determine the number of bands needed to be selected instead of using the range of $[n_{\text{VD}}, 2n_{\text{VD}}]$ by PBDP. The details of implementing BP/BD-PBS are summarized in the following.

Algorithm for BP/BD-PBS

1. Use BP to prioritize and rank all bands, Ω_L.
2. Implement BD to remove bands from Ω_L to obtain a new set of prioritized bands, Ω_{BD}, in such a manner that all bands in Ω_{BD} have interband correlation greater than a prescribed threshold ε.
3. Apply either PBDE or PBDR to select a final desired set of bands from Ω_{BD}.
4. Use DDA to determine how many bands should be processed to terminate the PBS.

23.3.2 PBS: BD Followed by BP

A second approach to PBS in conjunction with DDA is to implement BD followed by BP, referred to as BD/BP-PBS, that reverses the two processes of BP and BD as carried out by the BP/BD-PBS described above. It first de-correlates all bands and then prioritizes the resulting de-correlated bands to be used for PBS.

Algorithm for BD/BP-PBS

1. Select a criterion such as entropy to find a band, denoted by $\mathbf{B}_{l^*}$, as the first band for BD and an error threshold ε.
2. Form a new band set Ω_L^* whose bands are ordered beginning with the l^*th band and increasing as well as decreasing one band at a time in both directions until reaching both ends. It should be noted that if one end is reached first, then the ordering process will still continue until another end is reached. For example, if the end of $l = 1$ is reached first, the bands in Ω_L^* will be ordered by $l^*, l^* - 1, l^* + 1, l^* - 2, l^* + 2, \ldots, 1, 2l^*, 2l^* + 1, 2l^* + 2, \ldots, L$.
3. Implement BD to remove bands from Ω_L^* to obtain a new set of prioritized bands, denoted by Ω_{BD}, in such a manner that all the bands in Ω_{BD} have interband correlation greater than a prescribed threshold ε.
4. Use BP to prioritize and rank all bands in Ω_{BD}.
5. Apply either PBDE or PBDR to select a final desired set of bands from Ω_{BD}.
6. Use DDA to determine how many bands should be processed to terminate the PBS.

A main difference between BD/BP-PBS and BP/BD-PBS is that the former requires selecting an appropriate band as an initial band for BD, whereas the latter does not because all the bands have already been prioritized prior to BD in BP/BD-PBS. So, the disadvantage of using BD/BP-PBS is that it needs to find a good criterion to select an initial band that has a significant impact on the subsequent BS. However, one way to alleviate this dilemma is to use GSO-BD to orthogonalize and normalize all band images to unit vectors and then select one with maximal entropy as an initial band. In this case, BD and selection of an initial band can be done in one-shot operation.

Finally, it should be noted that the effectiveness of PBS in conjunction with BD is determined by three parameters: the prescribed threshold ε that is used to determine how close interband correlation would be, the criterion used for BP, and the criterion used by BD. Specifically, ε is an empirical choice that generally varies with a chosen criterion from case to case and application to application. In addition, BD is not generally required for progressive dimensionality reduction since the data information compacted in a low dimensional reduced data space is always contained or embedded in a higher dimensional reduced data space.

23.4 Experiments for Progressive Band Selection

When BS is implemented with no prior knowledge, the uniform BS is generally preferred because it attempts to select bands with interband correlation as least as possible by spreading bands as widest as possible in terms of wavelengths. The same reason also applies to PBDP, which prevents it from being used for BS due to the fact that bands highly prioritized by PBDP may also be highly correlated. Consequently, if a spectral band with a high priority score is selected, its neighboring bands may also have similar high priority scores and, thus, the chance for these bands to be selected is also very high. So, in order for PBDP to be implemented as PBS, we need to take care of this issue by including a preprocessing of BD removing highly correlated bands in conjunction with DDA determining the number of bands to adapt to different applications. In this section, the same three applications conducted for experiments in Chapters 20–22 are also performed for PBS for a comparative study and analysis. Since Gram-Schmidt orthogonalzation is a second-order statistics-based criterion that is not as effective as a high-order statistics-based ID as shown in Liu (2011), only ID is used for BD, referred to as ID-BD, for PBS. Furthermore, according to our extensive experiments, PBS with BD/BP generally does not perform as effectively as PBS with BP/BD. So, experiments were performed only for PBS with BP/ID-BD.

23.5 Endmember Extraction

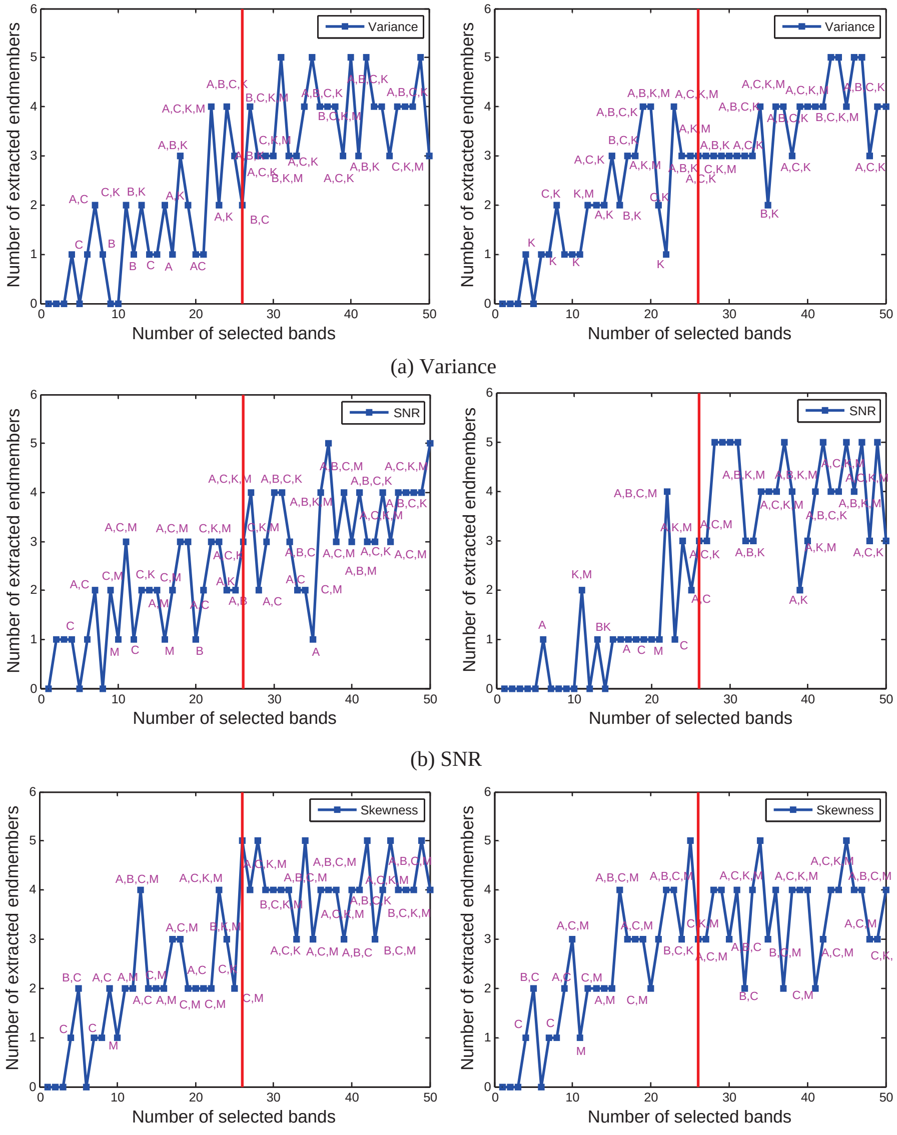
The reflectance Cuprite data scene in Figure 1.12 was used for experiments by PBS with BP/ID-BD to investigate an application in endmember extraction where a threshold value ε for ID-BD was empirically set to 0.01 to remove correlated bands. After BP/ID-BD, only 95 spectral bands were retained. Table 23.1 tabulates the first 50 spectral bands after BP/ID-BD using 7 BP criteria: variance, SNR, skewness, kurtosis, entropy, ID, and negentropy. Interestingly, the interband correlation for this scene is clearly characterized by second-order statistics rather than by high-order statistics (HOS), where only a few bands are removed by ID-BD using three specific HOS criteria: kurtosis, ID, and negentropy.

Figure 23.1(a)–(g) shows the results of IN-FINDR-extracted mineral signatures among A, B, C, K, and M using BP/ID-BD-generated spectral bands given in Table 23.1, where the figures on the right and the left were obtained by PBS with/without using ID-BD, respectively. Also in Figure 23.1, a vertical line in each figure shows a cutoff threshold for $\tilde{q}$ determined and set by DDA = 26 that was the largest value obtained by Huffman coding among five mineral signatures.

Comparing Figure 23.1 with Figure 21.8, the results produced by BP/ID-BD-generated bands are different from the results the generated by using BP without BD. However, the improvement of endmember extraction was not really significant.

Table 23.1 Seven BP criteria to produce the first 50 bands after BP/ID-BD with bands removed by ID-BD for Cuprite data by setting $\varepsilon = 0.01$

	50 BP/BD bands	Bands removed by ID-BD
Variance	87/78/98/189/75/94/96/95/67/123/124/122/120/125/117/114/129/112/54/131/46/40/187/108/133/102/106/25/20/188/18/141/143/142/145/140/146/144/139/147/16/138/149/137/150/151/152/15/153/154	85/88/86/89/84/91/80/90/92/83/82/79/93/81/99/97/77/76/100/74/73/72/101/71/70/69/68/121/66/65/119/64/118/63/127/126/128/62/116/115/61/113/59/60/58/57/55/56/130/111/53/52/51/50/110/49/48/47/41/44/43/45/42/39/38/109/37/36/132/35/34/33/135/32/134/107/31/29/30/28/27/26/103/104/24/23/105/22/21/19/17
SNR	68/26/89/20/18/53/76/39/16/15/14/114/120/112/117/124/122/12/123/125/11/129/9/97/131/108/96/8/94/95/7/133/6/5/106/149/142/150/146/147/151/4/152/153/143/141/145/3/144/155	71/67/69/70/72/66/73/65/74/25/27/23/24/28/22/64/21/75/63/87/88/85/86/91/19/83/84/42/29/90/52/17/49/43/54/51/38/50/62/45/35/57/32/55/41/31/37/47/33/48/34/44/113/36/56/82/118/13/93/119/46/30/115/126/121/92/111/127/110/128/116/40/109/61/10/130/98/77/99/100/81/132/60/58/59/101/80/134/107/135/79/78/104/148/103
Skewness	2/1/3/4/5/6/7/8/9/11/12/14/184/15/185/183/181/16/179/178/180/182/177/170/171/18/159/169/20/158/176/157/186/172/156/25/155/175/173/49/174/160/154/168/43/153/167/166/152/64	10/13/17/19/21/22/23/24/26/30/27/31/28/32/29/33/50/51/48/52/47/34/53/46/54/45/35/44/55/56/42/36/57/37/41/38/39/40/58/59/60/61/62/63
Kurtosis	2/3/1/4/5/6/7/8/9/185/181/183/184/179/178/180/170/182/177/11/169/171/12/176/172/14/159/158/168/157/156/175/155/15/154/160/167/173/153/174/166/16/152/186/151/165/150/164/18/161	10/13/17
Entropy	90/101/189/82/96/37/95/111/75/112/124/187/117/126/123/122/46/107/188/174/25/175/173/67/103/102/131/172/176/20/52/177/105/171/18/168/134/169/178/179/137/170/162/159/167/161/163/160/158/136	92/88/99/89/91/93/86/87/100/85/84/98/83/97/81/94/80/79/78/77/38/76/40/39/36/41/35/113/110/109/42/34/114/120/115/121/116/118/74/33/119/43/73/32/44/29/108/72/45/31/28/125/128/127/30/71/27/129/70/26/47/69/130/68/24/23/48/66/22/104/65/49/60/63/21/64/61/59/62/50/106/58/51/54/53/56/55/132/57/19/133/135/17
ID	1/2/3/5/153/154/152/4/151/150/9/6/8/185/144/146/7/11/139/140/155/145/149/184/156/12/182/183/147/142/180/143/165/138/181/164/186/141/160/157/167/14/170/166/163/158/169/179/15/161	10/148/13
Negentropy	2/3/1/4/5/6/7/8/9/185/181/184/183/179/178/180/11/170/182/177/171/169/12/176/14/159/172/158/157/15/168/156/175/155/154/16/160/173/167/153/174/186/166/152/151/165/18/150/164/161	10/13/17/19



(a) Variance

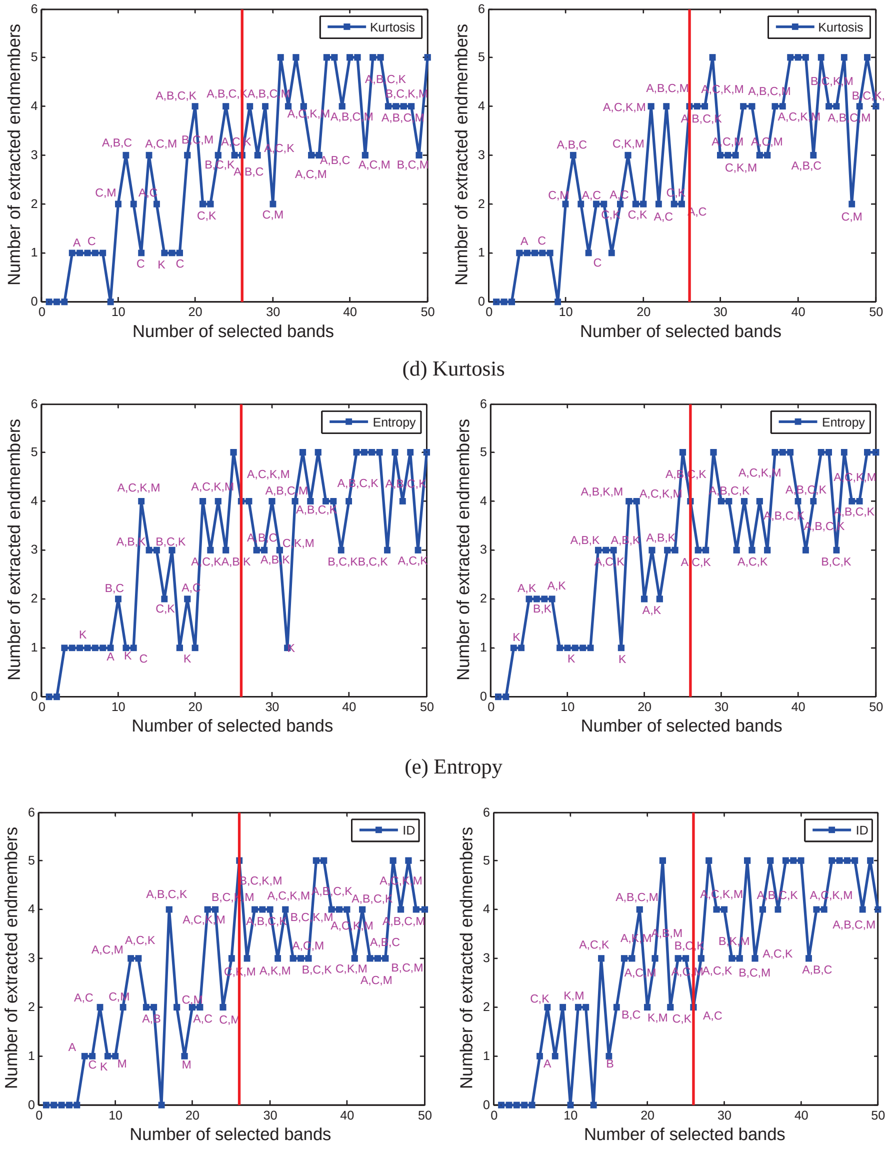
(b) SNR

(c) Skewness

Figure 23.1 IN-FIND-extracted endmembers from Cuprire scene using PBS with different seven BP criteria with DDA = 26 (Huffman coding) and figures on the right and the left are obtained by PBS with/without using ID-BD, respectively.

23.6 Land Cover/Use Classification

The Purdue Indiana Indian Pine test site in Figure 1.13 was used in experiments to demonstrate the maximum likelihood classification (MLC) performance resulting from PBS where once again seven BP criteria: variance, SNR, skewness, kurtosis, entropy, ID, and negentropy, were used for



(d) Kurtosis

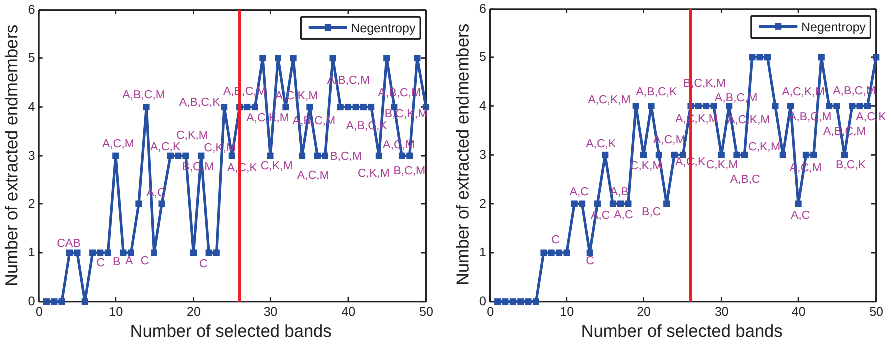
(e) Entropy

(f) ID

Figure 23.1 (*Continued*)

BP and SID was used as a BD criterion with ε set to 0.1. Table 23.2 tabulates the first 60 highest prioritized BD de-correlated bands and corresponding removed bands after applying ID-based BD.

Figure 23.2 shows MLC rates of 16 classes produced by operating PBS on seven different BP/BD prioritized bands where MLC is calculated based on (20.17) and the figures on the right and the left were obtained by PBS with/without using ID-BD, respectively. Also included in the last figure in Figure 23.2 is the overall averaged MLC rate among 16 classes. In order to find DDA, the means



(g) Negentropy

Figure 23.1 (*Continued*)

Table 23.2 First 60 highest prioritized BP/BD bands and corresponding removed bands after applying ID-based BD on seven BP results of Purdue data with $\varepsilon = 0.1$

	Decorrelated bands	Corresponding removed bands
Variance	29/42/32/33/34/12/10/9/37/69/8/7/6/36/117/5/4/3/89/2/1/100/162/155/156/150/111/169/109/170/159/174/173/171/175/172/176/151/177/79/178/179/180/149/154/110/181/182/183/102/108/142/184/185/186/188/187/189/191/192	28/27/26/25/30/41/24/23/31/43/22/44/39/21/48/49/50/20/45/51/52/19/53/38/47/18/46/17/16/15/14/13/11/54/40/35/55/56/70/68/67/66/64/71/65/72/73/63/57/74/62/116/118/115/75/119/114/122/120/61/121/125/123/126/124/113/127/128/129/130/131/76/132/90/133/88/134/91/87/135/94/85/86/60/97/96/84/136/83/112/93/137/98/82/59/95/138/92/139/58/99/140/161/163/164/165/160/166/81/167/168/101/141/157/158/77/80/78
SNR	28/118/22/68/165/155/159/175/171/170/174/173/156/176/172/177/41/32/178/179/150/34/111/180/12/151/181/109/183/182/185/154/184/149/10/186/97/142/187/9/110/188/37/108/8/189/101/191/7/190/192/36/6/195/193/153/194/148/152/196	27/26/29/30/116/117/25/115/119/114/24/124/126/122/125/120/123/23/121/127/128/130/129/132/131/21/133/113/134/20/135/31/136/19/137/138/18/66/112/67/163/164/70/69/65/166/161/64/162/139/72/167/71/53/160/73/52/17/74/63/168/16/51/169/50/56/140/55/54/49/62/15/48/75/57/33/14/43/44/42/157/47/158/13/45/61/141/38/46/39/11/76/40/98/35/99/100/96/88/60/89/90/87/85/84/83/95/59/91/82/94/58/92/93/86/102/81
Skewness	4/6/5/3/7/8/15/9/10/2/12/42/35/23/29/32/199/198/197/185/200/186/183/193/195/196/194/182/184/181/179/188/180/189/190/191/187/178/192/177/161/159/176/175/148/156/149/169/170/174/150/171/152/202/155/151/144/172/173/153	16/17/14/11/13/18/19/44/40/43/41/20/45/46/21/48/22/49/47/50/39/51/24/52/26/25/28/27/53/34/30/54/55/33/31/56/57/38/58/160/162/163/164/165/158/166/167/168/157
Kurtosis	95/80/3/2/79/37/4/36/99/5/6/77/1/7/39/202/8/9/35/201/145/10/104/14/103/144/102/200/146/23/32/29/105/199/147/198/197/196/194/143/106/131/193/195/107/190/188/	96/90/91/92/89/93/97/87/88/81/94/82/83/85/86/98/84/78/63/76/62/61/64/65/75/60/38/66/67/74/68/73/100/69/59/72/70/71/58/57/55/56/54/53/52/49/48/50/51/40/46/47/45/

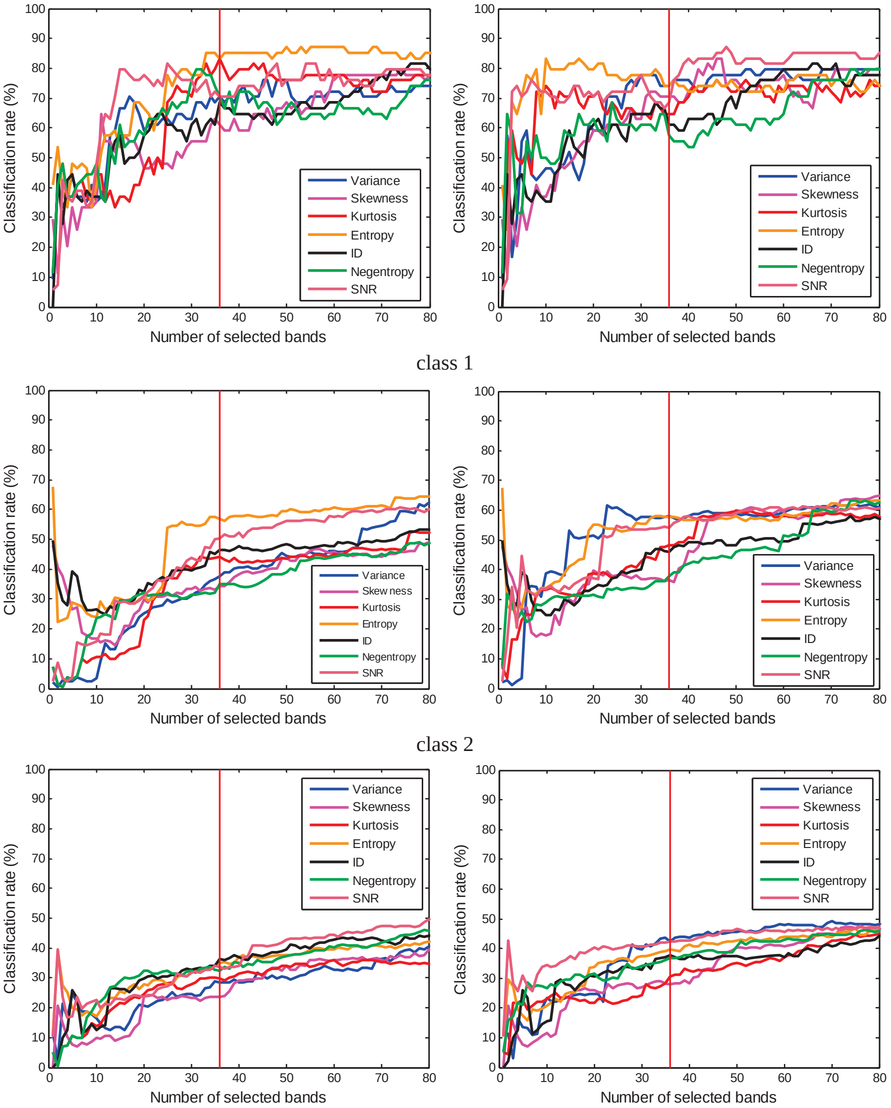
	185/191/192/142/187/183/186/189/184/182/181/179/180	43/42/44/41/15/101/16/11/13/12/17/18/19/34/20/21/22/24/25/33/26/31/27/28/30/130/132/133/129/128/134/127/135/136/126/125/124/123/137/140/122/138/121/141/139/120
Entropy	130/34/101/19/152/107/142/143/153/114/94/144/147/62/80/79/36/146/176/169/172/178/173/175/105/104/111/170/46/177/37/10/181/157/180/185/148/184/171/110/179/154/188/103/145/108/182/158/1/9/109/200/187/2/195/193/196/183/160/26	128/134/129/127/131/133/126/132/135/125/136/140/137/124/123/138/122/121/139/120/119/118/17/16/15/18/14/141/117/35/102/116/31/20/100/115/21/99/98/13/33/84/86/97/89/88/93/85/92/113/91/22/87/83/90/82/81/96/112/78/61/59/64/60/65/63/58/23/66/11/24/67/174/69/68/56/57/70/71/54/55/168/72/74/73/52/53/76/12/47/75/51/77/50/49/38/95/48/167/166/45/25/165/164/39/44/43/40/162/42/41/161/163
ID	202/4/6/5/199/201/29/194/7/3/2/155/8/32/1/156/198/189/149/192/145/150/200/159/151/106/197/190/191/186/37/196/95/193/103/160/41/195/9/183/187/77/105/104/158/110/182/109/108/10/36/184/146/188/80/179/79/185/181/154	28/30/27/42/26/43/39/40/45/49/38/44/75/48/46/163/73/74/76/50/161/51/53/63/71/72/47/55/67/52/70/68/162/54/60/56/66/57/59/65/69/64/78/81
Negentropy	95/3/149/150/151/155/156/186/154/4/158/160/183/159/182/157/181/184/180/179/185/167/109/178/108/169/189/177/148/176/174/175/171/170/173/187/153/172/112/111/191/188/192/152/190/195/110/193/194/6/139/196/5/197/198/107/106/143/2/29	96/90/91/92/89/93/97/163/162/164/161/165/166/168/113/114/115/88/87/116/117/118/119/81/120/141/142/94/121/140/138/122/137/123/124/136/125/135/126/134/127/133/128/132/82/129/131/130/83/85

of 16 classes are calculated to serve as 16 signatures required by DDA. A vertical line in each figure in Figure 23.2 shows a cutoff threshold for $\tilde{q}$ determined and set by DDA = 36 that was obtained from the Huffman coding using 16 class signatures.

Comparing Figure 23.2 with Figure 23.1, MLC rates of PBS using two second-order BP criteria, variance and SNR, benefited significantly from BD as the rates were increased in most of classes. It implies that second-order BP criteria were more appropriate than HOS BP criteria except the entropy criterion. Since MLC classification performance analysis is similar to that in Section 21.7.2, its detailed discussions will not be provided here.

The values of DDA given in Table 22.3 were further used in conjunction with PBS that used prioritized bands with/without ID-BD to perform MLC. Table 23.3 tabulates MLC rates of 16 classes resulting from PBS using various values of $\tilde{q}$ determined by VD, DDA, and optimal MLC performance where SID was used to measure spectral similarity and the numbers in parentheses in the last two columns are the true total numbers used for MLC. It should be noted that the column "total" in Table 23.3 is not 202 because some bands are already removed by BD. In this case, the numbers in parentheses were actual total numbers after BD. For example, for variance, skewness, and entropy, these numbers are 84, 86, and 85, respectively.

Comparing Table 23.1 with Table 23.2, it is seen that BP/BD improved MLC rates. Another observation is that variance and entropy criteria provided lower optimal band numbers, while skewness still required more bands in most classes.



class 1

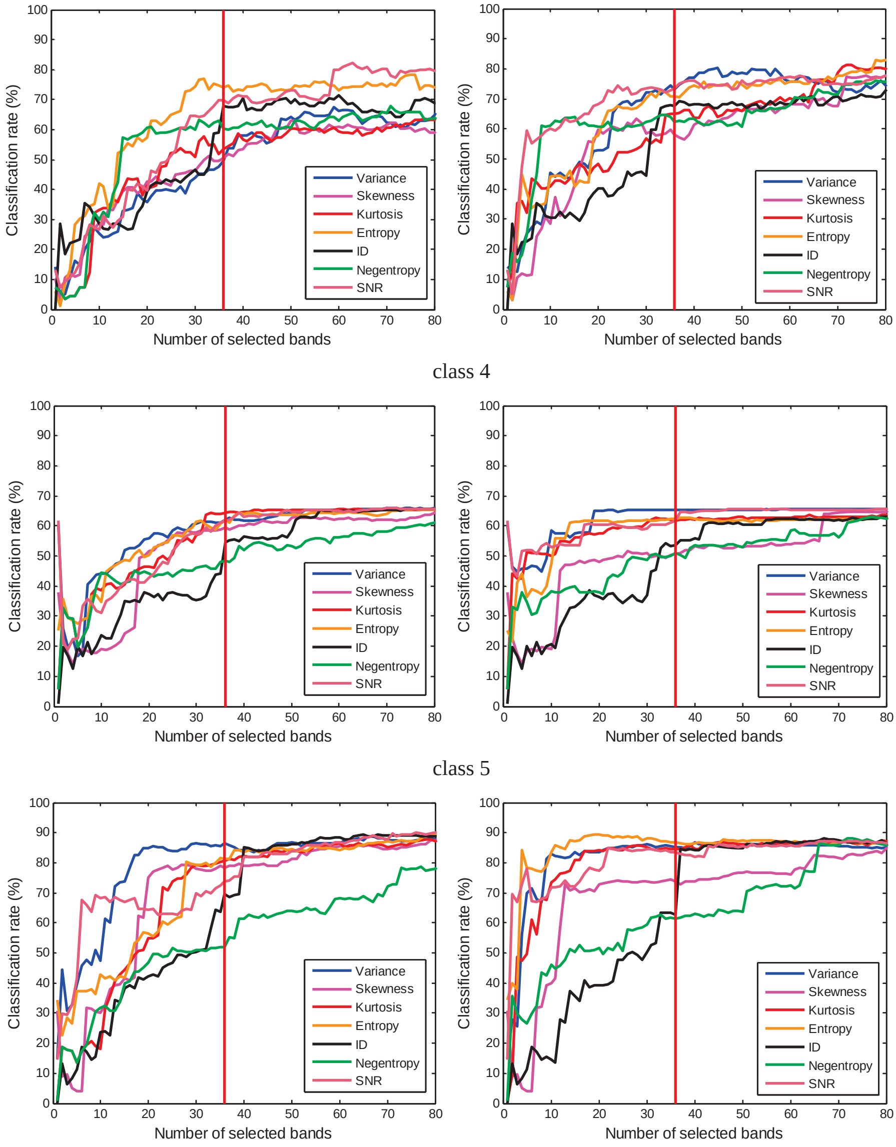
class 2

class 3

Figure 23.2 MLC rate of 16 classes using PBS prioritized bands for Purdue scene with DDA = 36 (Huffman coding) and figures on the right and on the left are obtained by PBS with/without ID-BD.

23.7 Linear Spectral Mixture Analysis

The HYDICE 15-panel scene in Figure 1.15 was used for experiments where the 19 R panel pixels, p_{11}, p_{12}, p_{13}, p_{211}, p_{221}, p_{22}, p_{23}, p_{311}, p_{312}, p_{32}, p_{33}, p_{411}, p_{412}, p_{42}, p_{43}, p_{511}, p_{521}, p_{52}, and p_{53}, provided by the ground truth in Figure 1.15(b) were used for quantitative analysis. According to the ground truth in Figures 1.15–1.17, nine signatures, $\mathbf{m}_1 = \mathrm{p}_1$, $\mathbf{m}_2 = \mathbf{p}_2$, $\mathbf{m}_3 = \mathbf{p}_3$, $\mathbf{m}_4 = \mathbf{p}_4$, and $\mathbf{m}_5 = \mathbf{p}_5$ in Figure 1.16 and background signatures, $\mathbf{m}_6 = \text{grass}$, $\mathbf{m}_7 = \text{road}$, $\mathbf{m}_8 = \text{tree}$, and



class 4

class 5

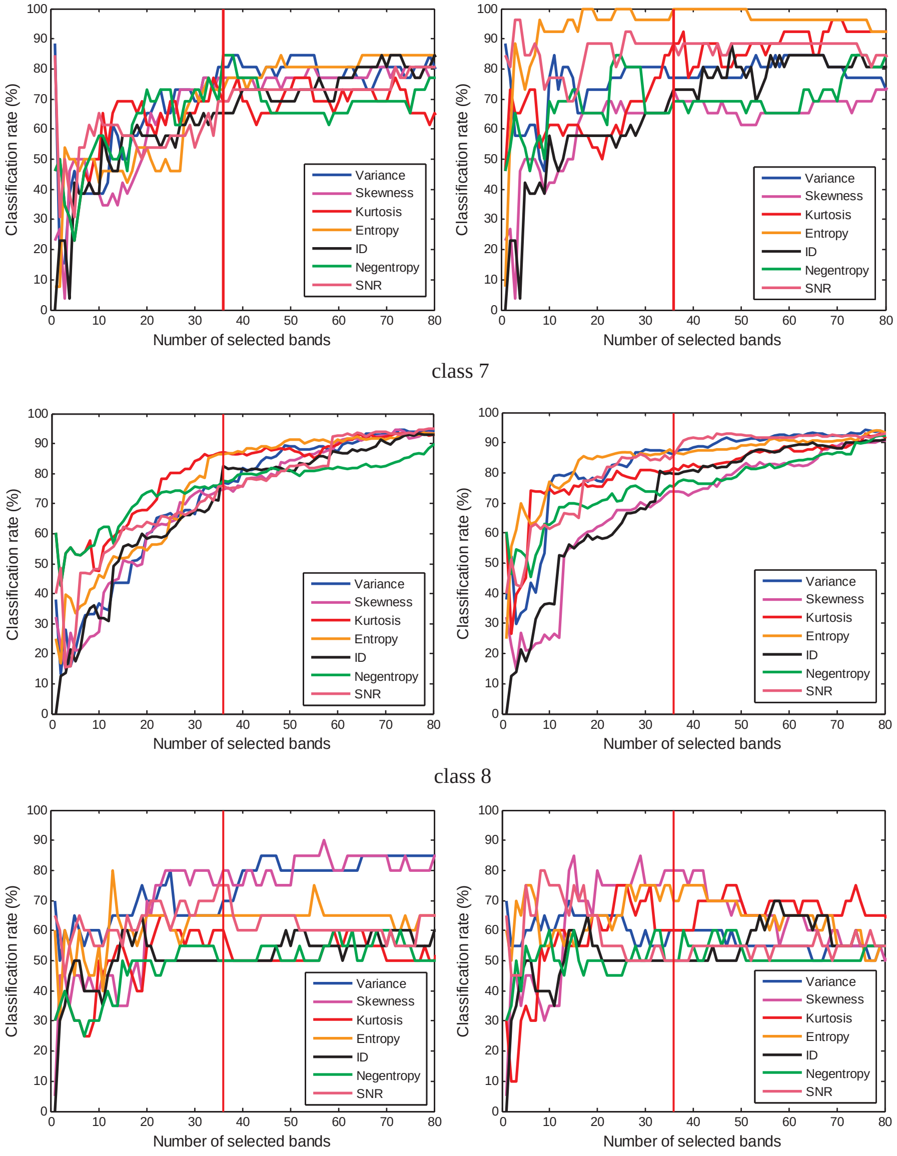
class 6

Figure 23.2 (*Continued*)

$\mathbf{m}_9$ = interferer in Figure 1.17, were used to unmix these 19 R panel pixels for panel detection and quantification.

Table 23.4 gives the first 35 highest prioritized “de-correlated bands” by ID-BD with the threshold value ε set to 0.1 and the corresponding “BD-removed bands” where seven BP criteria, variance, SNR, skewness, kurtosis, entropy, ID, and negentropy, are used to prioritize bands.

Using the nine signatures, $\mathbf{m}_1 = \mathrm{p}_1$, $\mathbf{m}_2 = \mathbf{p}_2$, $\mathbf{m}_3 = \mathbf{p}_3$, $\mathbf{m}_4 = \mathbf{p}_4$, and $\mathbf{m}_5 = \mathbf{p}_5$, $\mathbf{m}_6$ = grass $\mathbf{m}_7$ = road $\mathbf{m}_8$ = tree, and $\mathbf{m}_9$ = interferer, with the signature mean chosen to be the reference



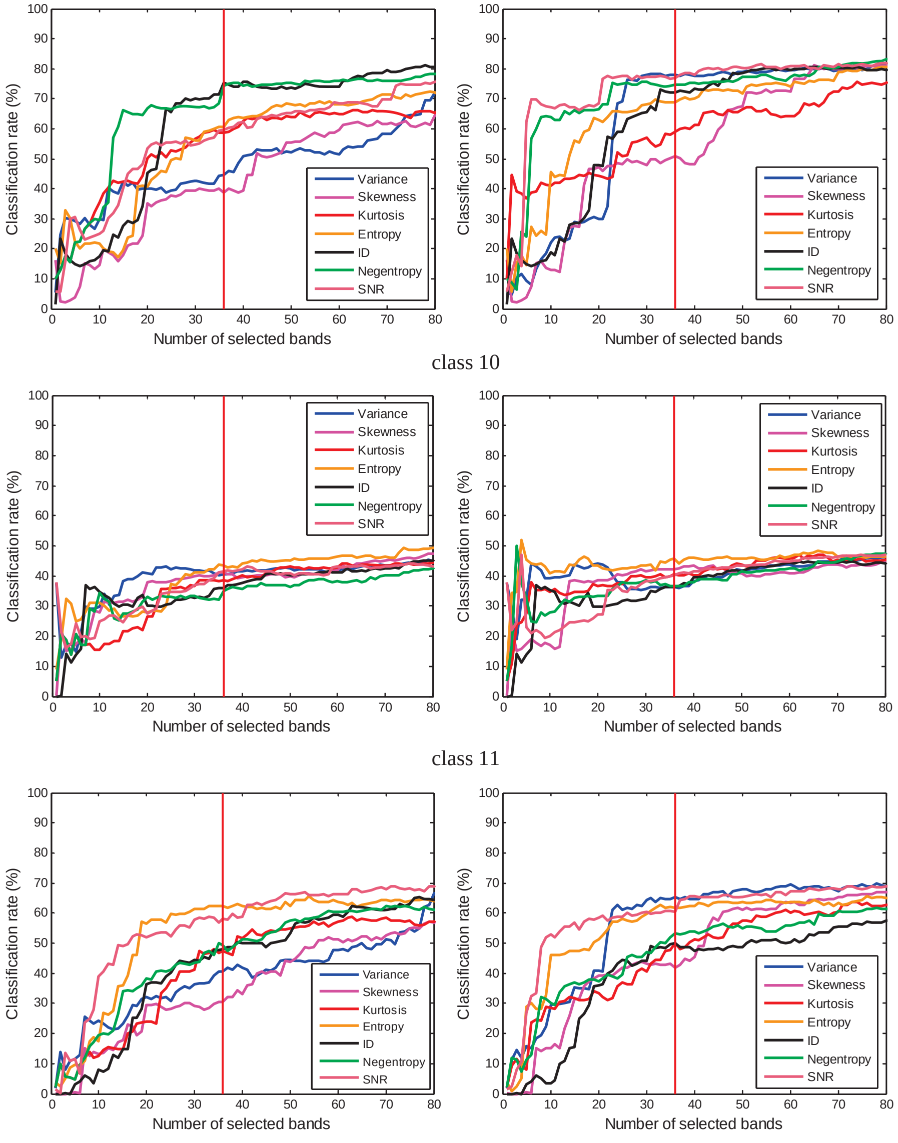
class 7

class 8

class 9

Figure 23.2 (*Continued*)

signature, DDA was calculated by Shannon coding, Huffman coding, and Hamming coding with SAM and SID used to measure spectral similarity and their results are given in Table 22.5. Table 23.5 gives FCLS-unmixed abundance fractions of 19 R panel pixels using prioritized bands with/without ID-BD via PBS with various values of $\tilde{q}$ determined by VD, DDA, and optimal FCLS performance. Also, note that the column "total" in Table 23.5 was



class 10

class 11

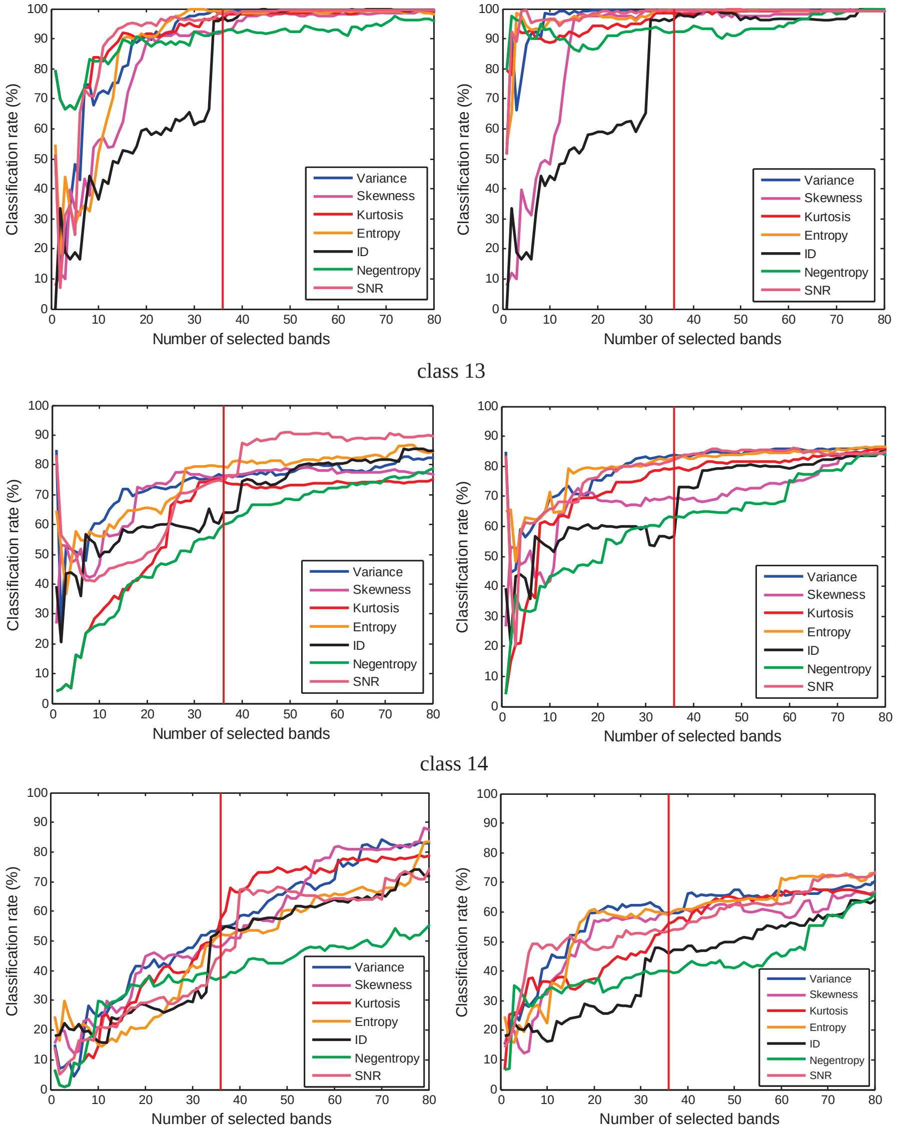
class 12

Figure 23.2 (*Continued*)

not 169 because some bands are already removed by BD. In this case, the numbers in parentheses were actual total numbers after BD.

From Table 23.5, the best estimate for $\hat{q}$ seems to be 9, 10, and 11.

Figure 23.3 shows FCLS-unmixed abundance fractions of the 19 R panel pixels using PBS with prioritized bands in Table 23.4 where the figures on the right and the left were obtained by PBS with/without using ID-BD, respectively.



class 13

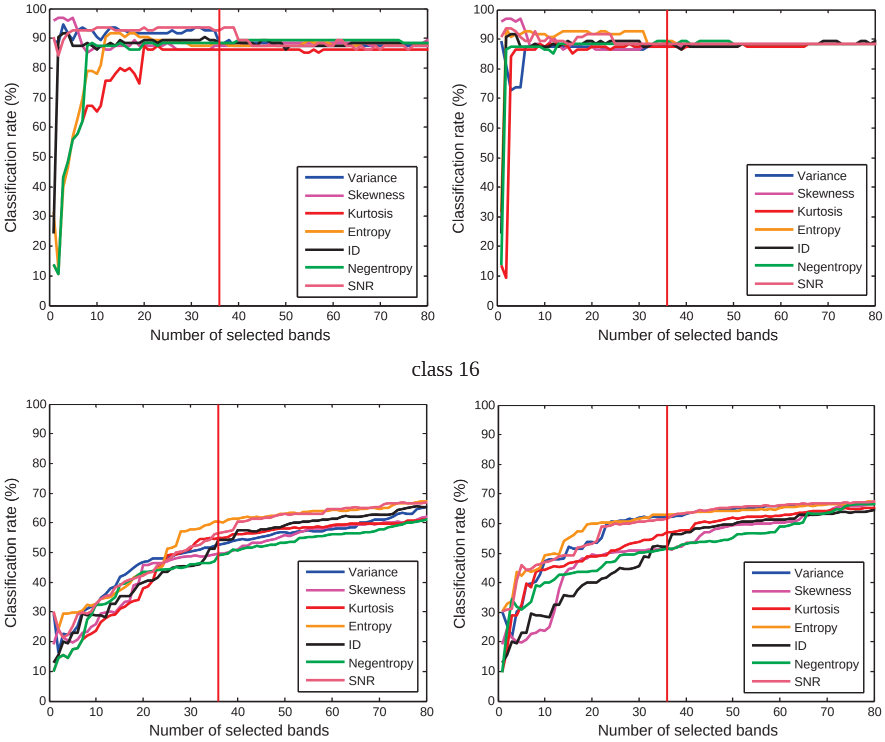
class 14

class 15

Figure 23.2 (*Continued*)

As shown in Figure 23.3, the performance of unmixing the 19 R pixels was significantly improved when ID-BD was implemented prior to FCLS. This implies that the ID-based BD effectively removed redundancy bands so as to increase FCLS performance. An interesting observation is worthwhile where some panel pixels using only a few bands performed better than those using more bands.

Finally, the ground truth of the 19 R panel pixels provided in Figure 1.15(b) was further used to calculate averaged detection rates of the panel pixels in each row according to ROC analysis in



class 16

Average

Figure 23.2 (*Continued*)

Table 23.3 MLC rates of 16 classes in the Purdue data using various values of $\tilde{q}$ determined by VD, DDA, and optimal FCLS performance

		n_{VD}	Shannon coding	Huffman coding	Hamming coding	$2n_{\mathrm{VD}}$	Total	Optimal
Class 1	$\tilde{q}$	29	36	36	34	58	202	
	Variance	74.07	74.07	74.07	74.07	79.63	79.63 (84)	79.63 (52)
	SNR	75.93	68.52	68.52	68.52	81.48	85.19 (83)	87.04 (48)
	Skewness	62.96	70.37	70.37	70.37	75.93	79.63 (86)	83.33 (46)
	Kurtosis	64.81	64.81	64.81	66.67	68.52	79.63 (86)	79.63 (83)
	Entropy	77.78	74.07	74.07	77.78	72.22	75.93 (85)	83.33 (17)
	ID	64.81	61.11	61.11	66.67	75.93	75.93 (90)	81.48 (66)
	Negentropy	59.26	57.41	57.41	64.81	62.96	79.63 (87)	79.63 (74)
Class 2	$\tilde{q}$	29	34	34	34	58	202	
	Variance	56.62	57.32	57.32	57.32	58.58	61.16 (84)	61.85 (78)
	SNR	53.35	53.97	53.97	53.97	60.95	60.32 (83)	61.37 (64)
	Skewness	38.15	36.68	36.68	36.68	59	64.85 (86)	65.34 (85)
	Kurtosis	40.79	47.56	47.56	47.56	57.53	60.46 (86)	60.88 (82)
	Entropy	55.58	57.88	57.88	57.88	56.69	63.6 (85)	63.6 (85)

(*Continued*)

Table 23.3 (*Continued*)

		n_{VD}	Shannon coding	Huffman coding	Hamming coding	$2n_{\mathrm{VD}}$	Total	Optimal
	ID	39.61	46.93	46.93	46.93	50.49	64.3 (90)	64.3 (90)
	Negentropy	33.68	36.19	36.19	36.19	46.65	62.55 (87)	63.39 (74)
Class 3	$\tilde{q}$	29	34	34	34	58	202	
	Variance	40.05	43.41	43.41	43.41	47.6	48.44 (84)	49.04 (70)
	SNR	40.53	41.85	41.85	41.85	46.04	47 (83)	47.36 (80)
	Skewness	26.62	29.02	29.02	29.02	41.13	48.2 (86)	48.2 (86)
	Kurtosis	23.62	27.22	27.22	27.22	36.57	45.8 (86)	45.8 (83)
	Entropy	37.05	38.73	38.73	38.73	42.45	46.04 (85)	46.76 (75)
	ID	33.09	36.69	36.69	36.69	37.89	47.12 (90)	47.12 (90)
	Negentropy	33.45	35.49	35.49	35.49	42.33	46.64 (87)	46.64 (87)
Class 4	$\tilde{q}$	29	36	36	34	58	202	
	Variance	69.66	73.08	73.08	73.08	78.21	74.36 (84)	80.34 (45)
	SNR	73.08	73.5	73.5	72.22	76.92	76.5 (83)	77.78 (62)
	Skewness	61.97	58.12	58.12	58.55	66.67	76.92 (86)	77.78 (76)
	Kurtosis	54.27	64.96	64.96	64.53	70.51	81.2 (86)	82.05 (81)
	Entropy	68.38	70.94	70.94	72.22	76.5	82.48 (85)	82.91 (78)
	ID	45.73	67.52	67.52	67.52	68.8	75.64 (90)	75.64 (90)
	Negentropy	61.54	62.39	62.39	64.53	67.09	76.92 (87)	76.92 (87)
Class 5	$\tilde{q}$	29	32	32	34	58	202	
	Variance	65.19	65.39	65.39	65.39	65.59	65.59 (84)	65.59 (55)
	SNR	58.95	61.77	61.77	60.56	65.59	65.59(83)	65.59 (47)
	Skewness	50.5	50.7	50.7	49.5	53.52	65.79 (86)	65.79 (85)
	Kurtosis	59.76	61.77	61.77	62.37	62.58	63.58 (86)	63.78 (79)
	Entropy	61.77	62.37	62.37	62.17	61.97	62.17 (85)	62.78 (38)
	ID	34.81	53.52	53.52	54.33	62.37	63.78 (90)	63.78 (85)
	Negentropy	49.09	50.91	50.91	49.7	55.33	63.78 (87)	63.78 (83)
Class 6	$\tilde{q}$	29	34	34	34	58	202	
	Variance	85.68	85.81	85.81	85.81	86.08	85.81 (84)	86.75 (40)
	SNR	84.34	84.07	84.07	84.07	85.81	86.48 (83)	87.15 (66)
	Skewness	73.63	74.03	74.03	74.03	76.31	86.08 (86)	86.08 (86)
	Kurtosis	85.54	85.01	85.01	85.01	86.48	86.61 (86)	87.55 (79)
	Entropy	87.55	87.01	87.01	87.01	87.01	86.61 (85)	89.29 (19)
	ID	47.79	63.45	63.45	63.45	86.88	85.54 (90)	88.09 (67)
	Negentropy	58.37	61.45	61.45	61.45	72.29	86.61 (87)	88.09 (72)
Class 7	$\tilde{q}$	29	37	36	34	58	202	
	Variance	80.77	76.92	76.92	80.77	80.77	73.08 (84)	84.62 (53)
	SNR	88.46	84.62	88.46	88.46	88.46	80.77 (83)	92.31 (26)
	Skewness	65.38	69.23	73.08	65.38	65.38	73.08 (86)	76.92 (81)
	Kurtosis	69.23	88.46	88.46	84.62	88.46	84.62 (86)	96.15 (69)
	Entropy	96.15	100	100	96.15	96.15	88.46 (85)	100 (17)
	ID	61.54	73.08	73.08	65.38	84.62	80.77 (90)	88.46 (48)
	Negentropy	80.77	69.23	69.23	65.38	65.38	76.92 (87)	84.62 (24)
Class 8	$\tilde{q}$	29	36	36	34	58	202	
	Variance	86.71	86.3	86.3	87.12	92.64	93.05 (84)	94.27 (76)
	SNR	86.09	85.89	85.89	85.89	91.82	93.66 (83)	93.66 (82)

	Skewness	70.35	73.82	73.82	72.39	82.62	92.23 (86)	92.43 (85)
	Kurtosis	80.98	81.6	81.6	80.16	87.53	93.46 (86)	93.46 (86)
	Entropy	86.5	86.71	86.71	86.71	90.8	94.68 (85)	94.68 (85)
	ID	68.3	79.75	79.75	79.75	87.12	93.05 (90)	93.46 (88)
	Negentropy	74.85	75.46	75.46	74.44	82	92.23 (87)	92.43 (80)
Class 9	$\tilde{q}$	29	36	35	34	58	202	
	Variance	55	60	60	55	55	55 (84)	65 (17)
	SNR	50	50	50	50	55	55 (83)	75 (17)
	Skewness	85	80	80	80	55	55 (86)	85 (29)
	Kurtosis	70	60	60	60	70	60 (86)	75 (24)
	Entropy	75	70	75	75	60	50 (85)	75 (26)
	ID	50	50	50	50	70	55 (90)	70 (57)
	Negentropy	50	50	50	50	50	55 (87)	60 (32)
Class 10	$\tilde{q}$	29	34	33	34	58	202	
	Variance	77.48	77.58	78.2	77.58	79.44	80.89 (84)	81.61 (79)
	SNR	76.96	77.38	77.48	77.38	80.06	81.3 (83)	81.92 (79)
	Skewness	48.66	49.59	48.66	49.59	72.83	81.61 (86)	81.92 (84)
	Kurtosis	56.51	58.26	54.65	58.26	63.84	83.26 (86)	83.26 (86)
	Entropy	67.77	68.7	69.21	68.7	75.1	81.92 (85)	81.92 (85)
	ID	65.08	72.42	73.04	72.42	79.86	83.57 (90)	83.57 (90)
	Negentropy	75.52	74.17	74.38	74.17	76.34	83.57 (87)	83.57 (87)
Class 11	$\tilde{q}$	29	34	34	34	58	202	
	Variance	35.49	36.18	36.18	36.18	43.56	46.39 (84)	46.52 (82)
	SNR	37.16	39.47	39.47	39.47	44.69	46.88 (83)	47.04 (80)
	Skewness	42.1	42.06	42.06	42.06	41.25	45.87 (86)	45.87 (86)
	Kurtosis	40.15	39.55	39.55	39.55	46.23	45.1 (86)	47.2 (67)
	Entropy	43.52	44.57	44.57	44.57	46.64	46.84 (85)	48.3 (66)
	ID	32.17	36.18	36.18	36.18	43.88	46.27 (90)	46.43 (89)
	Negentropy	37.8	37.16	37.16	37.16	42.22	47.16 (87)	47.33 (78)
Class 12	$\tilde{q}$	29	34	34	34	58	202	
	Variance	62.21	64.82	64.82	64.82	68.89	68.57 (84)	69.71 (78)
	SNR	60.1	60.59	60.59	60.59	66.78	68.73 (83)	68.89 (77)
	Skewness	43.81	42.67	42.67	42.67	63.03	67.26 (86)	67.43 (82)
	Kurtosis	41.53	47.39	47.39	47.39	60.26	61.56 (86)	63.84 (74)
	Entropy	58.79	61.73	61.73	61.73	63.84	65.15 (85)	65.64 (76)
	ID	44.3	48.37	48.37	48.37	50.49	61.4 (90)	61.4 (90)
	Negentropy	45.6	50	50	50	54.72	62.21 (87)	63.03 (82)
Class 13	$\tilde{q}$	29	35	34	34	58	202	
	Variance	99.06	99.53	99.06	99.06	99.53	98.58 (84)	99.53 (19)
	SNR	99.06	99.06	99.53	99.53	99.53	99.06 (83)	100 (47)
	Skewness	98.11	99.06	98.58	98.58	98.11	99.53 (86)	100 (73)
	Kurtosis	94.81	98.58	98.58	98.58	99.06	99.53 (86)	99.53 (76)
	Entropy	97.64	99.53	99.53	99.53	99.53	99.06 (85)	99.53 (33)
	ID	62.26	95.75	96.7	96.7	96.7	99.53 (90)	100 (75)
	Negentropy	92.92	91.98	92.45	92.45	94.34	100 (87)	100 (76)
Class 14	$\tilde{q}$	29	32	31	34	58	202	
	Variance	82.84	83.54	82.69	82.69	85.86	86.63(84)	86.63 (83)
	SNR	80.76	81.68	81.07	81.07	85.01	84.7 (83)	86.01 (61)
	Skewness	66.92	69.63	69.24	69.24	73.88	83.54 (86)	84.54 (75)

(*Continued*)

Table 23.3 (*Continued*)

		n_{VD}	Shannon coding	Huffman coding	Hamming coding	$2n_{\mathrm{VD}}$	Total	Optimal
	Kurtosis	75.35	79.06	79.13	79.13	81.3	87.17 (86)	87.17 (86)
	Entropy	80.06	82.23	82.3	82.3	84.39	87.4 (85)	87.4 (84)
	ID	59.12	56.11	56.72	56.72	79.98	85.24 (90)	85.24 (89)
	Negentropy	59.51	63.29	62.13	62.13	68.24	84.78 (87)	84.78 (87)
Class 15	$\tilde{q}$	29	34	33	34	58	202	
	Variance	62.11	61.58	61.84	61.58	65.26	69.47 (84)	70.53 (80)
	SNR	52.89	53.42	53.95	53.42	62.37	72.11 (83)	73.42 (80)
	Skewness	58.42	58.95	57.37	58.95	60.26	73.16 (86)	73.16 (86)
	Kurtosis	45.26	52.37	52.11	52.37	64.74	66.05 (86)	67.89 (67)
	Entropy	58.16	60	60	60	64.74	73.16 (85)	73.16 (79)
	ID	31.84	47.89	47.89	47.89	55	67.11 (90)	68.42 (87)
	Negentropy	38.68	40	39.47	40	45.53	66.84 (87)	66.84 (84)
Class 16	$\tilde{q}$	29	32	31	34	58	202	
	Variance	86.32	87.37	87.37	87.37	88.42	88.42 (84)	88.42 (35)
	SNR	88.42	88.42	88.42	88.42	88.42	88.42 (83)	91.58 (19)
	Skewness	86.32	87.37	87.37	87.37	88.42	88.42 (86)	89.47 (19)
	Kurtosis	87.37	87.37	87.37	87.37	87.37	88.42 (86)	88.42 (28)
	Entropy	92.63	88.42	88.42	88.42	88.42	88.42 (85)	92.63 (17)
	ID	89.47	88.42	88.42	88.42	88.42	89.47 (90)	89.47 (19)
	Negentropy	88.42	89.47	88.42	89.47	88.42	88.42 (87)	89.47 (34)

Table 23.4 First 35 highest prioritized BP/BD bands and corresponding removed bands after ID-BD with $\varepsilon = 0.1$ using 7 BP criteria for BP of HYDICE data

	BR/BD bands	Corresponding removed bands
Variance	60/56/78/53/52/55/54/49/45/92/38/34/28/26/27/25/24/23/22/20/21/102/19/18/17/16/15/14/13/12/11/10/84/9/8	61/67/66/65/59/57/68/62/64/77/76/79/63/80/58/75/81/69/50/82/48/51/46/70/47/44/43/42/41/74/93/91/40/95/39/90/94/37/96/89/83/36/35/33/88/32/71/31/30/97/29/87/73/103/108/105/104/109/101/106/107/111/110/112/113/100/114/72/115/99/86/116
Skewness	1/122/50/127/2/169/168/128/129/45/167/22/23/21/18/20/24/19/17/16/15/10/3/8/25/14/13/5/11/12/132/9/7/4/133	123/124/125/126/49/48/47/51/46/130/166/44/131/43/165/42/164/41
Kurtosis	1/59/69/74/83/56/85/86/93/55/54/122/109/2/120/3/127/130/128/132/4/133/134/144/145/143/135/141/140/146/138/139/136/53/137	60/61/62/64/63/65/66/58/67/68/57/70/80/82/81/72/79/78/77/76/75/71/84/73/92/95/91/94/90/88/89/87/96/97/108/113/112/110/111/123/114/107/106/115/105/116/104/117/103/118/121/119/124/102/101/126/131/100/129/125/99/98/142/152/153/154/147/151
Entropy	46/24/124/21/40/18/26/23/22/25/16/17/27/20/19/165/15/167/9/168/12/36/169/98/35/14/29/28/53/6/10/13/156/7/2	47/49/48/51/45/44/43/50/41/42/52/125/39/38/123/126/122/37/129/166/164/163/162/130/160/161/159/131/32/99/33/30/31/158/34/157/100/142/154/155

ID	59/81/56/85/94/86/71/84/55/4/145/120/109/135/146/139/138/133/141/136/143/132/137/134/140/54/150/128/8/1/127/11/3/2/13	60/63/62/61/65/58/57/64/66/80/79/67/82/70/83/74/68/72/77/76/69/78/91/92/93/75/88/95/90/89/96/87/73/97/113/115/112/114/111/116/110/118/108/106/117/107/105/104/119/121/103/144/151/148/102/149/152/147/155/142/101/154/153/100/157
ICA	1/81/93/85/68/84/86/56/55/111/54/120/53/33/149/146/145/28/143/37/141/139/140/137/138/27/52/136/26/135/25/44/24/19/14	82/80/79/78/92/95/77/94/91/76/96/90/75/89/88/83/74/67/70/66/87/69/65/64/61/60/59/62/63/72/57/97/58/73/71/112/110/113/114/109/115/108/116/107/106/117/105/118/104/119/121/103/102/34/32/101/31/150/148/30/36/147/151/29/35/152/100/153/142/144/154/155/38/39/156/99/40/157/158/159/41/42/160/43/161/162/45
SNR	78/93/102/62/56/49/44/53/55/52/54/38/120/137/138/136/139/134/140/133/141/85/135/132/86/143/131/128/145/32/84/144/146/122/28	80/91/92/95/89/94/90/88/96/79/82/105/107/108/104/109/101/110/103/63/106/77/61/111/70/112/100/114/113/75/99/59/116/57/83/60/117/118/46/47/66/69/64/45/115/50/119/97/51/67/87/43/42/58/71/41/72/98/39/37/65/73/142/31/33/147/130/129/30/29

Table 23.5 FCLS-unmixed abundance fractions of the 19 panel pixels in the HYDICE data using various values of $\tilde{q}$ determined by VD, DDA, and optimal FCLS performanc

		n_{VD}	Shannon coding	Huffman coding	Hamming coding	$2n_{\text{VD}}$	Total	Optimal
$\tilde{q}$ = Number of selected bands		9	15	14	13	18		
p_{11}	Variance	0.44	0.63	0.6	0.54	0.73	0.78 (65)	0.78 (32)
	SNR	0.41	0.52	0.52	0.52	0.52	0.77 (66)	0.77 (58)
	Skewness	0.27	0.84	0.84	0.82	0.85	0.81 (70)	0.92 (36)
	Kurtosis	0	0	0	0	0	0.7 (67)	0.7 (67)
	Entropy	0.83	0.87	0.85	0.84	0.88	0.77 (68)	1 (24)
	ID	0	0	0	0	0	0.76 (68)	0.76 (68)
	Negentropy	0	0	0	0	0.07	0.71 (67)	0.71 (63)
p_{12}	Variance	0.66	0.46	0.48	0.51	0.43	0.56 (65)	0.68 (11)
	SNR	0.57	0.59	0.59	0.59	0.59	0.56 (66)	0.59 (12)
	Skewness	0.54	0.53	0.36	0.3	0.52	0.56 (70)	0.68 (38)
	Kurtosis	0.38	0.47	0.46	0.39	0.49	0.54 (67)	0.6 (60)
	Entropy	0.93	0.9	0.86	0.86	0.72	0.53 (68)	0.93 (10)
	ID	0.47	0.34	0.33	0.33	0.36	0.55 (68)	0.61 (62)
	Negentropy	0.39	0.49	0.49	0.62	0.48	0.63 (67)	0.65 (11)
p_{13}	Variance	0	0	0	0	0	0 (65)	0 (10)
	SNR	0	0.05	0.05	0.05	0.05	0.01 (66)	0.07 (29)
	Skewness	0	0	0	0	0	0.05 (70)	0.23 (34)
	Kurtosis	0	0	0	0	0.01	0.02 (67)	0.18 (34)
	Entropy	0.68	0.69	0.67	0.69	0.67	0.01 (68)	0.83 (12)
	ID	0	0	0	0	0	0.02 (68)	0.2 (34)
	Negentropy	0.17	0	0	0.25	0	0.07 (67)	0.25 (13)

(*Continued*)

Table 23.5 (*Continued*)

		n_{VD}	Shannon coding	Huffman coding	Hamming coding	$2n_{\mathrm{VD}}$	Total	Optimal
$\tilde{q}$ = Number of selected bands		9	15	14	13	18		
p_{211}	Variance	0.85	0.87	0.87	0.87	0.87	0.89 (65)	0.89 (65)
	SNR	0.86	0.86	0.86	0.86	0.86	0.88 (66)	0.88 (66)
	Skewness	0.84	0.95	0.96	0.95	0.95	0.89 (70)	0.99 (36)
	Kurtosis	0.87	0.82	0.82	0.82	0.82	0.89 (67)	0.89 (67)
	Entropy	1.2	0.88	0.87	0.87	0.88	0.92 (68)	1.2 (9)
	ID	0.83	0.85	0.85	0.85	0.85	0.88 (68)	0.88 (68)
	Negentropy	0.86	0.84	0.84	0.83	0.84	0.89 (67)	0.89 (65)
p_{221}	Variance	0.53	0.71	0.7	0.7	0.72	0.75 (65)	0.75 (65)
	SNR	0.47	0.54	0.54	0.54	0.55	0.74 (66)	0.74 (66)
	Skewness	0.72	0.95	0.96	0.96	0.96	0.77 (70)	1 (37)
	Kurtosis	0.33	0.41	0.41	0.34	0.42	0.75 (67)	0.75 (67)
	Entropy	0	0.21	0.19	0.25	0.22	0.81 (68)	1 (38)
	ID	0.26	0.3	0.3	0.3	0.3	0.75 (68)	0.75 (68)
	Negentropy	0.33	0.51	0.51	0.39	0.54	0.76 (67)	0.76 (67)
p_{22}	Variance	0.75	0.86	0.86	0.85	0.81	0.78 (65)	0.86 (14)
	SNR	0.88	0.87	0.87	0.87	0.88	0.83 (66)	0.88 (29)
	Skewness	0.87	0.74	0.75	0.8	0.7	0.79 (70)	0.87 (9)
	Kurtosis	0.81	0.81	0.81	0.81	0.82	0.78 (67)	0.85 (42)
	Entropy	0.64	0.6	0.62	0.61	0.6	0.77 (68)	0.78 (67)
	ID	0.77	0.73	0.73	0.73	0.74	0.8 (68)	0.81 (37)
	Negentropy	0.73	0.78	0.78	0.78	0.82	0.82 (67)	0.86 (32)
p_{23}	Variance	0.46	0.49	0.49	0.49	0.48	0.46 (65)	0.49 (14)
	SNR	0.42	0.49	0.49	0.49	0.49	0.45 (66)	0.49 (12)
	Skewness	0.38	0.24	0.24	0.24	0.22	0.45 (70)	0.45 (70)
	Kurtosis	0.47	0.38	0.38	0.4	0.4	0.46 (67)	0.47 (38)
	Entropy	0	0.23	0.19	0.18	0.19	0.44 (68)	0.44 (68)
	ID	0.25	0.37	0.37	0.37	0.37	0.47 (68)	0.47 (39)
	Negentropy	0.18	0.43	0.43	0.4	0.44	0.45 (67)	0.46 (32)
$\tilde{q}$ = Number of selected bands		9	14	13	13	18		
p_{311}	Variance	0.98	0.97	0.97	0.97	0.96	0.95 (65)	0.98 (9)
	SNR	0.97	0.98	0.98	0.98	0.98	0.95 (66)	0.99 (11)
	Skewness	0.56	0.89	0.89	0.89	0.89	0.95 (70)	0.95 (69)
	Kurtosis	1	1	1	1	1	0.95 (67)	1 (13)
	Entropy	0.72	0.74	0.74	0.74	0.75	0.94 (68)	0.94 (68)
	ID	1	1	1	1	1	0.95 (68)	1 (11)
	Negentropy	1	1	1	1	1	0.95 (67)	1 (12)
p_{312}	Variance	1	1	1	1	1	1 (65)	1 (9)
	SNR	1	1	1	1	1	1	1 (9)
	Skewness	1	0.96	0.97	0.97	0.96	1 (70)	1 (65)
	Kurtosis	0.92	1	1	1	1	1 (67)	1 (13)

	Entropy	0.78	0.82	0.83	0.83	0.81	1 (68)	1 (68)
	ID	1	1	1	1	1	1 (68)	1 (11)
	Negentropy	0.99	1	1	1	1	1 (67)	1 (12)
p_{32}	Variance	0.71	0.7	0.7	0.7	0.68	0.63 (65)	0.71 (10)
	SNR	0.7	0.72	0.72	0.72	0.72	0.61 (66)	0.73 (11)
	Skewness	1	0.77	0.77	0.77	0.8	0.67 (70)	1 (9)
	Kurtosis	0.62	0.69	0.69	0.69	0.64	0.58 (67)	0.71 (40)
	Entropy	0	0	0	0	0	0.33 (68)	0.33 (68)
	ID	0.67	0.6	0.6	0.6	0.58	0.71 (68)	0.74 (67)
	Negentropy	0.47	0.71	0.72	0.72	0.7	0.53 (67)	0.72 (27)
p_{33}	Variance	0.33	0.32	0.32	0.32	0.32	0.32 (65)	0.33 (9)
	SNR	0.28	0.32	0.32	0.32	0.32	0.32 (66)	0.32 (11)
	Skewness	0.56	0.43	0.43	0.43	0.43	0.34 (70)	0.56 (9)
	Kurtosis	0.34	0.32	0.32	0.32	0.32	0.31 (67)	0.36 (10)
	Entropy	0	0	0	0	0	0.3 (68)	0.35 (61)
	ID	0.37	0.36	0.36	0.36	0.35	0.33 (68)	0.37 (9)
	Negentropy	0.42	0.29	0.29	0.29	0.29	0.31 (67)	0.42 (9)
$\tilde{q}$ = Number of selected bands		9	13	13	13	18		
p_{411}	Variance	0.52	0.55	0.55	0.55	0.67	0.71 (65)	0.72 (35)
	Skewness	0.61	0.85	0.85	0.85	0.86	0.8 (70)	0.88 (16)
	Kurtosis	0.4	0.44	0.44	0.44	0.45	0.75 (67)	0.75 (67)
	Entropy	0.74	0.74	0.74	0.74	0.77	0.77 (68)	0.87 (63)
	ID	0.43	0.43	0.43	0.43	0.44	0.75 (68)	0.75 (68)
	Negentropy	0.22	0.39	0.39	0.39	0.45	0.75 (67)	0.75 (56)
	SNR	0.45	0.53	0.53	0.53	0.53	0.72 (66)	0.72 (65)
p_{412}	Variance	0.79	0.83	0.83	0.83	0.66	0.79 (65)	0.83 (14)
	SNR	0.91	0.86	0.86	0.86	0.87	0.81 (66)	0.91 (9)
	Skewness	0.81	0	0	0	0.11	0.76 (70)	0.81 (9)
	Kurtosis	0.82	1	1	1	1	0.81 (67)	1 (11)
	Entropy	0.01	0.16	0.16	0.16	0.23	0.72 (68)	0.72 (68)
	ID	1	1	1	1	1	0.9 (68)	1 (10)
	Negentropy	1	0.98	0.98	0.98	0.96	0.86 (67)	1 (10)
p_{42}	Variance	0.81	0.69	0.69	0.69	0.56	0.75 (65)	0.81 (9)
	SNR	0.83	0.82	0.82	0.82	0.82	0.76 (66)	0.83 (9)
	Skewness	0.42	0.24	0.24	0.24	0.25	0.75 (70)	0.75 (70)
	Kurtosis	0.86	0.8	0.8	0.8	0.85	0.76 (67)	0.86 (9)
	Entropy	0.07	0.23	0.23	0.23	0.33	0.81 (68)	0.83 (65)
	ID	0.85	0.86	0.86	0.86	0.84	0.73 (68)	0.86 (16)
	Negentropy	0.86	0.65	0.65	0.65	0.63	0.76 (67)	0.86 (9)
p_{43}	Variance	0.11	0.12	0.12	0.12	0.16	0.17 (65)	0.17 (65)
	SNR	0.14	0.13	0.13	0.13	0.14	0.18 (66)	0.18 (66)
	Skewness	0.17	0.21	0.21	0.21	0.21	0.18 (70)	0.22 (24)
	Kurtosis	0.17	0.15	0.15	0.15	0.16	0.17 (67)	0.18 (65)
	Entropy	0.22	0.21	0.21	0.21	0.21	0.19 (68)	0.26 (54)
	ID	0.15	0.15	0.15	0.15	0.15	0.18 (68)	0.18 (60)
	Negentropy	0.17	0.06	0.06	0.06	0.17	0.19 (67)	0.19 (54)

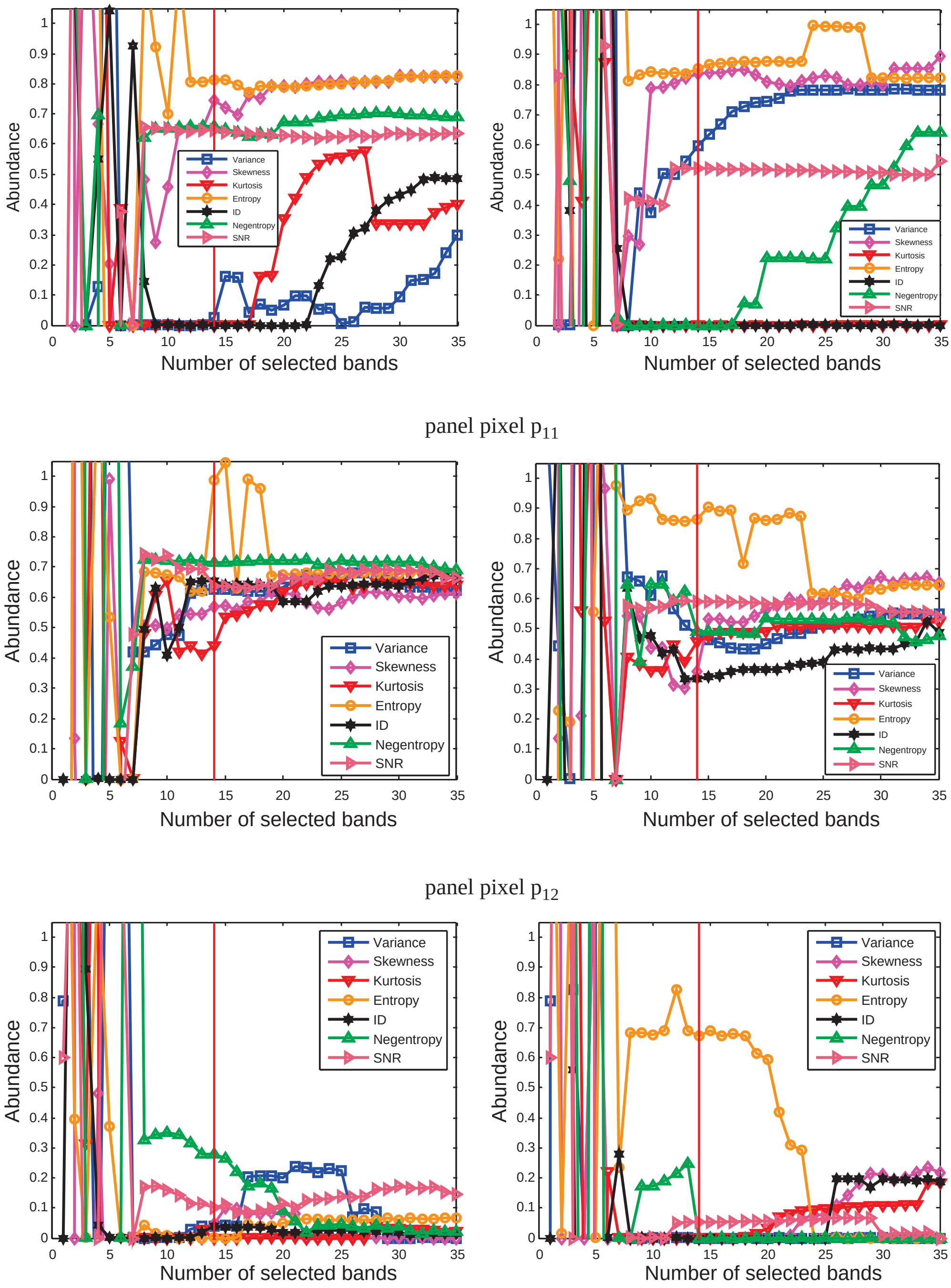
(Continued)

Table 23.5 (*Continued*)

		n_{VD}	Shannon coding	Huffman coding	Hamming coding	$2n_{\mathrm{VD}}$	Total	Optimal
$\tilde{q}$ = Number of selected bands		9	13	12	13	18		
p_{511}	Variance	0.82	0.83	0.8	0.83	0.84	0.83 (65)	0.84 (24)
	SNR	0.72	0.81	0.81	0.81	0.83	0.84 (66)	0.84 (47)
	Skewness	0.86	0.81	0.81	0.81	0.84	0.84 (70)	0.87 (17)
	Kurtosis	0.82	0.83	0.82	0.83	0.83	0.84 (67)	0.84 (67)
	Entropy	0.87	0.87	0.87	0.87	0.87	0.84 (68)	0.88 (25)
	ID	0.82	0.82	0.82	0.82	0.82	0.83 (68)	0.84 (31)
	Negentropy	0.82	0.82	0.81	0.82	0.83	0.84 (67)	0.84 (67)
p_{521}	Variance	1	1	1	1	1	1 (65)	1 (9)
	SNR	1	1	1	1	1	1 (66)	1 (9)
	Skewness	0.83	1	0.99	1	1	1 (70)	1 (13)
	Kurtosis	1	1	1	1	1	1 (67)	1 (13)
	Entropy	1	1	1	1	1	1 (68)	1 (17)
	ID	1	1	1	1	1	1 (68)	1 (10)
	Negentropy	1	1	1	1	1	1 (67)	1 (15)
p_{52}	Variance	0.92	0.92	0.93	0.92	0.92	0.93 (65)	0.93 (12)
	SNR	0.94	0.93	0.93	0.93	0.93	0.91 (66)	0.94 (9)
	Skewness	0.81	0.9	0.89	0.9	0.9	0.91 (70)	0.91 (64)
	Kurtosis	0.85	0.91	0.9	0.91	0.95	0.91 (67)	0.95 (16)
	Entropy	0.84	0.82	0.84	0.82	0.82	0.9 (68)	0.92 (67)
	ID	0.88	0.89	0.87	0.89	0.89	0.91 (68)	0.93 (35)
	Negentropy	0.92	0.94	0.93	0.94	0.93	0.91 (67)	0.94 (10)
p_{53}	Variance	0.11	0.11	0.11	0.11	0.14	0.1 (65)	0.14 (23)
	SNR	0.11	0.12	0.12	0.12	0.12	0.1 (66)	0.14 (45)
	Skewness	0.05	0	0	0	0	0.14 (70)	0.16 (63)
	Kurtosis	0.12	0.09	0.09	0.09	0.08	0.11 (67)	0.12 (35)
	Entropy	0.02	0.14	0.15	0.14	0.13	0.15 (68)	0.18 (63)
	ID	0.08	0.12	0.12	0.12	0.12	0.1 (68)	0.12 (31)
	Negentropy	0.1	0.12	0.09	0.12	0.11	0.11 (67)	0.15 (46)

Chapter 3. Figure 23.4(a)–(e) hows the areas under their 2D ROC curves of (P_D,P_F) resulting from using PBS with prioritized bands with/without ID-BD where figures on the right and on the left are obtained by PBS with/without ID-BD and DDA = 14 (Huffman coding) and vertical lines were also used to show the cutoff threshold determined by DDA = 14 calculated by the Huffman coding. Figure 23.4(f) shows the overall averaged panel pixel detection rates specified by the areas under 2D ROC curves of (P_D,P_F) obtained by averaging the areas under 2D ROC curves of (P_D,P_F) in Figure 23.4(a)–(e).

As we can see from Figure 23.4(f), the value 14 of DDA determined by the Huffman coding indeed provided a very good estimate for the value of $\tilde{q}$.

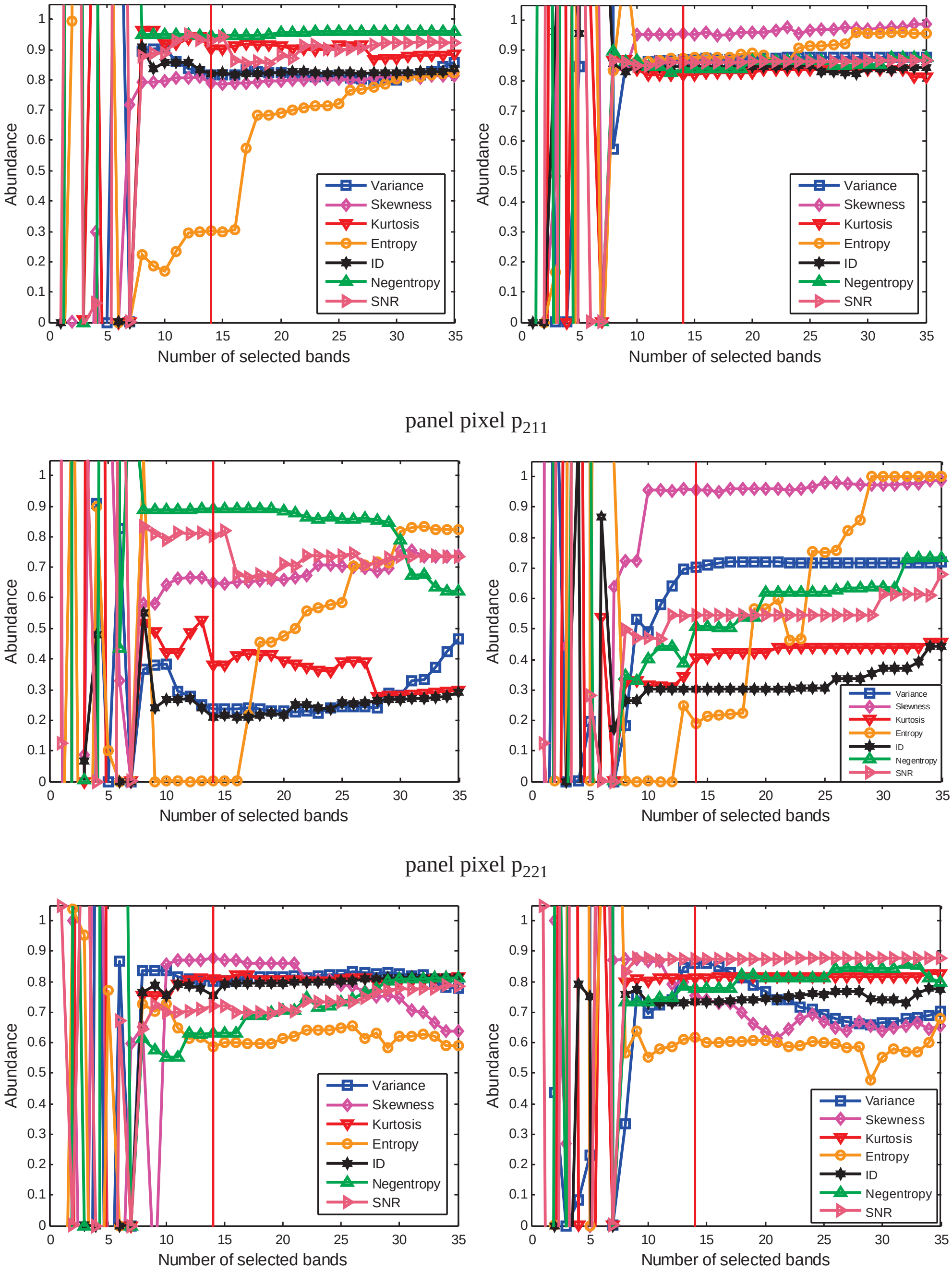


panel pixel p_{11}

panel pixel p_{12}

panel pixel p_{13}

Figure 23.3 FCLS-unmixed results of the 19 R pixels using PBS prioritized bands with figures on the right and on the left obtained by PBS with/without ID-BD and DDA = 14 (Huffman coding).

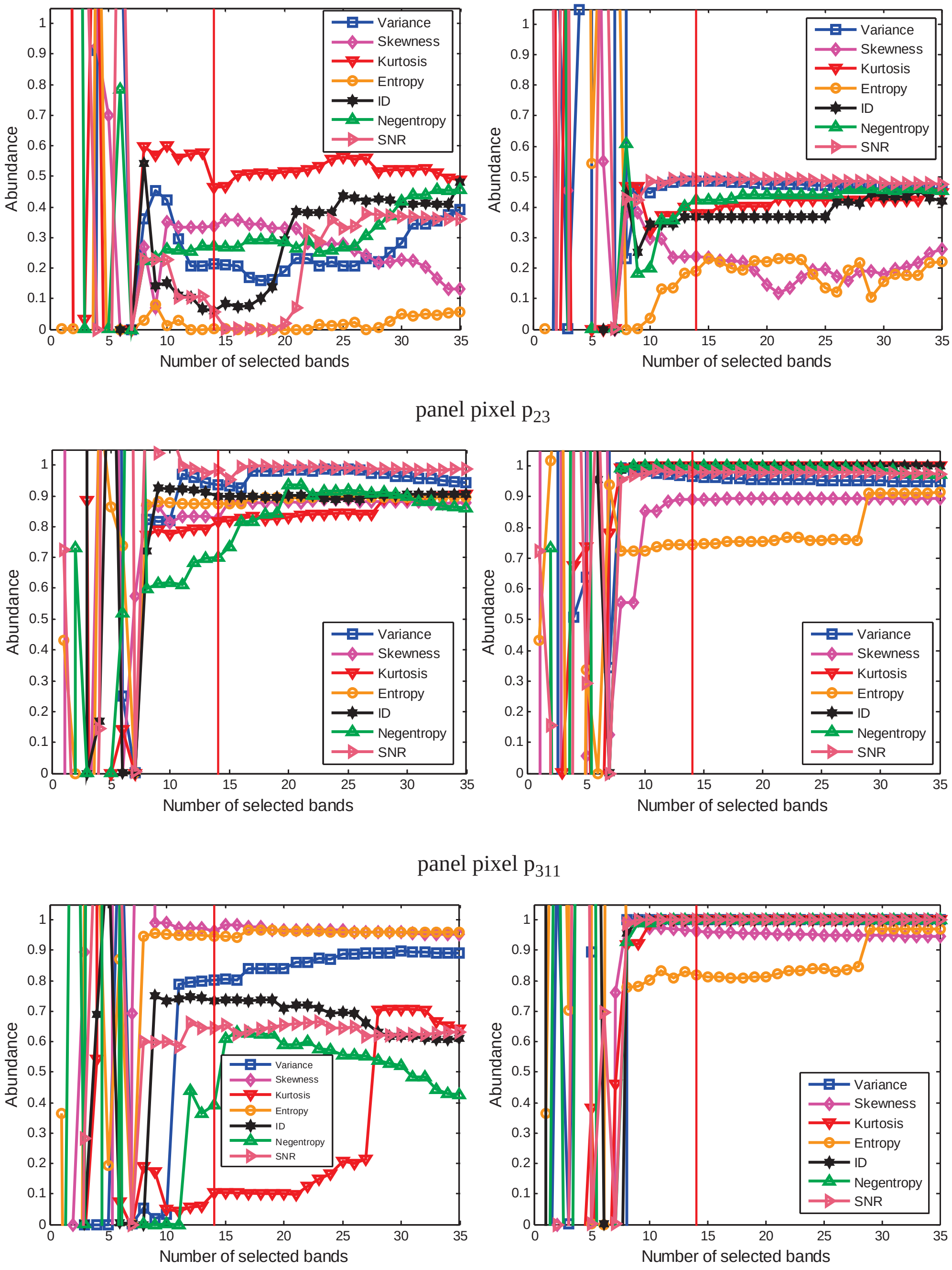


panel pixel p_{211}

panel pixel p_{221}

panel pixel p_{22}

Figure 23.3 *(Continued)*

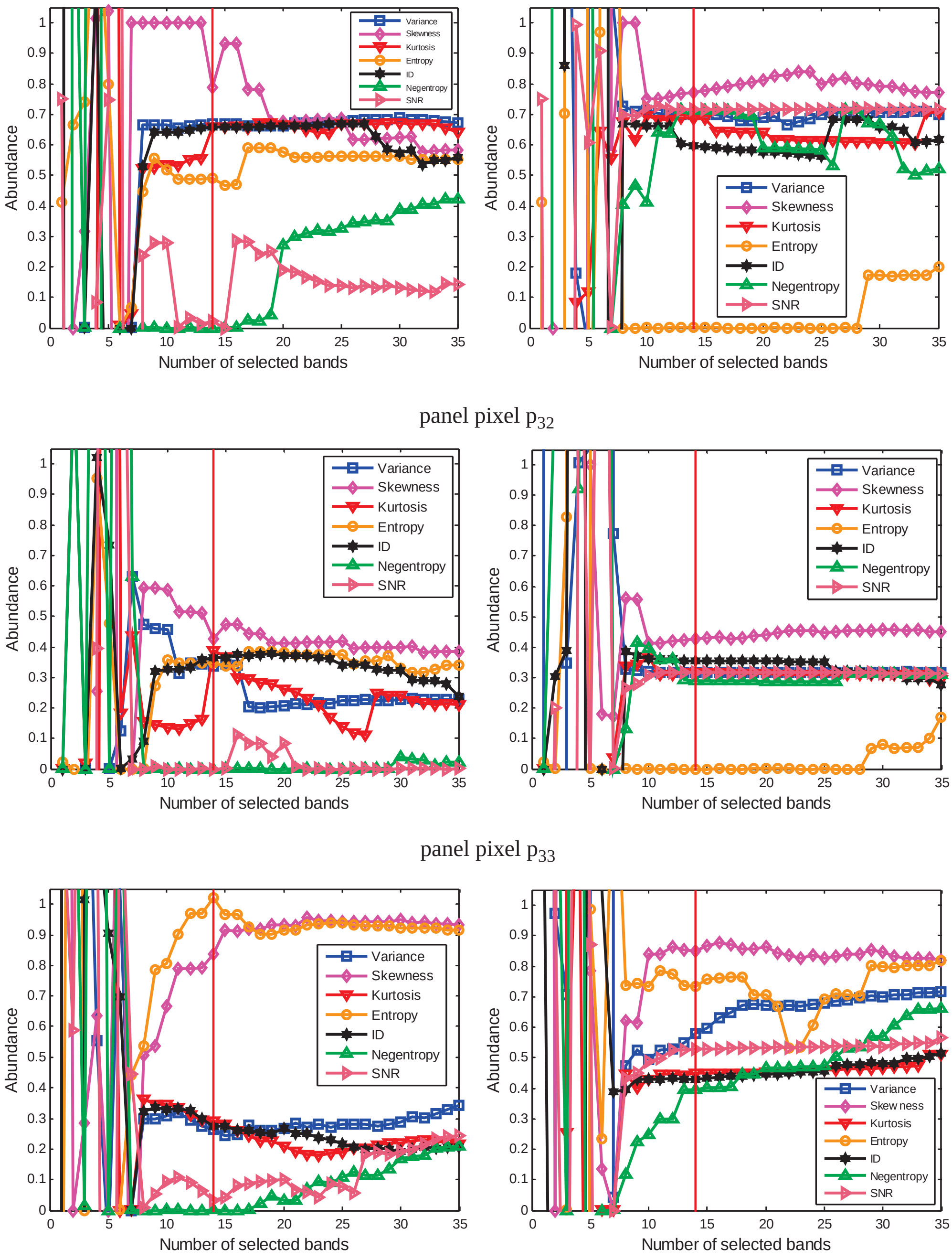


panel pixel p_{23}

panel pixel p_{311}

panel pixel p_{312}

Figure 23.3 (*Continued*)

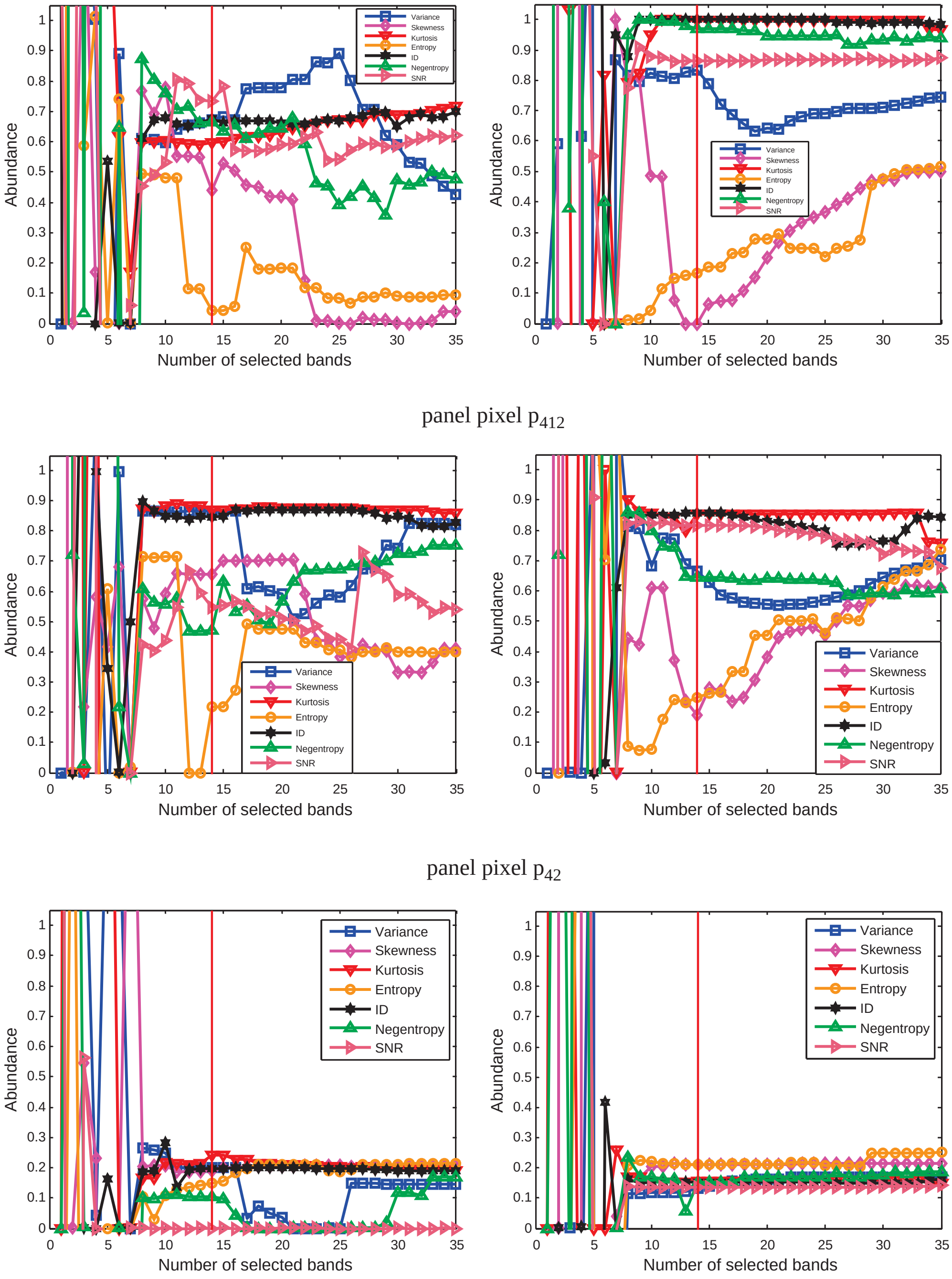


panel pixel p_{32}

panel pixel p_{33}

panel pixel p_{411}

Figure 23.3 (*Continued*)

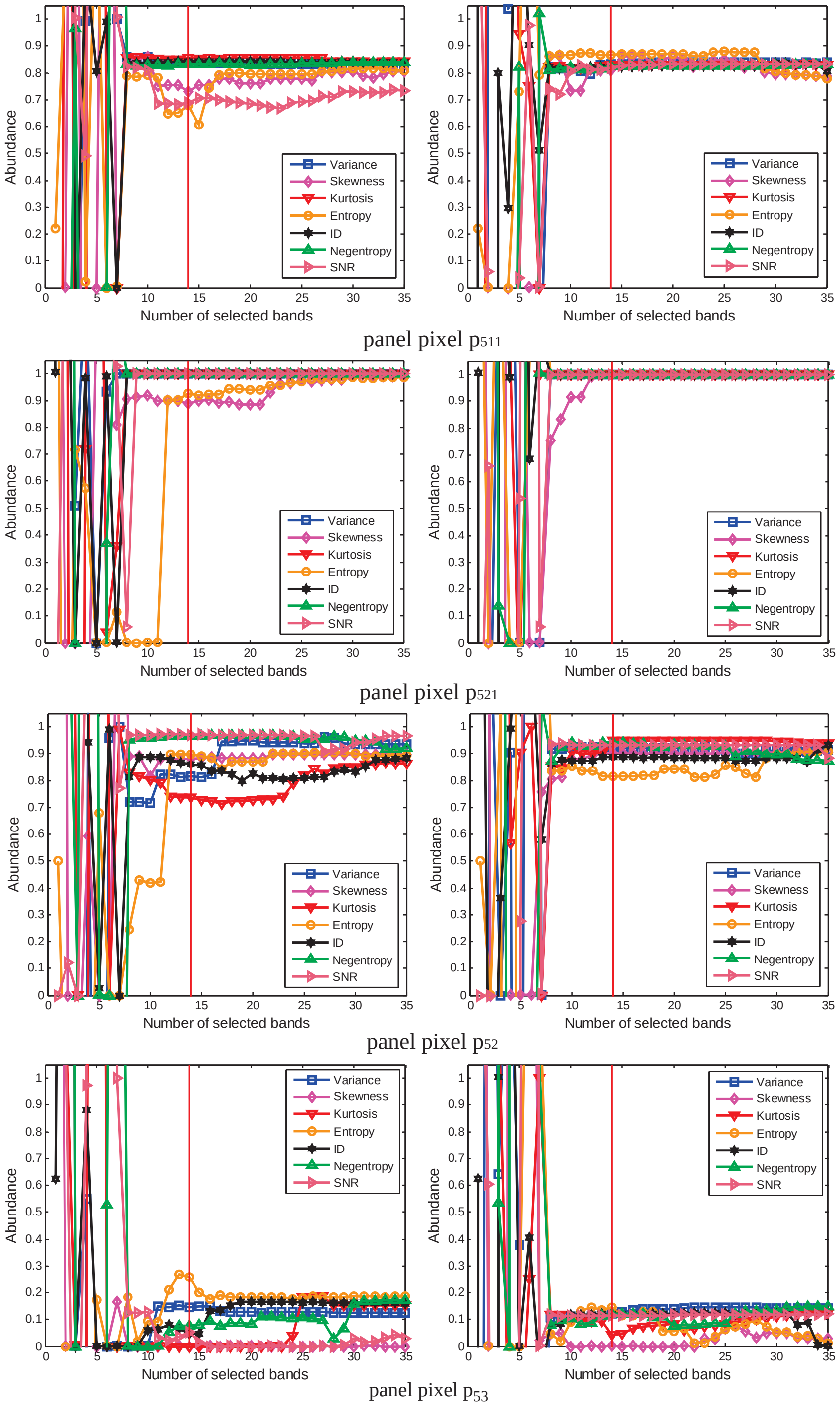


panel pixel p_{412}

panel pixel p_{42}

panel pixel p_{43}

Figure 23.3 (*Continued*)



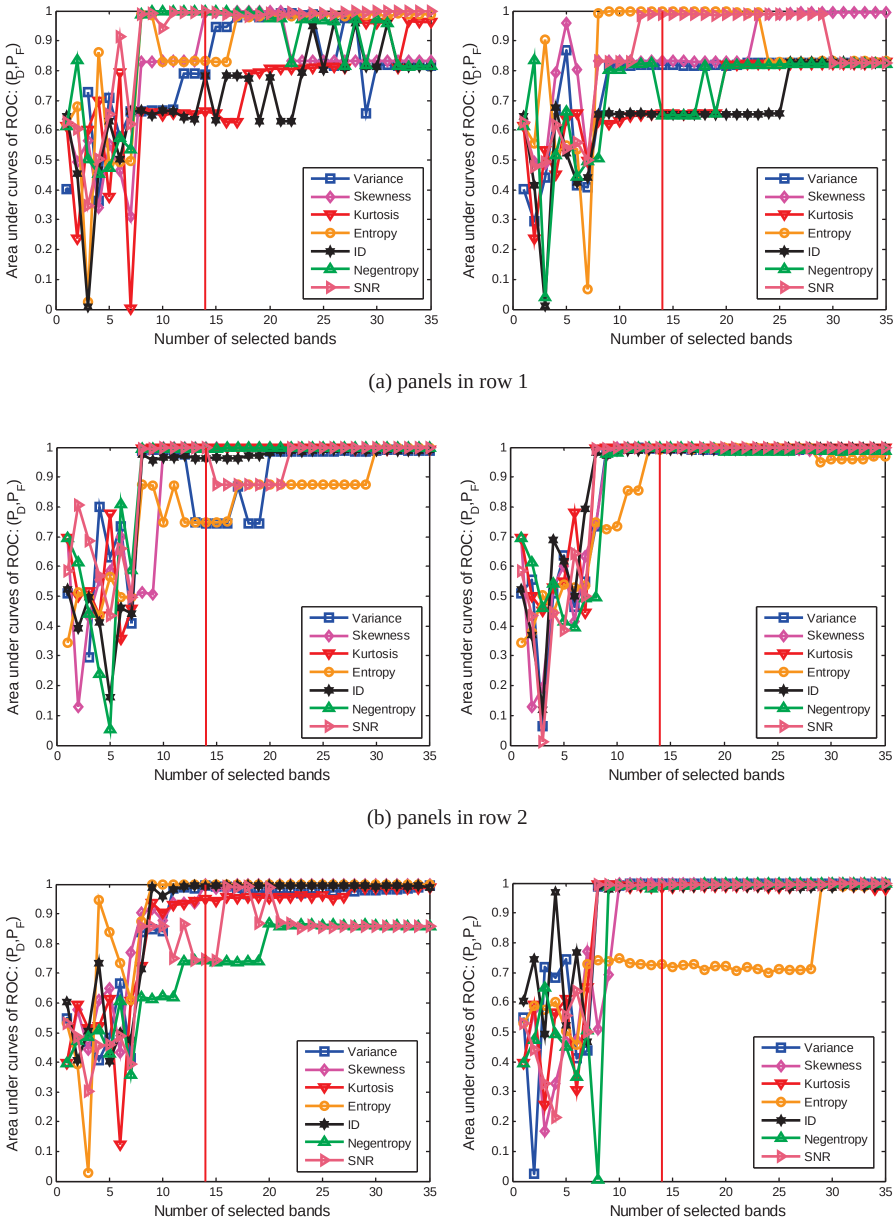
panel pixel p_{511}

panel pixel p_{521}

panel pixel p_{52}

panel pixel p_{53}

Figure 23.3 (*Continued*)

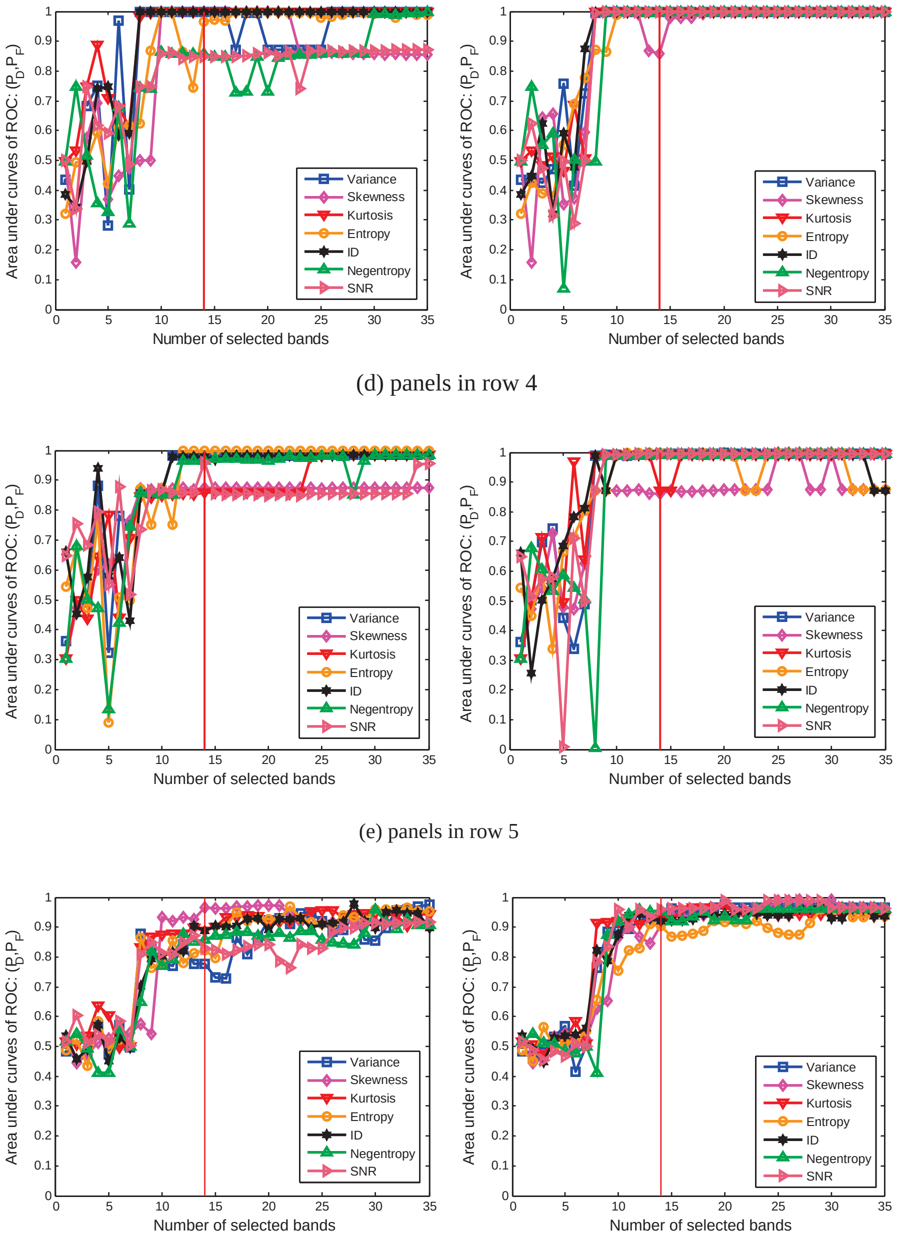


(a) panels in row 1

(b) panels in row 2

(c) panels in row 3

Figure 23.4 Areas under 2D ROC curves of (P_D, P_F) with figures on the right and on the left obtained by PBS with/without ID-BD and DDA = 14 (Huffman coding).



(d) panels in row 4

(e) panels in row 5

(f) five rows panels

Figure 23.4 (*Continued*)

23.8 Conclusions

Progressive band selection (PBS) is a new theory developed for band selection (BS), which is particularly useful in data communication and transmission. It can be considered as variable dimensionality BS (VDBS) as opposed to FDBS that implements BS dynamically by making band dimensionality adapt to various applications. It takes advantage of progressive band dimensionality process (PBDP), developed in Chapter 21, via band prioritization (BP) and band de-correlation (BD) to fine-tune BS in a progressive manner so that the band previously selected can be either expanded or removed to accommodate practical constraints. It offers several benefits over the traditional fixed band dimensionality BS. First, PBS does not need to have the precise knowledge of the number of bands needed to be selected because it makes the number of selected bands, *p*, vary and selects bands progressively in a forward and backward manner without repeatedly implementing BS. Second, PBS can be implemented by tuning variable bands according to its applications, such as selecting different ranges of wavelengths with different bands. Third, PBS keeps track of previously selected bands to update results in band expansion and reduction to accomplish various tasks, such as data compression, communication, storage, archiving and transmission, and so on. Finally, PBS can be implemented in causality and real time, a new emerging area, called progressive band processing in hyperspectral imaging, which will be one of the main themes in Chang (2013).

VI

Hyperspectral Signal Coding

The second category of this book, Category B, is devoted to hyperspectral signal processing. The question is for a given data sample vector treated as a signature vector without reference to others, what is the best possible we can do to explore as much spectral information across the entire wavelength range to specify the data sample vector for spectral characterization. There are two ways to process hyperspectral signals either in a discrete manner as discrete signal processing referred to as signal coding in Part VI or in a continuous manner as continuous signal processing referred to as signal characterization in Part VII.

In Part VI, the focus is placed on hyperspectral signal coding that represents a signature vector by a discrete vector as a code word that can be considered as its fingerprint. A simplest way to accomplish this task is binary coding that binarizes each component of a signature vector for its identification. An earliest attempt was made by Mazer et al. (1988) to develop a so-called spectral analysis manager (SPAM) that encoded a remote sensing image data sample vector into a binary code vector for signature discrimination, classification, and identification. It calculates spectral mean and interband spectral difference and uses them as thresholds to generate a binary code word for a given hyperspectral signature vector. The SPAM binary coding was then extended to a spectral feature-based binary coding (SFBC) developed by Qian et al. (1996) who incorporated an additional binary code word produced by thresholding the spectral mean deviation to improve SPAM. Chapter 24 studies these two coding schemes and follows similar ideas to further design several new binary signature coding schemes, median partition (MP), halfway partition (HP), and equal probability partition (EPP). Since the coding schemes in Chapter 24 perform component-wise encoding in the sense that the spectral value at each wavelength is encoded separately and independently without taking advantage of spectral correlation among adjacent bands, they are memoryless coding and can be viewed as scalar coding in context of the source coding. Therefore, an obvious extension to scalar coding is vector coding. Chapter 25 investigates such an extension where two vector coding approaches to encoding hyperspectral signals are developed, called spectral derivative feature coding (SDFC), and spectral feature probabilistic coding (SFPC) where SDFC is derived by including spectral textures as features to account for spectral correlation

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

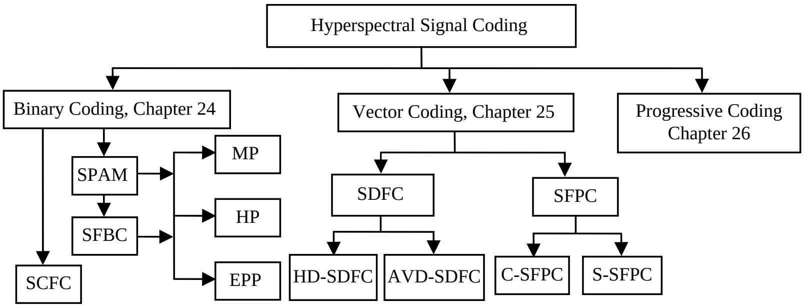

Hyperspectral Signal Coding
Binary Coding, Chapter 24
Vector Coding, Chapter 25
Progressive Coding Chapter 26
SPAM
SFBC
SCFC
MP
HP
EPP
SDFC
SFPC
HD-SDFC
AVD-SDFC
C-SFPC
S-SFPC

Figure VI.1 Topics of three chapters in Part VI.

among two successive adjacent bands and SFPC has its origin derived from arithmetic coding, which is a well-established coding method developed by Rissanen (1976) and Langdon and Rissanen (1981) and has been widely used in image coding.

One of major applications in hyperspectral signal coding is satellite data communication where data compression and transmission are critical and crucial. So, a progressive signature coding (PSC) Chapter 26 develops that allows signal coding to be progressive. In this case, hyperspectral signals can be encoded in a coarse-to-fine resolution manner. As a result, data transmission can take place more rapidly through each pass where each pass increases signal resolution for better data processing. The idea of PSC is similar to that used by progressive band selection (PBS) discussed in Chapter 23 where the coding process is carried out progressively in the exactly same way as band selection performed by PBS in the sense that the results produced by a previous process are always part of the following subsequent process to improve results. In PBS band prioritization (BP) is used to prioritize spectral bands so that progressive band dimensionality process (PBDP) is performed in band reduction or expansion. Unfortunately, such BP is developed for band images and is not applicable to signature coding. Therefore, a new coding scheme called multistage pulse code modulation (MPCM) is designed in Chapter 26 to encode a hyperspectral signal wavelength by wavelength progressively.

Finally, Figure VI.1 shows the topics covered in three chapters in Part VI.

24

Binary Coding For Spectral Signatures

Binary coding is the simplest way to characterize spectral features. One commonly used method is a binary coding-based image software system, called spectral analysis manager (SPAM) for remotely sensed imagery developed by Mazer et al. (1988). It makes use of spectral mean and interband spectral difference as thresholds to generate a binary code word for a spectral signature vector. It is generally effective and also very simple to implement. The SPAM binary coding was further extended to a spectral feature binary coding (SFBC) developed by Qian et al. (1996) by incorporating an additional binary code word produced by thresholding the spectral mean deviation to further improve the SPAM performance. This chapter revisits these two approaches and further develops three new binary coding methods, median partition (MP), halfway partition (HP), and equal probability partition (EPP), all of which can be implemented in conjunction with the ideas of SPAM and SFBC to create new sets of binary code words for spectral signature coding. As a result, different combinations of various binary coding methods can be used for spectral signature coding and applications in spectral discrimination and identification. Finally, a new criterion called *a posteriori* discrimination probability (APDP) is also introduced as a performance measure to coompare two different binray coding methods for performance analysis.

24.1 Introduction

Binary coding is one of simplest coding methods to represent data with binary values, {0,1}. One such method is the bit plane coding commonly used in image compression where an image is represented in accordance with its bit significance in terms of gray-scale values (Gonzalez and Woods, 2002). The idea of applying binary coding to spectral data was first proposed by Mazer et al. (1988) who developed a binary coding-based image software system called SPAM for remotely sensed imagery. Since remotely sensed data samples are collected by a number of spectral channels simultaneously, a data sample is actually a column vector with its components made up of data samples acquired by separate spectral channels. More specifically, assume that L is the number of spectral channels used for data acquisition. Each data sample vector, referred to as signature vector, is represented by an L-dimensional vector with the lth signature component specified by the data sample in the lth spectral band. SPAM binary coding encodes an L-dimensional signature vector as a $(2L\text{-}2)$-dimensional binary code word composed of the first L binary values encoding the sign of the difference between a signature component sample and its spectral signature mean, and a new set of

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

additional L-2 binary values encoding the sign of the difference between a component pixel in a signature vector and its adjacent signature components within the signature vector to be encoded. SPAM binary coding was further extended to so-called spectral feature binary coding (SFBC) by Qian et al. (1996) who included another set of additional L binary values to encode a signature vector as a (3L-2)-dimensional binary code word. The newly included L binary values were used to indicate whether the deviation of a signature component from the spectral signature mean was greater than a threshold that was obtained by averaging the absolute differences between each signature component and the spectral signature mean. Both binary coding methods have demonstrated some success in spectral signature coding.

This chapter investigates these two coding methods and further develops three new methods for binary coding, referred to as median partition (MP) binary coding, halfway partition (HP) binary coding, and equal probability partition (EPP) binary coding. MP binary coding is modified from the SPAM binary coding by replacing the spectral signature mean with the spectral signature median. HP binary coding is derived from the bit plane coding where the spectral signature mean is replaced with the halfway distance between the maximal and minimal spectral values of the spectrum of a signature vector. EPP binary coding is originated from the Shannon–Fano coding (Lynch, 1985). It normalizes a signature vector as a probability vector in the same fashion that the spectral information divergence (SID) was derived (Chang, 2000; Chapter 2, Chang, 2003a). Then the spectral signature mean used by SPAM is replaced with the threshold that divides the probabilities of the spectral values of the signature vector into two partitions with best possible equal probabilities. These three new binary coding methods can be also implemented in conjunction with the ideas proposed by Mazer et al. and Qian et al. to create new sets of binary code words. As a result, various binary coding methods can be derived to yield new binary coding schemes. In this chapter, eight different binary coding methods resulting from different combinations will be investigated and studied for analysis, which include SPAM binary coding, Qian et al.'s SFBC. In order to measure the distance between two binary code words, Hamming spectral distance (HSD) introduced in SPAM and average Hamming spectral distance (AHSD) are used for performance evaulation. On the other hand, in order to compare two different binary coding methods in spectral discrimination and identification, a new criterion called *a posteriori* discrimination probability (APDP) is also introduced for performance comparison. Interestingly, as demonstrated in our experimental results, each of the eight studied binary coding methods has its own merit in different applications and none of them can claim to be the best. This is because each signature vector has its own spectral characteristics and every coding method has its own ability in capturing different spectral behaviors. It is natural that one coding method that may perform well on one signature vector may perform poorly on other signature vectors. Therefore, it is expected that each one of all the eight coding methods has its own strengths and weaknesses. As a consequence, no one coding method is uniformly superior to another.

24.2 Binary Coding

Assume that a spectral signature vector **s** specified by a column vectior $\mathbf{s} = (s_1, s_2, \ldots, s_L)^T$ where L is the total number of spectral bands. Each signature component s_l in **s** represents a spectral value at the lth spectral band that is acquired by a particular wavelength ω_l in a specific spectral range.

24.2.1 SPAM Binary Coding

Let μ be the mean of a spectral signature **s** and given by

$$\mu = (1/L)\sum_{l=1}^{L} s_l \tag{24.1}$$

Using (24.1) we can encode the signature vector $\mathbf{s}$ as a (2L-2)-dimensional binary code word, denoted by $\left(\mathbf{s}^a\,\mathbf{s}^b\right)$ where $\mathbf{s}^a = \left(s_1^a s_2^a \cdots s_L^a\right)$ is an L-dimensional binary code word and $\mathbf{s}^b = \left(s_2^b \cdots s_{L-1}^b\right)$ is an $(L-2)$-dimensional binary code word, both of which are defined by

$$s_l^a = \begin{cases} 1; & \text{if } s_l \geq \mu \\ 0; & \text{otherwise} \end{cases} \quad \text{for } l = 1, 2, \ldots, L \tag{24.2}$$

and

$$s_l^b = \begin{cases} 1; & \text{if } s_{l+1} \geq s_{l-1} \\ 0; & \text{otherwise} \end{cases} \quad \text{for } l = 2, \ldots, L-1 \tag{24.3}$$

Now, for given two spectral signatures vectors $\mathbf{s}_i = (s_{i1}, s_{i2}, \ldots, s_{iL})^T$ and $\mathbf{s}_j = \left(s_{j1}, s_{j2}, \ldots, s_{jL}\right)^T$ we can measure their similarity based on the spectral distance between these two signature vectors via their encoded binary code words by an appropriate distance metric. More precisely, let $\left(\mathbf{s}_i^a\,\mathbf{s}_i^b\right)$ and $\left(\mathbf{s}_j^a\,\mathbf{s}_j^b\right)$ be the code words of the signature vectors $\mathbf{s}_i$ and $\mathbf{s}_j$, respectively, with $\mathbf{s}_i^a$, $\mathbf{s}_j^a$ defined by (24.2) and $\mathbf{s}_i^b$, $\mathbf{s}_j^b$ defined by (24.3). The spectral distance between $\mathbf{s}_i$ and $\mathbf{s}_j$ is then defined in SPAM by the Hamming spectral distance (HSD) between their respective binary code words, HSD($\mathbf{s}_i$,$\mathbf{s}_j$) as follows:

$$\text{HSD}(\mathbf{s}_i, \mathbf{s}_j) = \sum\nolimits_{l=1}^{L} \left(s_{il}^a \oplus s_{jl}^a\right) + \sum\nolimits_{l=2}^{L-1} \left(s_{il}^b \oplus s_{jl}^b\right) \tag{24.4}$$

where $\oplus$ is the exclusive OR operation. Using (24.4) the two signature vectors $\mathbf{s}_i$ and $\mathbf{s}_j$ are identical if their Hamming spectral distance HSD($\mathbf{s}_i$,$\mathbf{s}_j$) = 0.

As an alternative to HSD, we can also define another distance measure between their respective binary cord words, called average Hamming spectral distance (AHSD)/per band, denoted by AHSD($\mathbf{s}_i$,$\mathbf{s}_j$) as follows:

$$\text{AHSD}(\mathbf{s}_1, \mathbf{s}_2) = (1/L) \sum\nolimits_{l=1}^{L} \left(s_{1l}^a \oplus s_{2l}^a\right) + (1/(L-2)) \sum\nolimits_{l=2}^{L-1} \left(s_{1l}^b \oplus s_{2l}^b\right) \tag{24.5}$$

where $\oplus$ is the exclusive OR operation.

Due to natural variability a real perfect match seldom occurs. In this case, we can allow a tolerance to have some flexibility. For a prescribed tolerance ε, we define

$$\mathbf{s}_i = \mathbf{s}_j \text{ if and only if HSD } (\mathbf{s}_i, \mathbf{s}_j) \text{ or AHSD } (\mathbf{s}_j, \mathbf{s}_j) \leq \varepsilon \tag{24.6}$$

One way to determine the distance tolerance ε was suggested in Chang et al. (2006a).

24.2.2 Median Partition Binary Coding

As an alternative to the spectral signature mean, μ used in (24.2), we can replace the spectral signature mean with the spectral signature median, m, to derive a new L-dimensional binary code word s_l^{MP} defined by

$$s_l^{\text{MP}} = \begin{cases} 1; & \text{if } s_l \geq \text{m} \\ 0; & \text{otherwise} \end{cases} \quad \text{for } l = 1, 2, \ldots, L \tag{24.7}$$

Coupling with s_l^b defined in (24.3), a new $(2L-2)$ -dimensional binary coding, called median partition (MP) binary coding, can be further derived from the SPAM binary coding and is represented by $\left(\mathbf{s}^{\mathrm{MP}}\,\mathbf{s}^b\right)$.

24.2.3 Halfway Partition Binary Coding

In this subsection, we propose a very simple binary coding, referred to as halfway partition (HP) binary coding that is borrowed from the idea of bit plane coding. It calculates the half distance between the maximal value and the minimal value of the spectrum of **s** and uses it as the threshold value to partition the spectral values of **s** into two disjoint sets as the way that is done by SPAM's binary coding with the spectral mean. More precisely, let λ be the halfway partition threshold given by

$$\lambda = (1/2)[\max_t\{s_t\} - \min_t\{s_t\}] \tag{24.8}$$

Replacing μ in (24.2) with λ by (24.8) we can encode a pixel vector **s** as a $(2L-2)$ -dimensional binary code word denoted by $\left(\mathbf{s}^{\mathrm{HP}}\,\mathbf{s}^b\right)$ where $\mathbf{s}^{\mathrm{HP}}$ is an L-dimensional binary code word defined by

$$s_l^{\mathrm{HP}} = \begin{cases} 1; & \text{if } s_l - \min_t\{s_t\} \geq \lambda \\ 0; & \text{otherwise} \end{cases} \quad \text{for } l = 1, 2, \ldots, L \tag{24.9}$$

and $\mathbf{s}^b = \left(s_2^b \cdots s_{L-1}^b\right)$ is also given by (24.3). The coding scheme using (24.8)–(24.9) and (24.3) is called halfway partition (HP) binary coding.

24.2.4 Equal Probability Partition Binary Coding

In this section, we present a rather different spectral binary coding method called equal probability partition (EPP) spectral coding. It is derived from the Shannon-Fano coding used for noiseless source coding (Lynch, 1985) and can be briefly described as follows.

Suppose that we are given by one bit to discriminate among a set of source alphabets that are governed by a probability distribution, how can we best use this bit? The idea is to partition the source alphabet set into two disjoint subsets so that their respective probabilities are as close as possible in terms of equal probability so as to achieve the maximum entropy. As an example for illustration, we assume that S is a source alphabet set consisting of four letters, $\{a, b, c, d\}$ with probabilities $p(a) = 1/2$, $p(b) = 1/4$, $p(c) = 1/8$, and $p(d) = 1/8$. In order to best utilize the given one bit, we need partition S into two subsets $S_1 = \{a\}$ and $S_2 = \{b, c, d\}$ so that both have probability ½, that is, $p(S_1) = p(a) = 1/2$ and $P(S_2) = p(b) + p(c) + p(d) = 1/2$, in which case the set partition into two subsets $\{S_1, S_2\}$ achieves the maximum entropy, which is exactly one bit. Any other set partition will not produce the 1-bit maximum entropy. For instance, if we group $S_1 = \{a, b\}$ and $S_2 = \{c, d\}$, the probabilities of S_1 and S_2 will be $p(S_1) = p(a) + p(b) = 3/4$ and $P(S_2) = p(c) + p(d) = 1/4$. As a result, the entropy associated with this set partition $\{S_1, S_2\}$ is given by $-(1/4\log(1/4) + 3/4\log(3/4)) = 0.811278$ bits that are less than 1 bit. This implies that the set partition of $S_1 = \{a, b\}$ and $S_2 = \{c, d\}$ is not optimal because $0.188722 = 1 - 0.811278$ bits are not used and have been wasted. As a result, the partition of S into $S_1 = \{a, b\}$ and $S_2 = \{c, d\}$ is not optimal because the given one bit is not fully utilized.

To be more specific, let $\mathbf{p} = (p_1, p_2, \ldots, p_L)^T$ be the probability vector associated with the s vector $\mathbf{s}$ that can be generated in the same way that was done for SID by

$$p_l = s_l / \sum\nolimits_{l=1}^{L} s_l \quad \text{for all } 1 \leq l \leq L \tag{24.10}$$

Now we can rearrange all the probabilities $\{p_l\}_{l=1}^{L}$ according to their decreasing orders in magnitude as $p_{l_1}, p_{l_2}, \ldots, p_{l_m}, p_{l_{m+1}}, \ldots, p_{l_L}$ and define an index k^* such that it satisfies the following equation:

$$l_{k^*} = \min\nolimits_{1 \leq k \leq L} \left\{ \left| \sum\nolimits_{m=1}^{k} p_{l_m} - \sum\nolimits_{m=k+1}^{L} p_{l_m} \right| \right\} \tag{24.11}$$

If let η denote the spectral value of the particular band l_{k^*} found by (24.11), that is, $\eta = s_{l_{k^*}}$, then replacing μ in (24.2) with η produces a new (2L-2)-dimensional binary code word, denoted by $\left(\mathbf{s}^{\text{EPP}}\, \mathbf{s}^b\right)$ where $\mathbf{s}^{\text{EPP}} = \left(s_1^{\text{EPP}}\, s_2^{\text{EPP}} \ldots s_L^{\text{EPP}}\right)$ is an L-dimensional binary code word given by

$$s_l^{\text{EPP}} = \begin{cases} 1; & \text{if } s_l \geq \eta \\ 0; & \text{otherwise} \end{cases} \quad \text{for } l = 1, 2, \ldots, L \tag{24.12}$$

and $\mathbf{s}^b = \left(s_2^b \ldots s_{L-1}^b\right)$ is still given by (24.3). The coding method of using (24.10), (24.11), and (24.3) is called EPP binary coding.

With different combinations of $\mathbf{s}^a$ in (24.2), $\mathbf{s}^{\text{MP}}$ in (24.7), $\mathbf{s}^{\text{HP}}$ in (24.9) and $\mathbf{r}^{\text{EPP}}(i,j)$ in (24.12) in conjunction with $\mathbf{s}^b$ in (24.3) we can obtain various binary coding methods such as $(\mathbf{s}^a\, \mathbf{s}^{\text{MP}}\, \mathbf{s}^{\text{HP}})$, $(\mathbf{s}^a\, \mathbf{s}^{\text{HP}}\, \mathbf{s}^{\text{EPP}})$, $(\mathbf{s}^a\, \mathbf{s}^{\text{MP}}\, \mathbf{s}^{\text{EPP}})$, $(\mathbf{s}^{\text{MP}}\, \mathbf{s}^{\text{HP}}\, \mathbf{s}^{\text{EPP}})$, $\left(\mathbf{s}^a\, \mathbf{s}^{\text{MP}}\, \mathbf{s}^{\text{HP}}\, \mathbf{s}^b\right)$, $\left(\mathbf{s}^a\, \mathbf{s}^{\text{MP}}\, \mathbf{s}^{\text{HP}}\, \mathbf{s}^{\text{EPP}}\right)$, $\left(\mathbf{s}^a\, \mathbf{s}^{\text{MP}}\, \mathbf{s}^{\text{EPP}}\, \mathbf{s}^b\right)$, $\left(\mathbf{s}^a\, \mathbf{s}^{\text{HP}}\, \mathbf{s}^{\text{EPP}},\, \mathbf{s}^b\right)$, $\left(\mathbf{s}^a\, \mathbf{s}^{\text{MP}}\, \mathbf{s}^{\text{HP}}\, \mathbf{s}^b\right)$, $\left(\mathbf{s}^{\text{MP}}\, \mathbf{s}^{\text{HP}}\, \mathbf{s}^{\text{EPP}}\, \mathbf{s}^b\right)$, $\left(\mathbf{s}^a\, \mathbf{s}^{\text{MP}}\, \mathbf{s}^{\text{HP}}\, \mathbf{s}^{\text{EPP}}\, \mathbf{s}^b\right)$, etc., that can be also used for spectral signature coding.

24.3 Spectral Feature-Based Coding

Recently, Qian et al. (1996) proposed a spectral feature binary coding (SFBC) that extended the SPAM binary coding as follows.

Let

$$s_l^{\text{MD}} = \begin{cases} 1; & \text{if } |s_l - \mu| \geq \text{MD} \\ 0; & \text{otherwise} \end{cases} \quad \text{for } l = 1, 2, \ldots, L \tag{24.13}$$

where MD is the spectral signature mean deviation given by $\text{MD} = (1/L) \sum_{l=1}^{L} |s_l - \mu|$. Concatenating $\mathbf{s}^a$, $\mathbf{s}^b$ and $\mathbf{s}^{\text{MD}} = \left(s_1^{\text{MD}}\, s_2^{\text{MD}} \cdots s_L^{\text{MD}}\right)$ defined in (24.2), (24.3), and (24.13), respectively, creates a new $(3\,L - 2)$ -dimensional binary code word, referred to as SFBC defined by

$$s^{\text{SFBC}} = \left(\mathbf{s}^a\, \mathbf{s}^b\, \mathbf{s}^{\text{MD}}\right) \tag{24.14}$$

Similarly, we can define the spectrals median partition deviation (MPD) by

$$\text{MPD} = (1/L) \sum\nolimits_{l=1}^{L} |s_l - \text{m}| \tag{24.15}$$

and

$$s_l^{\mathrm{MPD}} = \begin{cases} 1; & \text{if } |s_l - \mathrm{m}| \geq \mathrm{MPD} \\ 0; & \text{otherwise} \end{cases} \quad \text{for } l = 1, 2, \ldots, L \tag{24.16}$$

In analogy with the Qian et al.'s SFBC we can concatenate the L-dimensional binary code word $\mathbf{s}^{\mathrm{MPD}} = \left(s_1^{\mathrm{MPD}}\, s_2^{\mathrm{MPD}} \cdots s_L^{\mathrm{MPD}}\right)$ defined by (24.16) to the MP binary code word to create a new MP spectral feature-based $(3\,L-2)$ -dimensional binary code word $(\mathbf{s}^{\mathrm{MP}},\mathbf{s}^{\mathrm{b}},\mathbf{s}^{\mathrm{MPD}})^T$ for a signature vector $\mathbf{s}$ where $\mathbf{s}^{\mathrm{MP}}$, $\mathbf{s}^b$, and $\mathbf{s}^{\mathrm{MPD}}$ are given by (24.7), (24.3), and (24.16), respectively.

For the halfway partition (HP) binary coding, its spectral deviation can be also defined by

$$s_l^{\mathrm{HPD}} = \begin{cases} 1; & \text{if } |s_l - \lambda - \{\min_t s_t\}| \geq \mathrm{HPD} \\ 0; & \text{otherwise} \end{cases} \quad \text{for } l = 1, 2, \ldots, L \tag{24.17}$$

where HPD is the spectral halfway partition deviation given by

$$\mathrm{HPD} = (1/L) \sum\nolimits_{l=1}^{L} |s_l - \lambda - \{\min_t s_t\}| \tag{24.18}$$

Like $\left(\mathbf{s}^{\mathrm{MP}}\, \mathbf{s}^b\, \mathbf{s}^{\mathrm{MPD}}\right)$, including an L-dimensional binary code word $\mathbf{s}^{\mathrm{HPD}} = \left(s_1^{\mathrm{HPD}}\, s_2^{\mathrm{HPD}} \cdots s_L^{\mathrm{HPD}}\right)$ in the HP binary coding also creates a new $(3\,L-2)$ -dimensional HP spectral feature-based binary codeword $\left(\mathbf{s}^{\mathrm{HP}}\, \mathbf{s}^b\, \mathbf{s}^{\mathrm{HPD}}\right)$ for a signature $\mathbf{s}$ where $\mathbf{s}^{\mathrm{HP}}$, $\mathbf{s}^b$, and $\mathbf{s}^{\mathrm{HPD}}$ are given by (24.9), (24.3), and (24.17), respectively.

For the equal probability partition (EPP) binary coding, its deviation is defined by

$$s_l^{\mathrm{EPP}} = \begin{cases} 1; & \text{if } s_l \geq \eta \\ 0; & \text{otherwise} \end{cases} \quad \text{for } l = 1, 2, \ldots, L \tag{24.19}$$

where EPPD is the spectral deviation from EPP given by

$$\mathrm{EPPD} = (1/L) \sum\nolimits_{l=1}^{L} |s_l - \eta| \tag{24.20}$$

Similarly, we can also include the L-dimensional binary code word $\mathbf{s}^{\mathrm{EPPD}} = \left(s_1^{\mathrm{EPPD}}\, s_2^{\mathrm{EPPD}} \cdots s_L^{\mathrm{EPPD}}\right)$ given by (24.19) to produce a new EPP spectral feature-based $(3\,L-2)$ -dimensional binary codeword $\left(\mathbf{s}^{\mathrm{EPP}}\, \mathbf{s}^b\, \mathbf{s}^{\mathrm{EPPD}}\right)$ for a signature $\mathbf{s}$ where $\mathbf{s}^{\mathrm{EPP}}$, $\mathbf{s}^b$, and $\mathbf{s}^{\mathrm{EPPD}}$ are given by (24.12), (24.3), and (24.19), respectively.

As demonstrated above, numerous variants can be derived from different combinations of $\mathbf{s}^a$ in (24.2), $\mathbf{s}^{\mathrm{MP}}$ in (24.7), $\mathbf{s}^{\mathrm{HP}}$ in (24.9), $\mathbf{s}^{\mathrm{EPP}}$ in (24.12), $\mathbf{s}^b$ in (24.3) in conjunction with $\mathbf{s}^{\mathrm{MD}}$ in (24.13), $\mathbf{s}^{\mathrm{MPD}}$ in (24.16), $\mathbf{s}^{\mathrm{HPD}}$ in (24.17), and $\mathbf{s}^{\mathrm{EPPD}}$ in (24.19), for example, we just name a few as follows.

1. 4 L-dimensional binary code words such as
   $\left(\mathbf{s}^a\ \ \mathbf{s}^{\mathrm{MD}}\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{HPD}}\right)$, $\left(\mathbf{s}^a\ \ \mathbf{s}^{\mathrm{MD}}\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{EPPD}}\right)$, $\left(\mathbf{s}^a\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{HPD}}\ \ \mathbf{s}^{\mathrm{EPPD}}\right)$,
   $\left(\mathbf{s}^{\mathrm{MD}}\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{HPD}}\ \ \mathbf{s}^{\mathrm{EPPD}}\right)$, $\left(\mathbf{s}^{\mathrm{MP}}\ \ \mathbf{s}^{\mathrm{MD}}\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{HPD}}\right)$, $\left(\mathbf{s}^{\mathrm{MP}}\ \ \mathbf{s}^{\mathrm{MD}}\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{EPPD}}\right)$,
   $\left(\mathbf{s}^{\mathrm{MP}}\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{HPD}}\ \ \mathbf{s}^{\mathrm{EPPD}}\right)$, $\left(\mathbf{s}^{\mathrm{HP}}\ \ \mathbf{s}^{\mathrm{MD}}\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{HPD}}\right)$, $\left(\mathbf{s}^{\mathrm{HP}}\ \ \mathbf{s}^{\mathrm{MD}}\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{EPPD}}\right)$,
   $\left(\mathbf{s}^{\mathrm{HP}}\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{HPD}}\ \ \mathbf{s}^{\mathrm{EPPD}}\right)$, $\left(\mathbf{s}^{\mathrm{EPP}}\ \ \mathbf{s}^{\mathrm{MD}}\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{HPD}}\right)$, $\left(\mathbf{s}^{\mathrm{EPP}}\ \ \mathbf{s}^{\mathrm{MD}}\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{EPPD}}\right)$,
   $\left(\mathbf{s}^{\mathrm{EPP}}\ \ \mathbf{s}^{\mathrm{MPD}}\ \ \mathbf{s}^{\mathrm{HPD}}\ \ \mathbf{s}^{\mathrm{EPPD}}\right)$,

2. $(4\quad L-2)$-dimensional binary code words such as
   $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{EPP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{EPPD}}\right)$, $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{EPP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\right)$, $\left(\mathbf{r}^a(i,j),\quad \mathbf{r}^{\mathrm{HP}}(i,j),\quad \mathbf{r}^b(i,j),\quad \mathbf{r}^{\mathrm{MD}}(i,j)\right)^T$, $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{HP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{HPD}}\right)$, $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{MP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\right)$, $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{MP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MPD}}\right)$,
3. $(5\quad L-2)$-dimensional binary code words such as
   $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{MP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\quad \mathbf{s}^{\mathrm{MPD}}\right)$, $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{MP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\quad \mathbf{s}^{\mathrm{HPD}}\right)$ $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{MP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\quad \mathbf{s}^{\mathrm{EPPD}}\right)$, $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{HP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\quad \mathbf{s}^{\mathrm{MPD}}\right)$, $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{HP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\quad \mathbf{s}^{\mathrm{HPD}}\right)$, $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{HP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\quad \mathbf{s}^{\mathrm{EPPD}}\right)$, $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{EPP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\quad \mathbf{s}^{\mathrm{MPD}}\right)$, $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{EPP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\quad \mathbf{s}^{\mathrm{HPD}}\right)$, $\left(\mathbf{s}^a\quad \mathbf{s}^{\mathrm{EPP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\quad \mathbf{s}^{\mathrm{EPPD}}\right)$, $\left(\mathbf{s}^{\mathrm{MP}}\quad \mathbf{s}^{\mathrm{HP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\quad \mathbf{s}^{\mathrm{HPD}}\right)$.

24.4 Experiments

Two data sets were used for computer simulations and real hyperspectral image experiments. As noted, many binary coding methods can be derived by various combinations of $\mathbf{s}^a$, $\mathbf{s}^b$, $\mathbf{s}^{\mathrm{MP}}$, $\mathbf{s}^{\mathrm{HP}}$, $\mathbf{s}^{\mathrm{EPP}}$, $\mathbf{s}^{\mathrm{MD}}$, $\mathbf{s}^{\mathrm{MPD}}$, $\mathbf{s}^{\mathrm{HPD}}$, and $\mathbf{s}^{\mathrm{EPPD}}$. So, the experiments presented in this section did not intend to represent a comprehensive study on all possible combinations. Instead, they intended to provide a glimpse of performance analysis on a selective set of eight binary coding methods that we believed to be representative. These include four SPAM-based binary coding methods: $\left(\mathbf{s}^a\quad \mathbf{s}^b\right)$ (SPAM), $\left(\mathbf{s}^{\mathrm{MP}}\quad \mathbf{s}^b\right)$ (MP), $\left(\mathbf{s}^{\mathrm{HP}}\quad \mathbf{s}^b\right)$ (HP) and $\left(\mathbf{s}^{\mathrm{EPP}}\quad \mathbf{s}^b\right)$ (EPP), and four Qian et al.'s SFBC methods: $\left(\mathbf{s}^a\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MD}}\right)$ (Qian et al.'s SFBC = SPAM + MD), $\left(\mathbf{s}^{\mathrm{MP}}\quad \mathbf{s}^b\quad \mathbf{s}^{\mathrm{MPD}}\right)$(MP + MPD), $\left(\mathbf{s}^{\mathrm{HP}}\, \mathbf{s}^b\, \mathbf{s}^{\mathrm{HPD}}\right)$ (HP + HPD), $\left(\mathbf{s}^{\mathrm{EPP}}\, \mathbf{s}^b\, \mathbf{s}^{\mathrm{EPPD}}\right)$ (EPP + EPPD).

24.4.1 Computer Simulations

The first data set to be used for experiments was the AVIRIS (airborne visible/infrared imaging spectrometer) laboratory reflectance data shown in Figure 1.8 where five-field reflectance spectra, blackbrush, creosote leaves, dry grass, red soil, and sagebrush with spectral coverage from 0.4 to 2.5 μm and 158 bands after the water bands are removed. With these five signatures a performance analysis was conducted for comparison. Table 24.1(a) and (b) tabulates the results produced by SPAM, MP, HP, EPP, Qian et al.'s method SFBC (SPAM + MD), MP + MPD, HP + HPD, and EPP + EPPD using HSD and AHSD, respectively, to distinguish the signature of blackbrush from the other four signatures where all the eight binary coding methods showed that the blackbrush was most close to the sagebrush. Also included in Table 24.1(a) for comparison is a commonly used spectral measures, Euclidean distance (ED) that simply measured the distance between two

Table 24.1(a) HSD and normalized HSD values between blackbrush (B) and creosote leaves (C), dry grass (D), red soil (R), and sagebrush (S)

Blackbrush (B)	C	D	R	S
SPAM	35(0.1301)	73(0.2714)	132(0.4907)	29(0.1078)
MP	29(0.1094)	82(0.3094)	135(0.594)	19(0.0717)
HP	38(0.1532)	86(0.3468)	92(0.371)	32(0.129)
EPP	38(0.1267)	97(0.3233)	142(0.4733)	23(0.0767)
SFBC = SPAM + MD	66(0.1312)	147(0.2922)	245(0.4871)	45(0.0895)
MP + MPD	70(0.1452)	146(0.3029)	218(0.4523)	48(0.0996)
HP + HPD	91(0.1743)	167(0.3199)	199(0.3812)	65(0.1245)
EPP + EPPD	59(0.1388)	117(0.2753)	214(0.5035)	35(0.0824)
ED	1.4873(0.1622)	3.6261(0.3955)	2.5471(0.2778)	1.5087(0.1645)

Table 24.1(b) AHSD and normalized AHSD values between blackbrush (B) and creosote leaves (C), dry grass (D), red soil (R), and sagebrush (S)

Blackbrush (B)	C	D	R	S
SPAM	0.1115	0.2325	0.4204	0.0924
MP	0.0924	0.2611	0.4299	0.0605
HP	0.121	0.2739	0.293	0.1019
EPP	0.121	0.3089	0.4522	0.0732
SFBC = SPAM + MD	0.1398	0.3114	0.5191	0.0953
MP + MPD	0.1483	0.3093	0.4619	0.1017
HP + HPD	0.1928	0.3538	0.4216	0.1377
EPP + EPPD	0.125	0.2479	0.4534	0.0742

signature vectors without encoding. As shown in these two tables, HSD and AHSD produced the same results with MP being the best method. It should be noted that in all the following tables, the highlighted values yielded the smallest values for each method and identified that signatures were most close to the signatures to be compared and the highlighted method was the best in the sense that its value was the smallest among the highlighted values.

Interestingly, if we would like to know if one method is more effective than another, using the Hamming spectral distance values to compare relative performance among these eight binary coding methods may be difficult. To cope with this problem, we normalized the Hamming spectral distance values and used their relative probabilities to evaluate the discriminability for each binary coding method. The numbers in parentheses in Tables 24.1(a) and (b) were normalized HSD, normalized AHSD, and normalized ED values. For example, the numbers, 35, 73, 132, and 29 in the first row of Table 24.1(a) were the HSD values produced by the SPAM between blackbrush and the other four signatures, creosote leaves, dry grass, red soil, and sagebrush, respectively. The values in parentheses were their corresponding normalized Hamming spectral distance values obtained by $0.1301 = 35/(35 + 73 + 132 + 29)$, $0.2714 = 73/(35 + 73 + 132 + 29)$, $0.4907 = 132/(35 + 73 + 132 + 29)$, and $0.1078 = 29/(35 + 73 + 132 + 29)$, respectively. Using these normalized values, the highlighted MP yielded the smallest values in Table 24.1(a) that implies that the MP was the most effective among the eight methods.

Similarly, Tables 24.2(a) and (b)–24.5(a) and (b) tabulate the results produced by SPAM, MP, HP, EPP, Qian et al.'s method SFBC (SPAM + MD), MP + MPD, HP + HPD, and EPP + EPPD using HSD and AHSD, respectively, along with ED to distinguish the signature of creosote leaves

Table 24.2(a) HSD and normalized HSD values between creosote leaves (C) and blackbrush (B), dry grass (D), red soil (R), and sagebrush (S)

Creosote leaves (C)	B	D	R	S
SPAM	35(0.1199)	90(0.3082)	153(0.524)	14(0.0479)
MP	29(0.1028)	89(0.3156)	146(0.5177)	18(0.0638)
HP	38(0.1348)	110(0.3901)	120(0.4255)	14(0.0496)
EPP	38(0.1131)	121(0.3601)	154(0.4583)	23(0.0685)
SFBC = SPAM + MD	66(0.1164)	187(0.3298)	281(0.4956)	33(0.0582)
MP + MPD	70(0.1346)	188(0.3615)	230(0.4423)	32(0.0615)
HP + HPD	91(0.1588)	230(0.4014)	218(0.3805)	34(0.0593)
EPP + EPPD	59(0.1185)	162(0.3253)	245(0.492)	32(0.0643)
ED	1.4873(0.1875)	3.0833(0.3888)	2.7578(0.3477)	0.6021(0.0759)

Table 24.2(b) AHSD and normalized AHSD values between creosote leaves (C) and blackbrush (B), dry grass (D), red soil (R), and sagebrush (S)

Creosote leaves (C)	B	D	R	S
SPAM	0.1115	0.2866	0.4873	0.0446
MP	0.0924	0.2834	0.465	0.0573
HP	0.121	0.3503	0.3822	0.0446
EPP	0.121	0.3854	0.4904	0.0732
SFBC = SPAM + MD	0.1398	0.3962	0.5953	0.0699
MP + MPD	0.1483	0.3983	0.4873	0.0678
HP + HPD	0.1928	0.4873	0.4619	0.072
EPP + EPPD	0.125	0.3432	0.5191	0.0678

from the other four signatures, the signature of dry grass from the other four signatures, the signature of red soil from the other four signatures and the signature of sagebrush from the other four signatures, respectively. Once again the highlighted values yielded the smallest values for each method and identified that signatures were most close to the signatures to be compared and the highlighted method was the best in the sense that its value was the smallest among the highlighted values. According to these tables, all the eight binary coding methods produced nearly the same

Table 24.3(a) HSD and normalized HSD values between dry grass (D) and blackbrush (B), creosote leaves (C), red soil (R), and sagebrush (S)

Dry grass (D)	B	C	R	S
SPAM	73(0.2039)	90(0.2514)	111(0.3101)	84(0.2346)
MP	82(0.2284)	89(0.2479)	105(0.2925)	83(0.2312)
HP	86(0.2337)	110(0.2989)	68(0.1848)	104(0.2826)
EPP	97(0.2304)	121(0.2874)	97(0.2304)	106(0.2518)
SFBC = SPAM + MD	147(0.2194)	187(0.2791)	170(0.2537)	166(0.2478)
MP + MPD	146(0.2141)	188(0.2757)	178(0.261)	170(0.2493)
HP + HPD	167(0.226)	230(0.3112)	128(0.1732)	214(0.2896)
EPP + EPPD	117(0.199)	162(0.2755)	171(0.2908)	138(0.2347)
ED	3.6261(0.3247)	3.0833(0.2761)	1.9236(0.1722)	2.5361(0.2271)

Table 24.3(b) AHSD and normalized AHSD values between dry grass (D) and blackbrush (B), creosote leaves (C), red soil (R), and sagebrush (S)

Dry grass (D)	B	C	R	S
SPAM	0.2325	0.2866	0.3535	0.2675
MP	0.2611	0.2834	0.3344	0.2643
HP	0.2739	0.3503	0.2166	0.3312
EPP	0.3089	0.3854	0.3089	0.3376
SFBC = SPAM + MD	0.3114	0.3962	0.3602	0.3517
MP + MPD	0.3093	0.3983	0.3771	0.3602
HP + HPD	0.3538	0.4873	0.2712	0.4534
EPP + EPPD	0.2479	0.3432	0.3623	0.2924

Table 24.4(a) HSD and normalized HSD values between red soil (R) and blackbrush (B), creosote leaves (C), dry grass (D), and sagebrush (S)

Red soil (R)	B	C	D	S
SPAM	132(0.2413)	153(0.2797)	111(0.2029)	151(0.2761)
MP	135(0.2547)	146(0.2755)	105(0.1981)	144(0.2717)
HP	92(0.2312)	120(0.3015)	68(0.1709)	118(0.2965)
EPP	142(0.2582)	154(0.28)	97(0.1764)	157(0.2855)
SFBC = SPAM + MD	245(0.2536)	281(0.2909)	170(0.176)	270(0.2795)
MP + MPD	218(0.2547)	230(0.2687)	178(0.2079)	230(0.2687)
HP + HPD	199(0.2636)	218(0.2887)	128(0.1695)	210(0.2781)
EPP + EPPD	214(0.2463)	245(0.219)	171(0.1968)	239(0.275)
ED	2.5471(0.2696)	2.7578(0.2918)	1.9236(0.2036)	2.2209(0.2350)

Table 24.4(b) AHSD and normalized AHSD values between red soil (R) and blackbrush (B), creosote leaves (C), dry grass (D), and sagebrush (S)

Red soil (R)	B	C	D	S
SPAM	0.4204	0.4873	0.3535	0.4809
MP	0.4299	0.465	0.3344	0.4586
HP	0.293	0.3822	0.2166	0.3758
EPP	0.4522	0.4904	0.3089	0.5
SFBC = SPAM + MD	0.5191	0.5953	0.3602	0.572
MP + MPD	0.4619	0.4873	0.3771	0.4873
HP + HPD	0.4216	0.4619	0.2712	0.4449
EPP + EPPD	0.4534	0.5191	0.3623	0.5064

results using the HSD and AHSD except one case in Tables 24.3(a) and (b) where EPP + EPPD was the best using HSD in Table 24.3(a) compared to HP using AHSD, which was the best in Table 24.3(b). Once again, ED never became a contender as the best method in all cases.

Obviously, according to Tables 24.1(a) and (b)–Tables 24.5(a) and (b), there was no single method that could be considered to be the best. This is a natural conclusion because every signature vector behaves very differently in terms of spectral characteristics and each of the eight considered binary coding methods has its own strengths and weaknesses in capturing spectral variability. Consequently, it can be expected that none of them may perform uniformly superior to another over all signature vectors. Additionally, the two distance measures, HSD and AHSD, also yielded nearly the same discrimination results but selected different best coding methods such as Tables 24.3(a) and (b) and Tables 24.4(a) and (b). More interestingly, ED never occurred as a best distance measure. This implies that a best binary coding method always performed better than a spectral measure such as ED. Furthermore, the results in these tables also suggested that using additional L-dimensional binary code words to account for interband spectral deviation as proposed by Qian et al. (1996) did not offer much advantage in spectral discrimination. As a matter of fact, only two out of five cases using HSD (i.e., Tables 24.3(a), 24.4(a)) and none of five cases using AHSD took advantage of the spectral deviation. The signature vectors, blackbrush, creosote leaves, and sagebrush in these three cases happened to be very similar compared to the other two cases of the dry grass and red soil, which are rather distinct. The results in Tables 24.3(a) and (b) and Tables 24.4(a) and (b) were also interesting. For example, from Tables 24.3(a) and (b) the spectral signature of dry grass was shown to be most close to either the blackbrush (such as

Table 24.5(a) HSD and normalized HSD values between sagebrush (S) and blackbrush (B), creosote leaves (C), dry grass (D), and red soil (R)

Sagebrush (S)	B	C	D	R
SPAM	29(0.1043)	14(0.0504)	84(0.3022)	151(0.5432)
MP	19(0.72)	18(0.0682)	83(0.3144)	144(0.5455)
HP	32(0.1194)	14(0.0522)	104(0.3881)	118(0.4403)
EPP	23(0.0744)	23(0.0744)	106(0.343)	157(0.5081)
SFBC = SPAM + MD	45(0.875)	33(0.0642)	166(0.323)	270(0.5253)
MP + MPD	48(0.10)	32(0.0667)	170(0.3542)	230(0.4792)
HP + HPD	65(0.1243)	34(0.065)	214(0.4092)	210(0.4015)
EPP + EPPD	35(0.788)	32(0.0721)	138(0.3108)	239(0.5383)
ED	1.5087(0.2197)	0.6021(0.0877)	2.5361(0.3693)	2.2209(0.3234)

SPAM, MP, SPAM + MD, MP + MPD, EPP + EPPD in Table 24.3(a)) or red soil (such as HP, HP + HPD in Table 24.3(a)) or both (such as EPP in Table 24.3(a)). However, on the other hand, Tables 24.4(a) and (b) showed that all the eight methods indicated that the spectral signature of red soil was most close to that of dry grass. When EPP produced the same Hamming spectral distance for blackbrush and red soil in Tables 24.3(a) and (b), its counterpart with inclusion of interband spectral deviation, EPP + EPPD broke the tie. A similar phenomenon was also observed in Tables 24.5(a) and (b).

In order for binary coding to further perform spectral identification, we assumed that a spectral signature data base, $\Delta = \{\mathbf{s}_j\}_{j=1}^{K}$ was provided for this purpose. Then for any given signature vector $\mathbf{r}$ we define APDP for each of spectral signature vectors $\mathbf{s}_i$ in Δ with respect to $\mathbf{r}$, $p_{\mathrm{APDP/HSD}}(\mathbf{s}_i|\mathbf{r},\Delta)$ using HSD as

$$p_{\mathrm{APDP/HSD}}(\mathbf{s}_i|\mathbf{r},\Delta) = \frac{\mathrm{HSD}(\mathbf{r},\mathbf{s}_i)}{\sum_{j=1}^{K}\mathrm{HSD}(\mathbf{r},\mathbf{s}_j)} \tag{24.21}$$

where HSD($\mathbf{r}$,$\mathbf{s}_i$) and HSD($\mathbf{r}$,$\mathbf{s}_j$) are spectral distance measures defined by HSD in (24.4) and AHSD($\mathbf{r}$,$\mathbf{s}_i$) and AHSD($\mathbf{r}$,$\mathbf{s}_j$) are spectral distance measures defined by AHSD in (24.5). In this case, the signature vector $\mathbf{r}$ will be identified by the one that yields the smallest APDP according to (24.21) or (24.22). In other words, the signature vector $\mathbf{r}$ identified by a spectral signature vector in Δ with the smallest APDP was the one that had the least probability of discrimination from $\mathbf{r}$. To illustrate the utility of APDP, let Δ = {blackbrush, creosote leaves, dry grass, red soil,

Table 24.5(b) AHSD and normalized AHSD values between sagebrush (S) and blackbrush (B), creosote leaves (C), dry grass (D), and red soil (R)

Sagebrush (S)	B	C	D	R
SPAM	0.4809	0.0446	0.2675	0.4809
MP	0.4586	0.0573	0.2643	0.4586
HP	0.3758	0.0446	0.3312	0.3758
EPP	0.5	0.0732	0.3376	0.5
SFBC = SPAM + MD	0.572	0.0699	0.3517	0.572
MP + MPD	0.4873	0.0678	0.3602	0.4873
HP + HPD	0.4449	0.072	0.4534	0.4449
EPP + EPPD	0.5064	0.0678	0.2924	0.5064

sagebrush}. Tables 24.6(a) and (b) to Tables 24.10(a) and (b) tabulate the results produced by the eight binary coding methods, SPAM, MP, HP, EPP, Qian et al.'s method SFBC (SPAM + MD), MP + MPD, HP + HPD, and EPP + EPPD using HSD and AHSD, respectively, to identify a signature vector **r** where the signature vector **r** in each table was randomly generated and mixed by the five signature vectors in the Δ as indicated in the tables. The highlighted values are the smallest APDP values that identified the mixed signure vector **r** for each method, and the highlighted method, was the one that yielded the best result among the eight methods.

Once again both distance measures, HSD and AHSD, produced nearly identical results except in the case of Table 24.7(b) where AHSD had a tie between SPAM and HP as opposed to the SPAM selected by HSD. Furthermore, a simialr conclusion resulting from the discrimination results in Tables 24.1(a) and (b) to Tables 24.5(a) and (b) can be also made. The results in Tables 24.6(a) and (b)–Tables 24.10(a) and (b) did not provide any evidence that including additional *L*-dimensional binary code words to account for interband spectral deviation would improve better spectral identification.

24.4.2 Real Hyperspectral Image Data

The second data set used for experiments was a real HYDICE (hyperspectral digital imagery collection experiment) image shown in Figure 1.15(a) and five panel signatures in Figure 1.16 were used for experiments. Following a similar treatment conducted for computer simulations in Section 24.4.1 two experiments are considered, one for spectral discrimination and the other for spectral identification. Tables 24.11(a) and (b)–Tables 23.15(a) and (b) tabulate results produced

Table 24.6(a) APDP/HSD identification of a mixed signature = 0.8055*B + 0.0292*C + 0.0272*D + 0.0588*R + 0.0944*S

r	B	*C*	D	R	S
SPAM	0.0297	*0.1375*	0.2491	0.4684	0.1152
MP	0.0232	*0.1042*	0.3012	0.5058	0.0656
HP	0.0543	*0.186*	0.2868	0.3101	0.1628
EPP	0.0166	*0.1229*	0.3189	0.4684	0.0731
SFBC = SPAM + MD	0.0276	*0.142*	0.2742	0.4596	0.0966
MP + MPD	0.0222	*0.1475*	0.2889	0.4384	0.103
HP + HPD	0.0414	*0.1982*	0.2631	0.3459	0.1514
EPP + EPPD	0.0164	*0.1402*	0.2664	0.493	0.0841
ED	0.0533	*0.15*	0.3915	0.2695	0.1366

Table 24.6(b) APDP/AHSD identification of a mixed signature = 0.8055*B + 0.0292*C + 0.0272*D + 0.0588*R + 0.0944*S

r	B	C	D	R	S
SPAM	0.0255	0.1178	0.2134	0.4013	0.0987
MP	0.0191	0.086	0.2484	0.4172	0.0541
HP	0.0446	0.1529	0.2357	0.2548	0.1338
EPP	0.0159	0.1178	0.3057	0.449	0.0701
SFBC=SPAM+MD	0.0297	0.1525	0.2945	0.4936	0.1038
MP+MPD	0.0233	0.1547	0.303	0.4597	0.1081
HP+HPD	0.0487	0.2331	0.3093	0.4068	0.178
EPP+EPPD	0.0148	0.1271	0.2415	0.447	0.0763

Table 24.7(a) APDP/HSD identification of a mixed signature = 0.1055*B + 0.7292*C + 0.0272*D + 0.0588*R + 0.0944*S

r	B	C	D	R	S
SPAM	0.1139	0.0178	0.3025	0.5267	0.0391
MP	0.892	0.026	0.3123	0.5242	0.0483
HP	0.1292	0.0185	0.3875	0.4244	0.0406
EPP	0.972	0.0282	0.3511	0.4734	0.0502
SFBC = SPAM + MD	0.859	0.0439	0.3244	0.5076	0.0382
MP + MPD	0.1066	0.041	0.3484	0.4631	0.041
HP + HPD	0.1375	0.0353	0.4071	0.3885	0.0316
EPP + EPPD	0.0939	0.0393	0.3144	0.5087	0.0437
ED	0.1909	0.047	0.3887	0.3333	0.04

Table 24.7(b) APDP/AHSD identification of a mixed signature = 0.1055*B + 0.7292*C + 0.0272*D + 0.0588*R + 0.0944*S

r	B	C	D	R	S
SPAM	0.1019	0.0159	0.2707	0.4713	0.035
MP	0.0764	0.0223	0.2675	0.449	0.0414
HP	0.1115	0.0159	0.3344	0.3662	0.035
EPP	0.0987	0.0287	0.3567	0.4809	0.051
SFBC = SPAM + MD	0.0953	0.0487	0.3602	0.5636	0.0424
MP + MPD	0.1102	0.0424	0.3602	0.4788	0.0424
HP + HPD	0.1568	0.0403	0.464	0.4428	0.036
EPP + EPPD	0.0911	0.0381	0.3051	0.4936	0.0424

by SPAM, MP, HP, EPP, Qian et al.'s method SFBC (SPAM + MD), MP + MPD, HP + HPD, and EPP + EPPD using the HSD and AHSD, respectively, to discriminate one panel signature from the other four panel signatures. According to the ground truth, the panels in rows 2 and 3 are made by the same material with different paints. As expected, panel signatures $\mathbf{p}_2$ and $\mathbf{p}_3$ should be very similar. So are panels in rows 4 and 5. From the results in Tables 24.11(a) and (b)–Tables 24.15(a) and (b) the eight binary coding methods using HSD and AHSD produced the same identification results and the Qian et al.'s method SFBC (SPAM + MD) produced the lowest either

Table 24.8(a) APDP/HSD identification of a mixed signature = 0.1055*B + 0.0292*C + 0.7272*D + 0.0588*R + 0.0944*S

r	B	C	D	R	S
SPAM	0.1745	0.2336	0.053	0.3302	0.2087
MP	0.1767	0.205	0.0946	0.3438	0.1798
HP	01753	0.2597	0.1039	0.2273	0.2338
EPP	0.1872	0.2567	0.0722	0.2727	0.2112
SFBC = SPAM + MD	0.1754	0.2508	0.0656	0.3016	0.2066
MP + MPD	0.1863	0.2546	0.1038	0.2319	0.2233
HP + HPD	0.1758	0.2773	0.0773	0.2227	0.247
EPP + EPPD	0.1555	0.248	0.0748	0.3248	0.1969
ED	0.3119	0.2582	0.07	0.1612	0.1981

Table 24.8(b) APDP/AHSD identification of a mixed signature = 0.1055*B + 0.0292*C + 0.7272*D + 0.0588*R + 0.0944*S

r	B	C	D	R	S
SPAM	0.1783	0.2389	0.0541	0.3376	0.2134
MP	0.1783	0.207	0.0955	0.3471	0.1815
HP	0.172	0.2548	0.1019	0.2229	0.2293
EPP	0.2229	0.3057	0.086	0.3248	0.2516
SFBC = SPAM + MD	0.2267	0.3242	0.0847	0.3898	0.2669
MP + MPD	0.2775	0.3792	0.1547	0.3453	0.3326
HP + HPD	0.2458	0.3877	0.1081	0.3114	0.3453
EPP + EPPD	0.1674	0.2669	0.0805	0.3496	0.2119

Table 24.9(a) APDP/HSD identification of a mixed signature = 0.1055*B + 0.0292*C + 0.0272*D + 0.7588*R + 0.0944*S

r	B	C	D	R	S
SPAM	0.2025	0.2543	0.1704	0.1235	0.2494
MP	0.2182	0.2468	0.161	0.1325	0.2416
HP	0.2133	0.2909	0.169	0.0416	0.2853
EPP	0.2071	0.2778	0.1136	0.1515	0.25
SFBC = SPAM + MD	0.2241	0.2797	0.1443	0.0861	0.2658
MP + MPD	0.2322	0.2568	0.1667	0.0874	0.2568
HP + HPD	0.2355	0.2961	0.105	0.0781	0.2853
EPP + EPPD	0.2034	0.2758	0.1495	0.1263	0.245
ED	0.2526	0.2667	0.2209	0.0564	0.2033

normalized HSD values or AHSD values in discrimination of all the five panel signature vectors $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, $\mathbf{p}_5$. Once again, in all the following tables, the highlighted values yielded the smallest values for each method and identified that signatures were most close to the signatures to be compared and the highlighted method was the best in the sense that its value was the smallest among the highlighted values.

The above real image experiment demonstrated an advantage of using extra *L*-binary code words implemented in SFBC that could account for spectral variability caused by atmospheric or interfering effects present in real data. A similar conclusion can be also drawn for real subpixel panel identification conducted in the following experiments.

Table 24.9(b) APDP/AHSD identification of a mixed signature = 0.1055*B + 0.0292*C + 0.0272*D + 0.7588*R + 0.0944*S

r	B	C	D	R	S
SPAM	0.2611	0.328	0.2197	0.1592	0.3217
MP	0.2675	0.3025	0.1975	0.1624	0.2962
HP	0.2452	0.3344	0.1943	0.0478	0.328
EPP	0.2611	0.3503	0.1433	0.1911	0.3153
SFBC = SPAM + MD	0.375	0.4682	0.2415	0.1441	0.4449
MP + MPD	0.3602	0.3983	0.2585	0.1356	0.3983
HP + HPD	0.3708	0.4661	0.1653	0.1229	0.4492
EPP + EPPD	0.2797	0.3792	0.2055	0.1737	0.3369

Table 24.10(a) APDP/HSD identification of a mixed signature = 0.1055*B + 0.0292*C + 0.0272*D + 0.0588*R + 0.7944*S

r	B	C	D	R	S
SPAM	0.0852	0.0667	0.2889	0.5296	0.0296
MP	0.0654	0.0692	0.3115	0.5385	0.0154
HP	0.093	0.0775	0.3721	0.4186	0.0388
EPP	0.0623	0.082	0.3344	0.5016	0.0197
SFBC = SPAM + MD	0.0639	0.0878	0.3054	0.509	0.0339
MP + MPD	0.0672	0.0966	0.3361	0.4622	0.0378
HP + HPD	0.0602	0.1222	0.3477	0.4041	0.0658
EPP + EPPD	0.0554	0.0947	0.2933	0.5266	0.03
ED	0.2128	0.1093	0.3542	0.2974	0.0262

Table 24.10(b) APDP/AHSD identification of a mixed signature = 0.1055*B + 0.0292*C + 0.0272*D + 0.0588*R + 0.7944*S

r	B	C	D	R	S
SPAM	0.0732	0.0573	0.2484	0.4554	0.0255
MP	0.0541	0.0573	0.258	0.4459	0.0127
HP	0.0764	0.0637	0.3057	0.3439	0.0318
EPP	0.0605	0.0796	0.3248	0.4873	0.0191
SFBC = SPAM + MD	0.0678	0.0932	0.3242	0.5403	0.036
MP + MPD	0.0678	0.0975	0.339	0.4661	0.0381
HP + HPD	0.0678	0.1377	0.3919	0.4555	0.0742
EPP + EPPD	0.0508	0.0869	0.2691	0.4831	0.0275

Since the five panels p_{13}, p_{23}, p_{33}, p_{43}, p_{53} in the third column of Figure 1.15(b) have only size of 1 m × 1 m, which is smaller than the pixel resolution 1.56 m × 1.56 m, they cannot be seen visually and can only be identified at the subpixel level. In this case, the eight binary coding methods are used in conjunction with the minimum APDP criterion defined in (24.21) and (24.22) for

Table 24.11(a) HSD and normalized HSD values between $\mathbf{p}_1$ and $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, $\mathbf{p}_5$

$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	16(0.1441)	16(0.1441)	36(0.3243)	43(0.3874)
MP	17(0.1977)	15(0.1744)	22(0.2558)	32(0.3721)
HP	16(0.1975)	17(0.2099)	22(0.2716)	26(0.321)
EPP	17(0.1977)	19(0.2209)	23(0.2674)	27(0.314)
SFBC = SPAM + MD	19(0.145)	20(0.1527)	42(0.3206)	50(0.3817)
MP + MPD	21(0.1875)	18(0.1607)	30(0.2679)	43(0.3839)
HP + HPD	23(0.1983)	26(0.2241)	30(0.2586)	37(0.319)
EPP + EPPD	20(0.1961)	23(0.2255)	28(0.2745)	31(0.3039)
ED	1301.6(0.106)	2033.3(0.1657)	4107.3(0.3346)	4831.6(0.3937)

Table 24.11(b) AHSD and normalized AHSD values between $\mathbf{p}_1$ and $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, $\mathbf{p}_5$

$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	0.051	0.051	0.1146	0.1369
MP	0.0541	0.0478	0.0701	0.1019
HP	0.051	0.0541	0.0701	0.0828
EPP	0.0541	0.0605	0.0732	0.086
SFBC = SPAM + MD	0.0403	0.0424	0.089	0.1059
MP + MPD	0.0445	0.0381	0.0636	0.0911
HP + HPD	0.0487	0.0551	0.0636	0.0784
EPP + EPPD	0.0424	0.0487	0.0593	0.0657

Table 24.12(a) HSD and normalized HSD values between $\mathbf{p}_2$ and $\mathbf{p}_1$, $\mathbf{p}_3$, $\mathbf{p}_4$, $\mathbf{p}_5$

$\mathbf{p}_2$	$\mathbf{P}_1$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	16(0.1416)	16(0.1416)	36(0.3186)	45(0.3982)
MP	17(0.1954)	16(0.1839)	21(0.2414)	33(0.3793)
HP	16(0.1975)	15(0.1852)	22(0.2716)	28(0.3457)
EPP	17(0.20)	18(0.2118)	22(0.2588)	28(0.3294)
SFBC = SPAM + MD	19(0.1439)	17(0.1288)	43(0.3258)	53(0.4015)
MP + MPD	21(0.1736)	19(0.157)	33(0.2727)	48(0.3967)
HP + HPD	23(0.187)	17(0.1382)	37(0.3008)	46(0.374)
EPP + EPPD	20(0.1961)	23(0.2255)	26(0.2549)	33(0.3235)
ED	1301.6(0.0969)	1340.4(0.0997)	5064.1(0.3768)	5733(0.4266)

Table 24.12(b) AHSD and normalized AHSD values between $\mathbf{p}_2$ and $\mathbf{p}_1$, $\mathbf{p}_3$, $\mathbf{p}_4$, $\mathbf{p}_5$

$\mathbf{p}_2$	$\mathbf{P}_1$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	0.051	0.051	0.1146	0.1433
MP	0.0541	0.051	0.0669	0.1051
HP	0.051	0.0478	0.0701	0.0892
EPP	0.0541	0.0573	0.0701	0.0892
SFBC = SPAM + MD	0.0403	0.036	0.0911	0.1123
MP + MPD	0.0445	0.0403	0.0699	0.1017
HP + HPD	0.0487	0.036	0.0784	0.0975
EPP + EPPD	0.0424	0.0487	0.0551	0.0699

Table 24.13(a) HSD and normalized HSD values between $\mathbf{p}_3$ and $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_4$, $\mathbf{p}_5$

$\mathbf{p}_3$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{P}_4$	$\mathbf{P}_5$
SPAM	16(0.1345)	16(0.1345)	40(0.3361)	47(0.395)
MP	15(0.1648)	16(0.1758)	25(0.2747)	35(0.3846)
HP	17(0.1889)	15(0.1667)	27(0.30)	31(0.3444)
EPP	19(0.1881)	18(0.1782)	30(0.297)	34(0.3366)
SFBC = SPAM + MD	20(0.1418)	17(0.1206)	48(0.3404)	56(0.3972)
MP + MPD	18(0.1475)	19(0.1557)	36(0.2951)	49(0.4016)
HP + HPD	26(0.1884)	17(0.1232)	44(0.3188)	51(0.3696)
EPP + EPPD	23(0.1811)	23(0.1811)	39(0.3071)	42(0.3307)
ED	2033.3(0.1376)	1340.4(0.0907)	5434.1(0.3678)	5968.7(0.4039)

Table 24.13(b) HSD and normalized HSD values between $\mathbf{p}_3$ and $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_4$, $\mathbf{p}_5$

$\mathbf{p}_3$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{P}_4$	$\mathbf{P}_5$
SPAM	0.051	0.051	0.1274	0.1497
MP	0.0478	0.051	0.0796	0.1115
HP	0.0541	0.0478	0.086	0.0987
EPP	0.0605	0.0573	0.0955	0.1083
SFBC = SPAM + MD	0.0424	0.036	0.1017	0.1186
MP + MPD	0.0381	0.0403	0.0763	0.1038
HP + HPD	0.0551	0.036	0.0932	0.1081
EPP + EPPD	0.0487	0.0487	0.0826	0.089

Table 24.14(a) HSD and normalized HSD values between $\mathbf{p}_4$ and $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_5$

$\mathbf{p}_4$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{P}_5$
SPAM	36(0.288)	36(0.288)	40(0.32)	13(0.104)
MP	22(0.2683)	21(0.2561)	25(0.3049)	14(0.1707)
HP	22(0.2716)	22(0.2716)	27(0.3333)	10(0.1235)
EPP	23(0.2706)	22(0.2588)	30(0.3529)	10(0.1176)
SFBC = SPAM + MD	42(0.2857)	43(0.2925)	48(0.3265)	14(0.0952)
MP + MPD	30(0.2586)	33(0.2845)	36(0.3103)	17(0.1466)
HP + HPD	30(0.2419)	37(0.2984)	44(0.3548)	13(0.1048)
EPP + EPPD	28(0.2642)	26(0.2453)	39(0.3679)	13(0.1226)
ED	4107.3(0.2611)	5064.1(0.3219)	5434.1(0.3454)	1125.4(0.0715)

Table 24.14(b) AHSD and normalized AHSD values between $\mathbf{p}_4$ and $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_5$

$\mathbf{p}_4$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{P}_5$
SPAM	0.1146	0.1146	0.1274	0.0414
MP	0.0701	0.0669	0.0796	0.0446
HP	0.0701	0.0701	0.086	0.0318
EPP	0.0732	0.0701	0.0955	0.0318
SFBC = SPAM + MD	0.089	0.0911	0.1017	0.0297
MP + MPD	0.0636	0.0699	0.0763	0.036
HP + HPD	0.0636	0.0784	0.0932	0.0275
EPP + EPPD	0.0593	0.0551	0.0826	0.0275

Table 24.15(a) HSD and normalized HSD values between $\mathbf{p}_5$ and $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$

$\mathbf{p}_5$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$
SPAM	43(0.2905)	45(0.3041)	47(0.3176)	13(0.0878)
MP	32(0.2807)	33(0.2895)	35(0.307)	14(0.1228)
HP	26(0.2737)	28(0.2947)	31(0.3263)	10(0.1053)
EPP	27(0.2727)	28(0.2828)	34(0.3434)	10(0.101)
SFBC = SPAM + MD	50(0.289)	53(0.3064)	56(0.3237)	14(0.0809)
MP + MPD	43(0.2739)	48(0.3057)	49(0.3121)	17(0.1083)
HP + HPD	37(0.2517)	46(0.3129)	51(0.3469)	13(0.0884)
EPP + EPPD	31(0.2605)	33(0.2773)	42(0.3529)	13(0.1092)
ED	4831.6(0.2736)	5733(0.3247)	5968.7(0.3380)	1125.4(0.0637)

Table 24.15(b) AHSD and normalized AHSD values between $\mathbf{p}_5$ and $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$

$\mathbf{p}_5$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$
SPAM	0.1369	0.1433	0.1497	0.0414
MP	0.1019	0.1051	0.1115	0.0446
HP	0.0828	0.0892	0.0987	0.0318
EPP	0.086	0.0892	0.1083	0.0318
SFBC = SPAM + MD	0.1059	0.1123	0.1186	0.0297
MP + MPD	0.0911	0.1017	0.1038	0.036
HP + HPD	0.0784	0.0975	0.1081	0.0275
EPP + EPPD	0.0657	0.0699	0.089	0.0275

spectral identification. Tables 24.16(a) and (b) – Tables 24.20(a) and (b) tabulate their APDP values with $\mathbf{r} = \mathrm{p}_{13}, \mathrm{p}_{23}, \mathrm{p}_{33}, \mathrm{p}_{43}, \mathrm{p}_{53}$, and $\Delta = \{\mathbf{p}_i\}_{i=1}^5$.

As shown in these tables, binary coding methods using additional L-dimensional binary code words to account for interband spectral deviation would improve better spectral identification such as HP + HPD and the Qian et al.'s method SFBC (SPAM + MD) that produced the lowest APDP values for subpixel identification. In particular, SFBC performed the best for the panels p_{33}, p_{43}, p_{53} that are identified as $\mathbf{p}_3$, $\mathbf{p}_2$, and $\mathbf{p}_2$, respectively. Interestingly, SFBC (SPAM + MD) wrongly identified p_{43}, p_{53} as $\mathbf{p_2}$ that are supposed to be $\mathbf{p}_4$ and $\mathbf{p}_5$. However, if we further applied two spectral measures, spectral angle mapper (SAM), spectral information divergence (SID) to measure the similarity values of p_{13}, p_{23}, p_{33}, p_{43}, p_{53} against $\Delta = \{\mathbf{p}_i\}_{i=1}^5$, the results tabulated in Table 24.21 that revealed a surprising fact that all the five panels p_{13}, p_{23}, p_{33}, p_{43}, p_{53} were shown to be most close to the panel signature $\mathbf{p}_2$. Therefore, by virtue of SAM and SID these five panels would be all identified as $\mathbf{p}_2$, which obviously contradicts the ground truth map in Figure 1.15(b).

The above experiments demonstrated a fact that spectral measures or coding methods carried out on a single pixel basis may not be effective for subpixel discrimination and identification. This made sense because a spectral signature embedded in a single pixel may be mixed by its background signatures and using such a contaminated signature for discrimination and identification may not perform as effectively as laboratory data in computer simulations where the eight binary coding methods performed similarly in most cases. This also explained why SFBC using (3L-2)-length binary code words performed better in real data than other binary coding methods using only (2L-2)-length binary code words because the additional L-length binary code words

Table 24.16(a) APDP/HSD identification of p_{13}

$\mathbf{r} = p_{13}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	0.0889	0.1481	0.1185	0.2963	0.3481
MP	0.1089	0.198	0.1584	0.2277	0.3069
HP	0.0941	0.1882	0.1529	0.2588	0.3059
EPP	0.115	0.1947	0.1593	0.2478	0.2832
SFBC = SPAM + MD	0.0962	0.141	0.109	0.3013	0.3526
MP + MPD	0.1079	0.1583	0.1511	0.2518	0.3309
HP + HPD	0.1278	0.1353	0.0977	0.2932	0.3459
EPP + EPPD	0.1377	0.1856	0.1677	0.2455	0.2635
ED	0.1329	0.1209	0.1534	0.2778	0.3149

Table 24.16(b) APDP/AHSD identification of p_{13}

$\mathbf{r} = p_{13}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	0.0382	0.0637	0.051	0.1274	0.1497
MP	0.035	0.0637	0.051	0.0732	0.0987
HP	0.0255	0.051	0.0414	0.0701	0.0828
EPP	0.0414	0.0701	0.0573	0.0892	0.1019
SFBC = SPAM + MD	0.0318	0.0466	0.036	0.0996	0.1165
MP + MPD	0.0318	0.0466	0.0445	0.0742	0.0975
HP + HPD	0.036	0.0381	0.0275	0.0826	0.0975
EPP + EPPD	0.0487	0.0657	0.0593	0.0869	0.0932

Table 24.17(a) APDP/HSD identification of p_{23}

$\mathbf{r} = p_{23}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	0.1533	0.0867	0.1267	0.30)	0.3333
MP	0.1852	0.1204	0.1574	0.2407	0.2963
HP	0.1889	0.0778	0.1556	0.2778	0.30
EPP	0.1743	0.1101	0.1835	0.2569	0.2752
SFBC = SPAM + MD	0.152	0.0877	0.117	0.3041	0.3392
MP + MPD	0.1656	0.106	0.1523	0.2583	0.3179
HP + HPD	0.1884	0.0652	0.1014	0.3043	0.3406
EPP + EPPD	0.1692	0.1077	0.1769	0.2615	0.2846
ED	0.1312	0.1037	0.1397	0.2941	0.3313

Table 24.17(b) APDP/AHSD identification of p_{23}

$\mathbf{r} = p_{23}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	0.0732	0.0414	0.0605	0.1433	0.1592
MP	0.0637	0.0414	0.0541	0.0828	0.1019
HP	0.0541	0.0223	0.0446	0.0796	0.086
EPP	0.0605	0.0382	0.0637	0.0892	0.0955
SFBC = SPAM + MD	0.0551	0.0318	0.0424	0.1102	0.1229
MP + MPD	0.053	0.0339	0.0487	0.0826	0.1017
HP + HPD	0.0551	0.0191	0.0297	0.089	0.0996
EPP + EPPD	0.0466	0.0297	0.0487	0.072	0.0784

Table 24.18(a) APDP/HSD identification of p_{33}

$\mathbf{r} = \mathrm{p}_{33}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	0.1481	0.1333	0.0889	0.2963	0.3333
MP	0.1961	0.1667	0.1275	0.2157	0.2941
HP	0.1939	0.1531	0.1224	0.2551	0.2755
EPP	0.1969	0.1732	0.1102	0.252	0.2677
SFBC = SPAM + MD	0.1529	0.121	0.0764	0.3057	0.3439
MP + MPD	0.1773	0.1277	0.1206	0.2482	0.3262
HP + HPD	0.1918	0.1164	0.0822	0.2877	0.3219
EPP + EPPD	0.1927	0.1835	0.1239	0.2477	0.2523
ED	0.1475	0.1112	0.1291	0.2907	0.3214

Table 24.18(b) APDP/AHSD identification of p_{33}

$\mathbf{r} = p_{33}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	0.0637	0.0573	0.0382	0.1274	0.1433
MP	0.0637	0.0541	0.0414	0.0701	0.0955
HP	0.0605	0.0478	0.0382	0.0796	0.086
EPP	0.0796	0.0701	0.0446	0.1019	0.1083
SFBC = SPAM + MD	0.0508	0.0403	0.0254	0.1017	0.1144
MP + MPD	0.053	0.0381	0.036	0.0742	0.0975
HP + HPD	0.0593	0.036	0.0254	0.089	0.0996
EPP + EPPD	0.089	0.0847	0.0572	0.1144	0.1165

Table 24.19(a) APDP/HSD identification of p_{43}

$\mathbf{r} = p_{43}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	0.1418	0.097	0.1418	0.2761	0.3433
MP	0.18	0.13	0.17	0.20	0.32
HP	0.202	0.1414	0.1919	0.202	0.2626
EPP	0.1852	0.1389	0.213	0.213	0.25
SFBC = SPAM + MD	0.1382	0.0921	0.1382	0.2829	0.3487
MP + MPD	0.1594	0.1087	0.1594	0.2319	0.3406
HP + HPD	0.1974	0.125	0.1711	0.2237	0.2829
EPP + EPPD	0.1938	0.1318	0.2171	0.2093	0.2481
ED	0.1394	0.1243	0.1645	0.2663	0.3055

Table 24.19(b) APDP/AHSD identification of p_{43}

$\mathbf{r} = p_{43}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	0.1418	0.097	0.1418	0.2761	0.3433
MP	0.18	0.13	0.17	0.20	0.32
HP	0.202	0.1414	0.1919	0.202	0.2626
EPP	0.1852	0.1389	0.213	0.213	0.25
SFBC = SPAM + MD	0.1382	0.0921	0.1382	0.2829	0.3487
MP + MPD	0.1594	0.1087	0.1594	0.2319	0.3406
HP + HPD	0.1974	0.125	0.1711	0.2237	0.2829
EPP + EPPD	0.1938	0.1318	0.2171	0.2093	0.2481

Table 24.20(a) APDP/HSD identification of p_{53}

$\mathbf{r} = p_{53}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	0.1575	0.1164	0.1575	0.2671	0.3014
MP	0.1964	0.1518	0.1875	0.1964	0.2679
HP	0.2162	0.1622	0.2072	0.1982	0.2162
EPP	0.1967	0.1557	0.2213	0.2049	0.2213
SFBC = SPAM + MD	0.1524	0.1098	0.1524	0.2744	0.311
MP + MPD	0.1733	0.1267	0.1733	0.2267	0.30
HP + HPD	0.2118	0.1471	0.1647	0.2235	0.2529
EPP + EPPD	0.2028	0.1469	0.2238	0.2028	0.2238
ED	0.1381	0.1193	0.1591	0.2725	0.3109

Table 24.20(b) APDP/AHSD identification of p_{53}

$\mathbf{r} = p_{43}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SPAM	0.0605	0.0414	0.0605	0.1178	0.1465
MP	0.0573	0.0414	0.0541	0.0637	0.1019
HP	0.0637	0.0446	0.0605	0.0637	0.0828
EPP	0.0637	0.0478	0.0732	0.0732	0.086
SFBC = SPAM + MD	0.0445	0.0297	0.0445	0.0911	0.1123
MP + MPD	0.0466	0.0318	0.0466	0.0678	0.0996
HP + HPD	0.0636	0.0403	0.0551	0.072	0.0911
EPP + EPPD	0.053	0.036	0.0593	0.0572	0.0678

used by SFBC actually show its advantage in capturing spectral variation resulting from real data, but very little from laboratory data.

As a concluding remark, two comments are noteworthy. Despite that two distance measures, HSD and AHSD, were introduced by (24.4) and (24.5), respectively, for the eight binary coding methods, their discrimination and identification results were shown to be the same with only a few cases that different coding methods were selected as the best ones. Another comment is that binary coding methods using Euclidean distance (ED) as distance measure were proved to be less effective in spectral discrimination and identification. None of binary coding methods using ED was shown to be the best method in all the experiments conducted in this chapter.

Table 24.21 Similarity values of $p_{13}, p_{23}, p_{33}, p_{43}, p_{53}$ against $\Delta = \{\mathbf{p}_i\}_{i=1}^{5}$ measured by SAM and SID

$\mathbf{r} = p_{13}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SAM	0.0579	0.0435	0.0740	0.1516	0.1640
SID	0.0068	0.0039	0.0069	0.0338	0.0437

$\mathbf{r} = p_{23}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SAM	0.0660	0.0338	0.0647	0.1643	0.1752
SID	0.0066	0.0019	0.0050	0.0353	0.0445

$\mathbf{r} = p_{33}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SAM	0.1066	0.0684	0.0768	0.2028	0.2131
SID	0.0152	0.0064	0.0072	0.0530	0.0635

$\mathbf{r} = p_{43}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SAM	0.0777	0.0646	0.0963	0.1429	0.1568
SID	0.0069	0.0039	0.0092	0.0275	0.0366

$\mathbf{r} = p_{53}$	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
SAM	0.0748	0.0568	0.0889	0.1470	0.1597
SID	0.0067	0.0031	0.0080	0.0289	0.0375

24.5 Conclusions

Binary coding is a simple and effective technique to be used for spectral characterization. This chapter explores various ways to extend one of most widely used binary coding technique, SPAM, developed by Mazer et al. (1988) as well as a spectral feature-based coding method developed by Qian et al. (1996). Three new binary coding techniques, median partition, halfway partition, and equal probability partition are developed and can be further implemented in conjunction with SPAM and Qian et al.'s method to derive new binary coding methods. In order to effectively conduct performance analysis and evaluation between two coding methods, a new criterion, APDP, is also introduced for performance measure. With the help of the proposed APDP, a set of selected eight binary coding methods is evaluated for comparative study via computer simulations and real hyperspectral image experiments in spectral discrimination and identification. Experimental results demonstrate that each method has its own merit and can be used in different applications.

25

Vector Coding for Hyperspectral Signatures

Spectral signature coding is generally performed by encoding a spectral signature vector across its spectral coverage and using the Hamming distance to measure signature similarity as discussed in Chapter 24. The effectiveness of such a spectral signature coding largely relies on how well the Hamming distance can capture spectral variations that characterize a signature vector. Unfortunately, in most cases, such binary coding does not provide sufficient information for signature analysis due to inability of the Hamming distance in capturing the band-to-band variation, in which case the Hamming distance can be considered a memoryless distance. Therefore, one approach is to extend the Hamming distance to a distance with memory. This chapter introduces two new concepts, referred to as spectral derivative feature coding (SDFC) and spectral feature probabilistic coding (SFPC), for signature coding. SDFC is derived from texture features used in texture classification to dictate gradient changes among adjacent bands in characterizing spectral variations so as to improve spectral discrimination and classification. SFPC implements the well-known arithmetic coding (AC) in two different ways to encode a signature vector in a probabilistic manner, called circular-SFPC (C-SFPC) and split-SFPC (S-SFPC). The values resulting from AC are then used to measure the distance between two signature vectors. The experimental results show that these two signature vector coding methods indeed perform better than binary coding methods discussed in Chapter 24 due to their use of memory in coding at the expense of coding complexity.

25.1 Introduction

Spectral binary coding methods such as SPAM binary coding (1988) and spectral feature binary coding (SFBC) (Qian et al., 1996) presented in Chapter 24 are the simplest ways to characterize a spectral signature vector via a custom-designed binary code book to capture spectral variation across its spectral wavelength coverage. For example, SPAM encodes an L-dimensional signature vector as a $(2L-2)$-dimensional binary code word that is composed of the first L binary values encoded as the sign of the difference between a signature component and its spectral signature mean, and additional $L-2$ binary values encoded as the sign of the difference between a signature component and its corresponding signature components in its adjacent spectral bands within a signature vector. SFBC extends SPAM by including another set of additional L binary values to encode a signature vector as a $(3L-2)$-dimensional binary code word where the new added set of

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

L binary values is used to indicate whether the deviation of a spectral value in a band from the spectral signature mean of a signature vector to be encoded is greater than a threshold that is an average of the absolute differences between the value in each signature components and the spectral signature mean.

As an alternative, SPAM and SFBC can also be considered as encoders that use 1 bit to encode the current change in spectral variation and 1 bit (SPAM) or 2 bits (SFBC) to memorize the spectral changes among adjacent bands. Accordingly, SPAM and SFBC can also be interpreted as 2-bit encoder using 1-bit process unit with 1-bit memory and 3-bit encoder using 1-bit process unit with 2-bit memory, respectively. So, both SPAM and SFBC can be viewed as memory coding methods. By virtue of this interpretation, one way to improve the performance of SPAM and SFBC is to increase the number of bits used for memory. In doing so, it requires a more sophisticated coding structure than binary coding structure used in SPAM and SFBC. This chapter re-invents the wheel by introducing vector coding into the design structure so that drastic changes in spectral variations across the wavelength range can be captured more effectively via a vector of bits instead of binary bits. Here, the vector used in the vector coding indicates that the coding is performed on a block of adjacent bands instead of the scalar coding performed by SPAM and SFBC band-by-band.

The first vector coding method of interest, referred to as spectral derivative feature coding (Chang et al., 2009), was developed to improve both SPAM and SFBC in the sense of signature characterization. It is a spectral texture feature-based coding method and can be considered as a generalization of both SPAM and SFBC by encoding gradient changes in adjacent bands in a signature vector as spectral texture features. The idea is derived from the texture analysis based on a recently developed texture feature coding method (TFCM) (Hong et al., 2002; Hong, 2003) that has shown a promise in medical imaging applications such as liver disease classification in sonograms and mass detection in mammograms. The proposed SDFC converts image-based texture features to spectral derivative features in terms of spectral variations across three adjacent bands. It encodes a signature vector as a two-dimensional vector of bits while keeping track of gradient changes in spectral variation as spectral derivatives in three adjacent bands. In light of this interpretation, both SPAM and SFBC actually perform as 1-bit binary encoders using 1-bit memory and 2-bit memory, respectively.

The second vector coding method is quite different from commonly used techniques designed for spectral signature coding. It is based on the arithmetic coding developed by Rissanen (1976, 1978). Three features arising from AC but not found in binary coding methods are used to design an AC-based coding technique, called spectral feature probabilistic coding, for vector coding (Chang et al., 2010). First, instead of using a bit stream by SPAM and SFBC to encode spectral values of each band, a set of quantized values is used for encoding via a vector of bits. Second, it calculates probabilities of spectral-quantized values to perform variable length coding as opposed to 1-bit coding performed by binary coding with binary values being considered to be equally likely. Finally, it uses the entire range of wavelengths as a block to perform coding compared to 1-bit band-by-band coding such as SPAM and SFBC or a 2-bit 3-band block coding such as SDFC. SFPC encodes a spectral signature vector across the entire range of wavelengths into a real value. Its decoder then decodes the encoded real value precisely to recover the original spectral signature vector. So, unlike SPAM and SFBC that encode a spectral signature vector into a binary code word, the proposed SFPC encodes a spectral signature vector as a real value in (0,1). As a result, the similarity between two spectral signature vectors is measured by the absolute value between their respective AC-encoded real values rather than the Hamming distance commonly used in spectral signature coding. The introduction of AC in spectral signature coding by SFPC has several advantages. First, it can dictate and capture between-band spectral variations in a probabilistic manner where the spectral variations are modeled by a range of probabilities that are determined by the number of bits used in AC. This cannot be accomplished by SPAM and SFBC. Second, SFPC can be developed as an n-bit encoder with n being the number of bits allowed to describe variations in spectral characterization compared to SPAM and SFBC that can only be implemented as 2-bit or 3-bit encoders. This is a significant advantage when subtle spectral variations are required for signature coding. Third, it remedies the drawback of using Hamming distance suffered by SPAM and SFBC, which can only measure the difference between two encoded code words bit-by-bit. However, all these advantages are traded for one disadvantage, that is, its complexity is higher that that of SPAM and SFBC in the sense of algorithmic implementation.

25.2 Spectral Derivative Feature Coding

This section presents a new coding method, referred to as spectral derivative feature coding (SDFC), that extends and improves both SPAM and SFBC in the sense of signature characterization. The idea is derived from texture analysis based on a feature coding method developed by Hong et al. (2002). Instead of characterizing texture features according to spatial correlation among nine pixels in a 3×3 window, SDFC converts image-based texture features to spectral derivative features by looking into spectral variations among three adjacent bands within a signature vector. In doing so, it reinterprets SPAM and SFBC as 1-bit binary encoders using various numbers of bits to store memory as follows.

25.2.1 Re-interpretation of SPAM and SFBC

Using the sample mean of spectral value of all bands in a spectral signature vector $\mathbf{s} = (s_1, s_2, \ldots, s_L)^T$ specified by (23.1), with L being the total number of spectral bands, we can implement SPAM as a 1-bit binary encoder with 1-bit memory to encode $\mathbf{s}$ as follows.

For each lth band, we encode the s_l by the following code word, denoted by s_l^{SPAM}:

$$s_l^{\mathrm{b}} = \begin{cases} 1; & \text{if } r_l \geq \mu \\ 0; & \text{otherwise} \end{cases} \quad \text{for } 1 \leq l \leq L \tag{25.1}$$

and

$$s_l^{\mathrm{SPAM\_m}} = \begin{cases} 1; & \text{if } s_{l+1} \geq s_{l-1} \\ 0; & \text{otherwise} \end{cases} \quad \text{for } 2 \leq l \leq L-1 \tag{25.2}$$

Concatenating s_l^{b} in (25.1) with $s_l^{\mathrm{SPAM\_m}}$ in (25.2) results in a code word for s_l, denoted by s_l^{SPAM}:

$$s_l^{\mathrm{SPAM}} = \left(s_l^{\mathrm{b}}\, s_l^{\mathrm{SPAM\_m}}\right) \tag{25.3}$$

Since (25.2) is not defined for the first and last bands, $s_1^{\mathrm{SPAM}} = s_1^{\mathrm{b}}$ and $s_L^{\mathrm{SPAM}} = s_L^{\mathrm{b}}$. Therefore, the code word for the signature $\mathbf{s}$ is a $(2L-2)$-dimensional binary code word, given by

$$\mathbf{s}^{\mathrm{SPAM}} = \left(s_1^{\mathrm{b}}\, s_2^{\mathrm{b}}\, s_2^{\mathrm{SPAM\_m}} \ldots s_{L-1}^{\mathrm{b}}\, s_{L-1}^{\mathrm{SPAM\_m}}\, s_L^{\mathrm{b}}\right) \tag{25.4}$$

If the order of $s_1^{\mathrm{b}}\, s_2^{\mathrm{b}}\, s_2^{\mathrm{SPAM\_m}} \ldots s_{L-1}^{\mathrm{b}}\, s_{L-1}^{\mathrm{SPAM\_m}}\, s_L^{\mathrm{b}}$ in $\mathbf{s}^{\mathrm{SPAM}}$ in (25.4) is re-arranged as

$$\left(s_1^{\mathrm{b}}\, s_2^{\mathrm{b}} \cdots s_{L-1}^{\mathrm{b}}\, s_L^{\mathrm{b}}\, s_2^{\mathrm{SPAM\_m}} \cdots s_{L-1}^{\mathrm{SPAM\_m}}\right) \tag{25.5}$$

that turns out to be the same as the original SPAM binary code word with the first L binary bits obtained for each band by (24.2) and the follow-up $(L-2)$ bits representing 1-bit memory for each band obtained by (24.3).

Similarly, Qian et al.'s spectral feature-based binary coding that extended SPAM binary coding is as follows. Let MD be the spectral mean deviation defined by

$$\mathrm{MD} = (1/L)\sum\nolimits_{l=1}^{L} |r_l - \mu| \tag{25.6}$$

$$s_l^{\mathrm{SFBC\_m}} = \begin{cases} 0; & \text{if } s_{l+1} < s_{l-1} \text{ and } |s_l - \mu| < \mathrm{MD} \\ 1; & \text{if } s_{l+1} < s_{l-1} \text{ and } |s_l - \mu| \geq \mathrm{MD} \\ 2; & \text{if } s_{l+1} \geq s_{l-1} \text{ and } |s_l - \mu| < \mathrm{MD} \\ 3; & \text{if } s_{l+1} \geq s_{l-1} \text{ and } |s_l - \mu| \geq \mathrm{MD} \end{cases} \tag{25.7}$$

that represents 2-bit memory:

$$s_l^{\mathrm{SFBC}} = \left(s_l^{\mathrm{b}}\, s_l^{\mathrm{SFBC\_m}}\right) \tag{25.8}$$

Following the same treatment given for SPAM, we can obtain a similar binary code word for SFBC as follows:

$$\mathbf{s}^{\mathrm{SFBC}} = \left(s_1^{\mathrm{b}}\, s_2^{\mathrm{b}}\, s_2^{\mathrm{SFBC\_m}} \ldots s_{L-1}^{\mathrm{b}}\, s_{L-1}^{\mathrm{SFBS\_m}}\, s_L^{\mathrm{b}}\right) \tag{25.9}$$

If the order of $s_1^{\mathrm{b}}\, s_2^{\mathrm{b}}\, s_2^{\mathrm{SFBC\_m}} \ldots s_{L-1}^{\mathrm{b}}\, s_{L-1}^{\mathrm{SFBC\_m}}\, s_L^{\mathrm{b}}$ in $\mathbf{s}^{\mathrm{SFBC}}$ in (25.9) is rearranged as

$$\left(s_1^{\mathrm{b}}\, s_2^{\mathrm{b}} \ldots s_{L-1}^{\mathrm{b}}\, s_L^{\mathrm{b}}\, s_2^{\mathrm{SFBC\_m}} \ldots s_{L-1}^{\mathrm{SFBC\_m}}\right) \tag{25.10}$$

that turns out to be the original SFBC binary code word defined by (24.14) with the first L binary bits obtained for each band by (25.2) and the follow-up $(2L - 2)$ bits representing 2-bit memory for each band obtained by (24.3) and (24.13).

25.2.2 Spectral Derivative Feature Coding

The idea of the spectral derivative feature coding can be traced back to the texture feature coding method (Hong et al., 2002) that was developed to capture texture features based on gradient changes in spectral variation of two successive adjacent pixels among these three pixels. More specifically, assume that $\mathbf{s} = (s_1, s_2, \ldots s_L)^T$ is a hyperspectral signature vector, where L is the total number of spectral bands and r_l is the lth spectral band. Also, let Δ be a desired spectral variation tolerance. Four types of successive gradient changes in spectral value can be described as follows:

$$\begin{aligned} &\text{Type 1: if } |s_l - s_{l-1}| \leq \Delta \text{ and } |s_{l+1} - s_l| \leq \Delta \\ &\text{Type 2: if either } |s_l - s_{l-1}| \leq \Delta \text{ and } |s_{l+1} - s_l| > \Delta \\ &\qquad\qquad \text{or } |s_l - s_{l-1}| > \Delta \text{ and } |s_{l+1} - s_l| \leq \Delta \\ &\text{Type 3: if either } s_l - s_{l-1} < \Delta \text{ and } s_{l+1} - s_l < \Delta \\ &\qquad\qquad \text{or } s_l - s_{l-1} > \Delta \text{ and } s_{l+1} - s_l > \Delta \\ &\text{Type 4: if either } s_l - s_{l-1} < \Delta \text{ and } s_{l+1} - s_l > \Delta \\ &\qquad\qquad \text{or } s_l - s_{l-1} > \Delta \text{ and } s_{l+1} - s_l < \Delta \end{aligned} \tag{25.11}$$

According to degrees of successive gradient changes in spectral values among three consecutive bands, Type 1 corresponds to zero-order spectral variation since there is no gradient change in

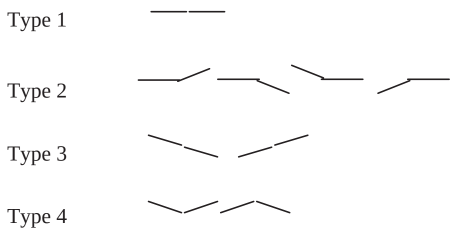


Figure 25.1 Graphic representation of Types 1–4.

spectral value of two successive adjacent bands. Type 2 represents first-order spectral variation since there is one gradient change in spectral value that occurs in only one pair of two adjacent bands. Type 3 and Type 4 describe second-order spectral variation since there are drastic gradient changes in spectral value among three bands. The graphic representations of these four types of spectral value changes are illustrated in Figure 25.1.

For $2 \leq l \leq L-1$

$$s_l^{\mathrm{SDFC\_m}} = \begin{cases} 0 & \text{if } s_l \text{ is Type 1} \\ 1 & \text{if } s_l \text{ is Type 2} \\ 2 & \text{if } s_l \text{ is Type 3} \\ 3 & \text{if } s_l \text{ is Type 4} \end{cases} \tag{25.12}$$

that represents 2-bit memory to keep track of changes in spectral derivatives among the three adjacent bands $l-1$, l, and $l+1$, and the Δ used in (25.11) is set to

$$\Delta = (1/(L-1)) \sum\nolimits_{l=2}^{L} |r_l - r_{l-1}| \tag{25.13}$$

By virtue of (25.1) and (25.13), we can encode the s_l as

$$s_l^{\mathrm{SDFC}} = \left(s_l^{\mathrm{b}}\, s_l^{\mathrm{SDFC\_m}}\right) \tag{25.14}$$

with the signature vector $\mathbf{s}$ encoded as $\mathbf{s}^{\mathrm{SDFC}}$ by

$$\mathbf{s}^{\mathrm{SDFC}} = \left(s_1^{\mathrm{b}}\, s_2^{\mathrm{b}}\, s_2^{\mathrm{SDFC\_m}} \ldots s_{L-1}^{\mathrm{b}}\, s_{L-1}^{\mathrm{SDFC\_m}}\, s_L^{\mathrm{b}}\right) \tag{25.15}$$

Like (25.5) and (25.10), we can also rearrange the order of $s_1^{\mathrm{b}}\, s_2^{\mathrm{b}}\, s_2^{\mathrm{SDFC\_m}} \ldots s_{L-1}^{\mathrm{b}}\, s_{L-1}^{\mathrm{SDFC\_m}}\, s_L^{\mathrm{b}}$ in $\mathbf{s}^{\mathrm{SDFC}}$ in (25.15) to produce a new binary code word

$$\left(s_1^{\mathrm{b}}\, s_2^{\mathrm{b}} \ldots s_{L-1}^{\mathrm{b}}\, s_L^{\mathrm{b}}\, s_2^{\mathrm{SDFC\_m}} \ldots s_{L-1}^{\mathrm{SDFC\_m}}\right) \tag{25.16}$$

that is similar to the original SPAM and SFBC binary code words with the first L binary bits obtained for each band by (25.1) and the follow-up $(2L-2)$ bits representing 2-bit memory for each band obtained by (25.12). It should be noted that SDFC as detailed above can be implemented by two different performance measures. We can either designate the four spectral texture feature types as a numeric set of {0,1,2,3} or express them as 2-bit binary values {00,01,10,11}. For the former case least squares error (LSE) can be used as a distance measure between different types, while a modified Hamming distance measure is used for the latter case. These two implementations are referred to as ED-SDFC and Hamming-SDFC, respectively.

Assume that $\mathbf{s} = (s_1, s_2, \ldots, s_L)^T$ and $\mathbf{t} = (t_1, t_2, \ldots, t_L)^T$ are two signature vectors with $\mathbf{s}^{\text{SDFC}}$ and $\mathbf{t}^{\text{SDFC}}$ being their respective code words encoded by SDFC.

1. Hamming distance (HD)

$$\text{HD}(\mathbf{s}, \mathbf{t}) = (1/L) \sum\nolimits_{l=1}^{L} |s_l^a - t_l^a| + (1/(L-2)) \sum\nolimits_{l=2}^{L-1} \left| s_l^{\text{SDFC\_b}} - t_l^{\text{SDFC\_b}} \right| \tag{25.17}$$

where $s_l^{\text{SDFC\_b}} \in \{00, 01, 10, 11\}$, $t_l^{\text{SDFC\_b}} \in \{00, 01, 10, 11\}$, and $|s_l^{\text{SDFC\_b}} - t_l^{\text{SDFC\_b}}|$ is the Hamming distance. For example, if $s_l^{\text{SDFC\_b}} = 00$ and $t_l^{\text{SDFC\_b}} = 11$, $|s_l^{\text{SDFC\_b}} - t_l^{\text{SDFC\_b}}| = 2$.

2. Absolute value distance (AVD)

$$\text{AVD}(\mathbf{s}, \mathbf{t}) = (1/L) \sum\nolimits_{l=1}^{L} |s_l^a - t_l^a| + (1/(L-2)) \sum\nolimits_{l=2}^{L-1} \left| s_l^{\text{SDFC\_b}} - t_l^{\text{SDFC\_b}} \right| \tag{25.18}$$

where $s_l^{\text{SDFC\_b}} \in \{0, 1, 2, 3\}$, $t_l^{\text{SDFC\_b}} \in \{0, 1, 2, 3\}$, and $|s_l^{\text{SDFC\_b}} - t_l^{\text{SDFC\_b}}|$ is the absolute value. For example, if $s_l^{\text{SDFC\_b}} = 0$ and $t_l^{\text{SDFC\_b}} = 3$, $|s_l^{\text{SDFC\_b}} - t_l^{\text{SDFC\_b}}| = 3$.

25.2.3 AVIRIS Data Experiments

An AVIRIS (Airborne Visible Infrared Imaging Spectrometer) laboratory data set that consists of five reflectance spectra, alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M) shown in Figure 25.2, was used for experiments and reproduced from Figure 1.9. It is provided by the USGS and available online at website. It allows us to simulate various scenarios to explore many interesting insights into SDFC.

Three sets of experiments were performed to illustrate applications of signature discrimination and classification. It should be noted that there is a significant difference between these two where

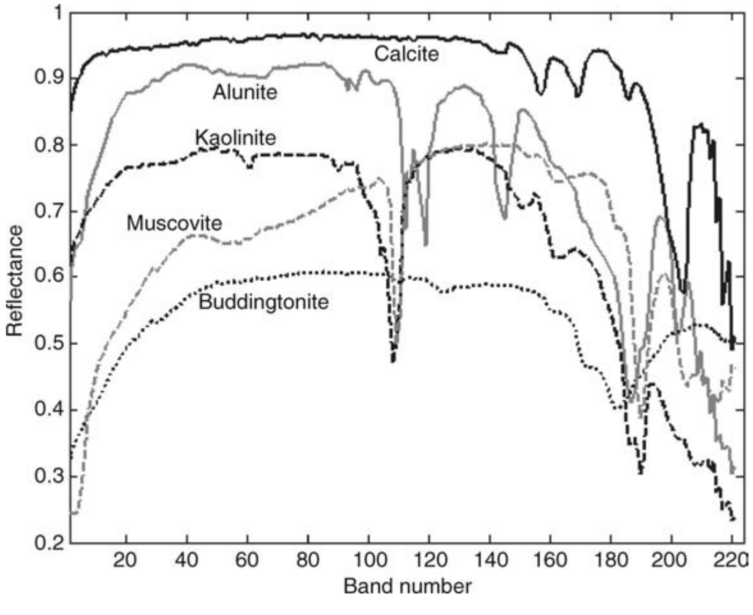


Figure 25.2 Five USGS ground-truth mineral spectra.

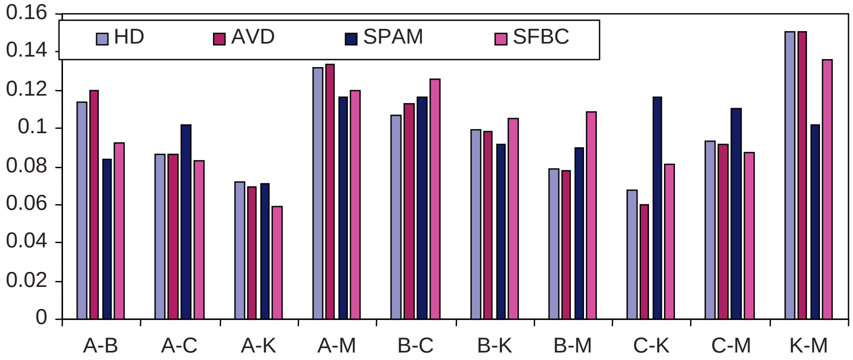


Figure 25.3 Spectral similarity values among A, B, C, K, and M produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC.

discrimination intends to discern one signature from another, but classification attempts to assign a label to a signature.

25.2.3.1 Signature Discrimination

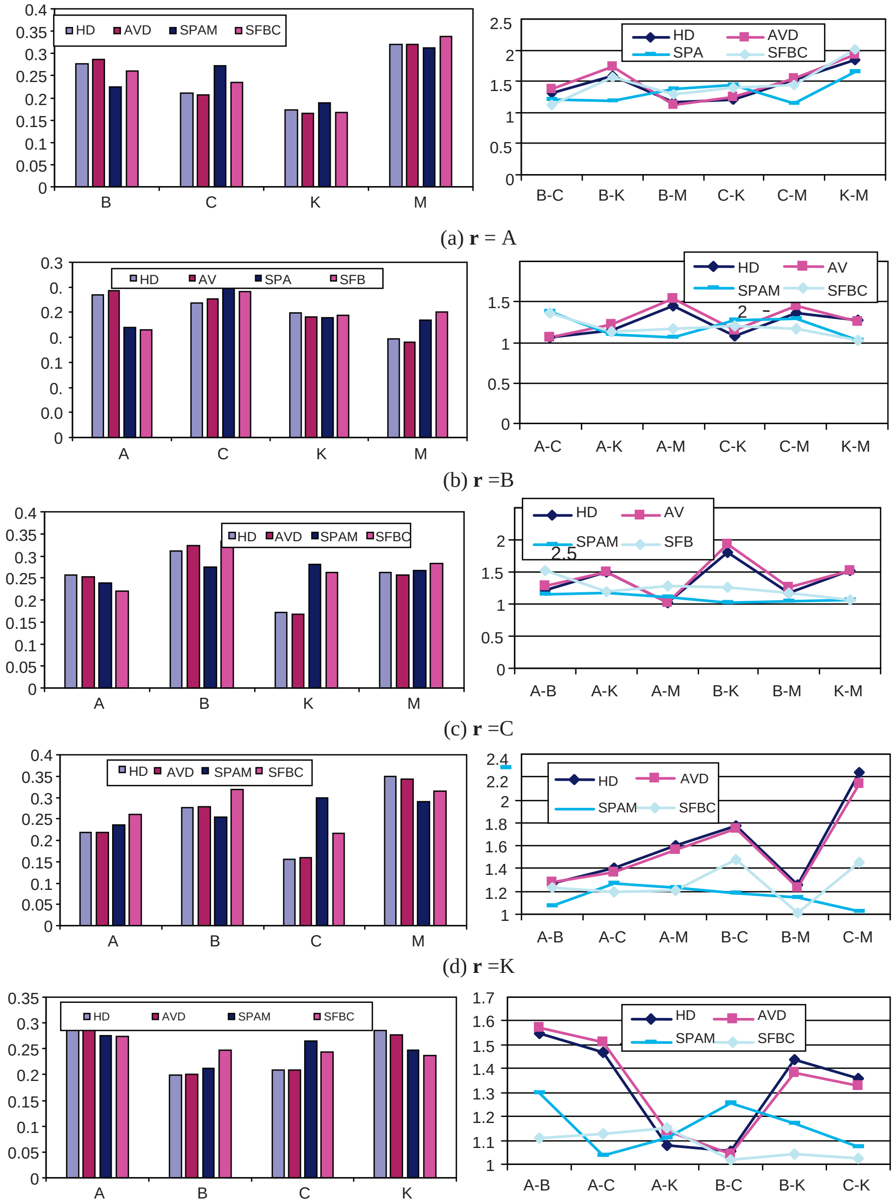
In this section, a comparative study and analysis was conducted for discrimination among the five mineral signature vectors in Figure 25.2, alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M), using four coding schemes, SPAM, SFBC, HD-SDFC, and AVD-SDFC. Figure 25.3 plots the spectral similarity values among the five signature vectors produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC.

In order to better resolve the discrimination power of different algorithms, relative spectral discriminatory power (RSDPW) developed by Chang (2000, 2003a) was used, where RSDPW provided a quantitative measure of spectral discrimination capability of a hyperspectral measure between a pair of spectral signatures relative to a reference signature. It is defined as

$$\mathrm{RSDPW}_m(\mathbf{s}_1, \mathbf{s}_2; \mathbf{r}) = \max\{m(\mathbf{s}_1, \mathbf{r})/m(\mathbf{s}_2, \mathbf{r}), m(\mathbf{s}_2, \mathbf{r})/m(\mathbf{s}_1, \mathbf{r})\} \tag{25.19}$$

where $\mathbf{r}$ is the spectral signature of a reference signature, $\mathbf{s}_1$ and $\mathbf{s}_2$ are the pair of spectral signatures to be measured, and m is the measure applied. As seen from (25.19), $\mathrm{RSDPW}_m(\mathbf{s}_1, \mathbf{s}_2; \mathbf{r})$ selects as the discriminatory power of the measure m, the maximum of the two ratios, ratio of $m(\mathbf{s}_1, \mathbf{r})$ to $m(\mathbf{s}_2, \mathbf{r})$ and ratio of $m(\mathbf{s}_2, \mathbf{r})$ to $m(\mathbf{s}_1, \mathbf{r})$. RSDPW is a symmetric measure and the higher the $\mathrm{RSDPW}_m(\mathbf{s}_1, \mathbf{s}_2; \mathbf{r})$ the better the discriminatory power of measure m. The lower bound of the measure is one that occurs in the case $\mathbf{s}_1 = \mathbf{s}_2$. The signature $\mathbf{s}_1$ in (25.19) can be at designated as the signature of interest while the $\mathbf{s}_2$ is used to compare against the $\mathbf{s}_1$ with respect to the reference signature $\mathbf{r}$. Each of the five signature vectors in Figure 25.2 was selected as a reference signature vector $\mathbf{r}$ that was also set to the signature vector of interest $\mathbf{s}_1 = \mathbf{r}$. Figure 25.4(a)–(e) shows comparative graphic plots of spectral similarity values for reference signature vector $\mathbf{r}$ = A, B, C, K, and M produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC along with plots of their corresponding RSDPW values.

The above results show better discrimination performance of SDFC measures than the two existing coding methods, SPAM and SFBC.



(a) **r** = A

(b) **r** =B

(c) **r** =C

(d) **r** =K

(e) **r** =M

Figure 25.4 Comparative plots of RSDPW values produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC.

25.2.3.2 Mixed Signature Classification

In this section, a mixed signature vector was generated by mixing equal proportions of the five signature vectors in Figure 25.2 given by

$$\mathbf{t}_{\text{mix}} = 1/4(\mathbf{s}_A + \mathbf{s}_B + \mathbf{s}_C + \mathbf{s}_K + \mathbf{s}_M) \quad (25.20)$$

and shown in Figure 25.5 for comparison.

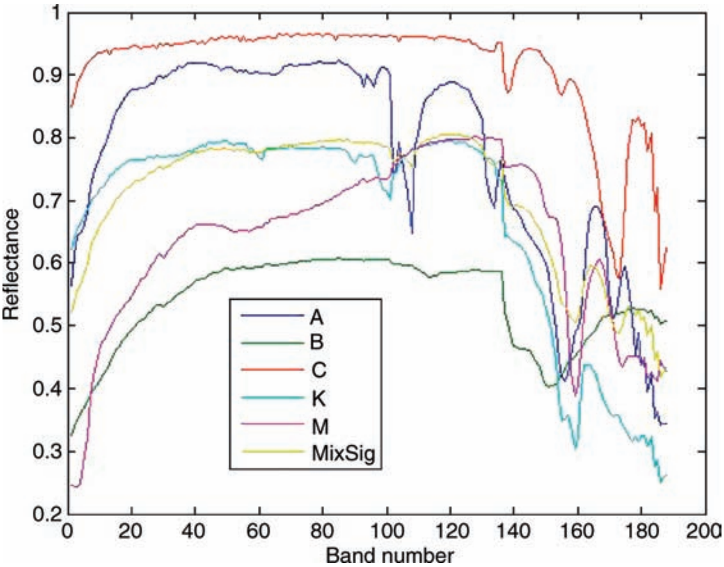


Figure 25.5. Five USGS ground-truth mineral spectra together with the mixed signature.

From the spectral plot of the mixed signature vector shown in Figure 25.5 we notice that spectral signature of kaolinite lied in between the spectral signatures of the other four signature vectors in the set and thus should be closest to the mixed signature vector $\mathbf{t}_{\text{mix}}$. Thus, it can be expected that the mixed signature vector would be classified as kaolinite. The results of Table 25.1 verified this assessment and further showed that SDFC-based spectral measures were able to perform this classification correctly.

Figure 25.6 shows comparative graphic plots of their spectral similarity values of A, B, C, K, and M between $\mathbf{t}_{\text{mix}}$ produced by different measures in Table 25.1.

For the purpose of studying discrimination among different measures, RSDPW is calculated with the selection of reference as the average of the five signature vectors, $\mathbf{t}_{\text{mix}}$. K was assumed to be the signature vector of interest $\mathbf{s}_1$ and other signature vectors are chosen to be $\mathbf{s}_2$ to be compared against $\mathbf{s}_1$. The plots of their RSDPW values are presented in Figure 25.7.

25.2.4 NIST Gas Data Experiments

A second data set used for experiments was spectral signatures of gas agent data shown in Figure 1.10 and discussed in Section 1.6.3. It is provided by the National Institute of Standards and Technology (NIST) and is available on website (webbook.nist.gov/chemistry). The frequency range of $\mathbf{s}_1$ is 550–3846 cm^{-1} acquired by 825 bands, while that of $\{\mathbf{s}_i\}_{i=2}^{9}$ is 450–3966 cm^{-1} acquired by 825 bands, giving each signature a spectral resolution of about 4 cm^{-1} per band. Since

Table 25.1 Classification results of simulation AVIRIS data

Distance measure w.r.t. $\mathbf{t}_{\text{mix}}$	A	B	C	K	M
Hamming	0.16239	0.19658	0.25356	0.1567	0.2307
(SPAM/SFBC)	0.17492	0.28218	0.19142	0.12376	0.2277
HD-SDFC	0.20332	0.29054	0.23205	0.11954	0.2445
AVD-SDFC	0.19245	0.30144	0.22361	0.11158	0.2409

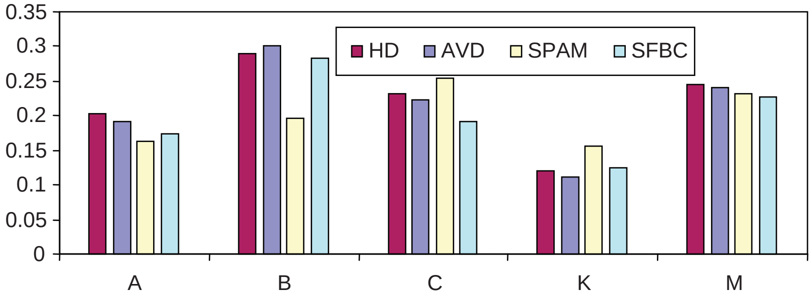


Figure 25.6 Comparative plot of spectral similarity values of the simulation data produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC.

the $\mathbf{s}_1$ is acquired by 825 bands compared to the other eight signature vectors $\{\mathbf{s}_i\}_{i=2}^{9}$ that are collected by 880 bands, only the eight signature vectors $\{\mathbf{s}_i\}_{i=2}^{9}$ are used for our experiments.

25.2.4.1 Signature Discrimination

Once again, we first evaluated effectiveness of SPAM, SFBC, HD-SDFC, and AVD-SDFC in spectral similarity. Figure 25.8 plots histograms of the spectral similarity values among the eight signatures $\{\mathbf{s}_i\}_{i=2}^{9}$ produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC, where there were $\binom{28}{2} = \frac{8\cdot 7}{2} = 28$ pairs of gas agents to be measured for spectral similarity by the four methods, SPAM, SFBC, HD-SDFC, and AVD-SDFC.

Obviously, from Figure 25.8 it difficult to see how a coding method performs in signature discrimination. So, in order to compare the discriminatory powers of these four different coding methods, SPAM, SFBC, HD-SDFC, and AVD-SDFC, RSDPW in (25.19) is also used as a measure of discriminatory power produced by a coding method. The spectral signature $\mathbf{s}_9$ was chosen as the signature vector of interest $\mathbf{s}_1$ in (25.19) and other signature vectors $\{\mathbf{s}_i\}_{i=2}^{8}$ were used as $\mathbf{s}_2$ in (25.19) to be compared against $\mathbf{s}_9$. The selection of $\mathbf{s}_9$ was arbitrary and any other signature vector could also be used for the same purpose. Table 25.4 tabulates the RSDPW values among $\{\mathbf{s}_i\}_{i=2}^{8}$ using $\mathbf{r} = \mathbf{s}_9$ as the reference signature vector produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC and Figure 25.9 plots their corresponding RSDPW values for visual assessment.

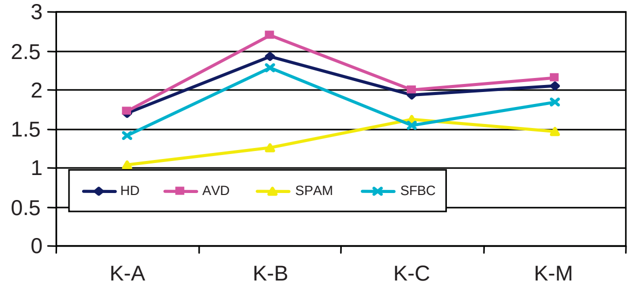


Figure 25.7 Comparative plots of RSDPW values of Figure 25.6 produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC with K designated as the desired signature.

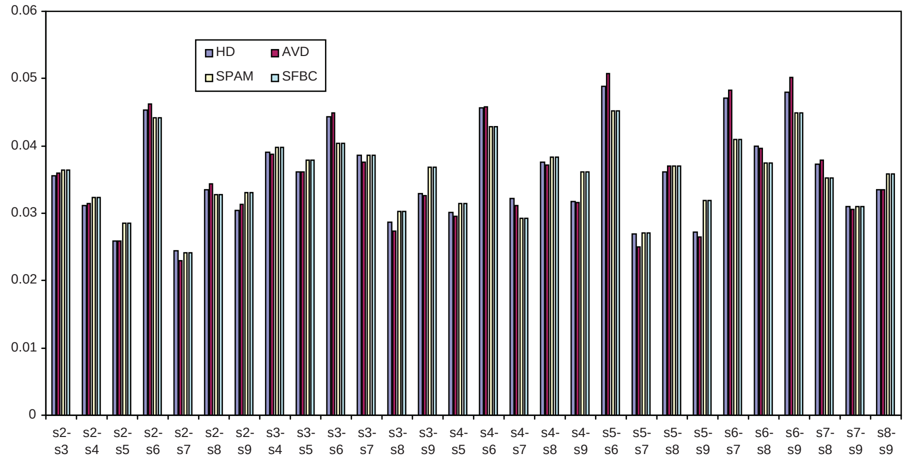


Figure 25.8 Spectral similarity values among $\{\mathbf{s}_i\}_{i=2}^{9}$ produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC.

According to the definition of (25.19), a measure is effective if the more spectrally distinct the two signatures, the greater their RSDPW values, and conversely, the more similar the two signatures, the smaller their RSDPW values. With this interpretation, both versions of the SFDC apparently performed better than SPAM and SFBC.

25.2.4.2 Mixed Signature Classification

This section performed classification of mixed signature vectors with $\mathbf{G}_{\mathbf{mix}}$ chosen to be the average of the eight gas signatures, $\{\mathbf{s}_i\}_{i=2}^{9}$. Due to nearly a thousand of spectral bands used to acquire the gas data, there is a wide range of spectral variations of the gas data as shown in Figure 1.10. To address this issue, the value of the Δ in (25.13) used for signature analysis must be adapted to sensitivity of gradient changes in spectral variation. While SDFC worked by capturing adjacent band gradient variation, a large amount of local variation may acted like noise, and further reduce its key features of a spectral signature. One way to address this issue was to use multiples of Δ in (25.13) as thresholds. This is similar to low-pass filtering of the data. The value of a new threshold

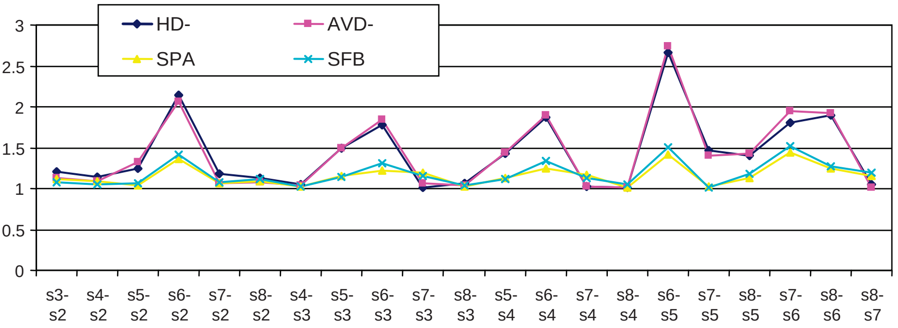


Figure 25.9 Comparative plots of RSDPW values produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC using $\mathbf{r} = \mathbf{s}_9$.

Table 25.2 Different thresholds for the signatures $\{\mathbf{s}_i\}_{i=2}^{9}$ obtained for Δ, 5Δ, 10Δ, and 20Δ

	Threshold = Δ	Threshold = 5Δ	Threshold = 10Δ	Threshold = 20Δ
$\mathbf{s}_2$	44.19	220.95	441.9	883.8
$\mathbf{s}_3$	12,765	63,825	127,650	255,300
$\mathbf{s}_4$	60.74	303.7	607.4	1214.8
$\mathbf{s}_5$	43.47	217.35	434.7	869.4
$\mathbf{s}_6$	616.86	3084.3	6168.6	12337.2
$\mathbf{s}_7$	70.81	354.05	708.1	1416.2
$\mathbf{s}_8$	48.95	244.75	489.5	979
$\mathbf{s}_9$	77.74	388.7	777.4	1554.8

should be chosen such that much of the noise was eliminated without sacrificing the key features of the signature vector. Table 25.2 shows the different values of thresholds for the signature vectors $\{\mathbf{s}_i\}_{i=2}^{9}$ obtained for Δ, 5Δ, 10Δ, and 20Δ.

Figure 25.10 shows the RSDPW values for the two SDFC-based measures (HD and AVD), obtained by Δ, 5Δ, 10Δ, and 20Δ.

As seen from Figure 25.10, 10Δ seemed to provide the best discrimination results while at higher values of Δ (20Δ or more) the distinction among signatures is not preserved. The classification results for the gas data are presented in Table 25.3 for two values Δ and 10Δ for the two SDFC measures together with SAM and SID as reference.

From Table 25.3 we saw that for 10Δ, we seemed to get a better discrimination among the spectral signatures. Also we noticed that signature vectors $\mathbf{s}_6$, $\mathbf{s}_7$, and $\mathbf{s}_8$ gave similar results when compared to mix signature vector ($\mathbf{G}_{\mathbf{mix}}$). Figure 25.11 plots RSDPW values of Table 25.3, where reference signature vector is chosen as the average of the eight signature vectors $\{\mathbf{s}_i\}_{i=2}^{9}$ and signature vector $\mathbf{s}_9$ is chosen as the signature vector of interest $\mathbf{s}_1$, as mentioned in (25.19).

For a better visual inspection of signature vectors $\mathbf{s}_6$, $\mathbf{s}_7$, $\mathbf{s}_8$, and $\mathbf{G}_{\mathbf{mix}}$ for discrimination, Figure 25.12 shows their spectral profiles for comparison.

Based on the above classification results, we see that signature vectors $\mathbf{s}_6$, $\mathbf{s}_7$, and $\mathbf{s}_8$ were very similar, while signature vector $\mathbf{s}_9$ was most distinct. So, we formed a subset of signature vectors $\mathbf{s}_6$, $\mathbf{s}_7$, $\mathbf{s}_8$, and $\mathbf{s}_9$ to perform discrimination among each other. Each of the above four signature vectors was selected as a reference signature vector $\mathbf{r}$ that is also set to the signature vector of interest $\mathbf{s}_1 = \mathbf{r}$. Figure 25.13(a)–(d) shows comparative graphic plots of spectral

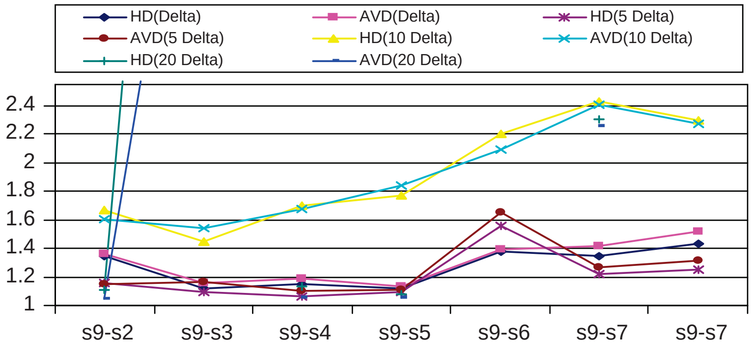


Figure 25.10 RSDPW values produced for Δ, 5Δ, 10Δ, and 20Δ.

Table 25.3 Classification results of gas data

Distance measure w.r.t. $\mathbf{G}_{\text{mix}}$	$\mathbf{s}_2$	$\mathbf{s}_3$	$\mathbf{s}_4$	$\mathbf{s}_5$	$\mathbf{s}_6$	$\mathbf{s}_7$	$\mathbf{s}_8$	$\mathbf{s}_9$
SPAM	0.12202	0.1281	0.14295	0.13305	0.10626	0.10333	0.11481	0.14948
SFBC	0.12439	0.12245	0.13718	0.13301	0.12161	0.10618	0.11341	0.14177
SAM	0.13532	0.1215	0.128	0.1338	0.12767	0.085297	0.12542	0.14299
SID	0.13023	0.10697	0.12353	0.14853	0.092757	0.065252	0.088639	0.2691
HD-SDFC (Δ)	0.1141	0.13771	0.13379	0.13732	0.11138	0.11456	0.10737	0.15377
AVD-SDFC (Δ)	0.11595	0.13599	0.13246	0.1395	0.11309	0.11128	0.10402	0.15771
HD-SDFC (10Δ)	0.12652	0.14537	0.12402	0.119	0.095784	0.086657	0.091677	0.21097
AVD-SDFC (10Δ)	0.13015	0.13527	0.12451	0.1133	0.099838	0.086563	0.091653	0.20872

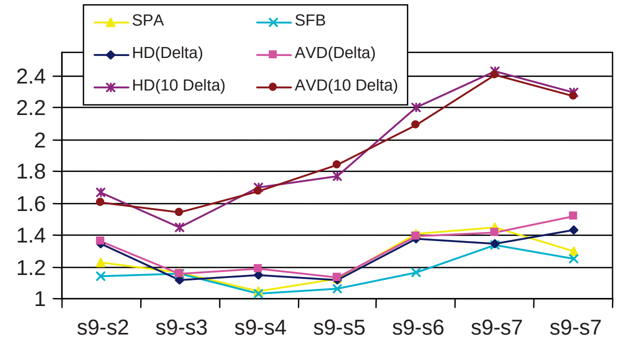


Figure 25.11 RSDPW values of the chemical/biological data classification result (where reference signature vector was $\mathbf{G}_{\text{mix}}$ and signature vector of interest was $\mathbf{s}_9$) produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC (for Δ and 10Δ).

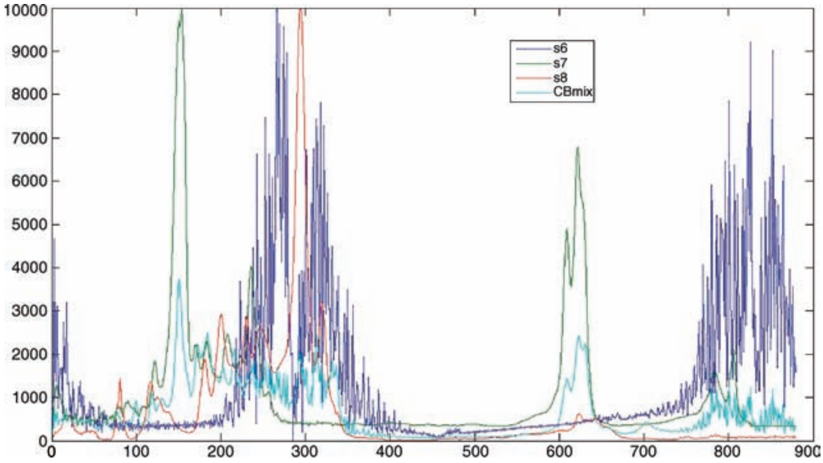
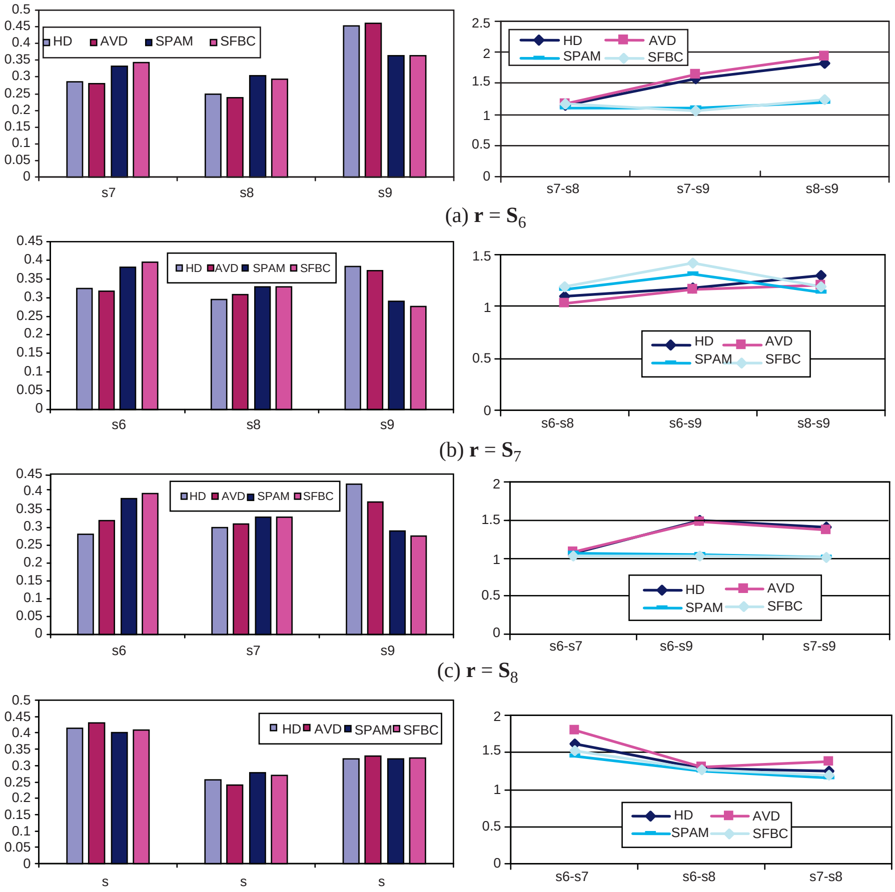


Figure 25.12 Signature vectors $\mathbf{s}_6$, $\mathbf{s}_7$, and $\mathbf{s}_8$ and average signature vector of the gas data set $\{\mathbf{s}_i\}_{i=2}^{9}$, $\mathbf{G}_{\text{mix}}$.



(a) $\mathbf{r} = \mathbf{S}_6$

(b) $\mathbf{r} = \mathbf{S}_7$

(c) $\mathbf{r} = \mathbf{S}_8$

(d) $\mathbf{r} = \mathbf{S}_9$

Figure 25.13 Comparative plots of spectral similarity values of the panel data produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC along with the plot of RSDPW values.

similarity values for reference signature vector $\mathbf{r} = \mathbf{s}_6$, $\mathbf{s}_7$, $\mathbf{s}_8$, and $\mathbf{s}_9$ produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC (for 10Δ) along with plots of their corresponding RSDPW values.

Two remarks on SDFC are worth noting:

1. SDFC-based measures generally performed better than SPAM and SFBC. Since SDFC-based measures (HD-SDFC and AVD-SDFC) caught gradient changes in spectral variation across bands, they worked more effectively when there was a fair amount of gradient variations in a spectral signature.
2. Among the two SDFC-based measures detailed in this chapter, AVD-SDFC seemed to give a better performance in general. This might be due to the fact that AVD-SDFC was able to pick up gradient changes more accurately. Nevertheless, their performances were very similar though, because they shared the same inherent idea.

For more complicated data sets such as the gas data used above with a high amount of local gradient variations increasing the threshold parameter, Δ of SDFC in (25.13) can achieve better discrimination, but a care should be taken to ensure that key features of the signature vector are not lost due to excessively high values of a threshold.

25.3 Spectral Feature Probabilistic Coding

The SDFC developed in Section 25.2 still follows a similar design philosophy used for SPAM and SFBC developed in Chapter 24. The spectral feature probabilistic coding presented in this section takes a completely different route to design codes for a signature vector. It replaces 1-bit threshold used by binary coding with a set of discrete values obtained from quantizing real spectral values. So, the number of bits to be used is determined by how fine the discrete values are desired to be used for encoding. Furthermore, SFPC uses the entire number of spectral bands as a block of memory to keep track of changes in the complete spectral profile of a signature vector as opposed to binary coding that uses only two or three adjacent bands as blocks of memory to capture changes in a very limited spectral range. As expected, SFPC can more accurately describe spectral characteristics of a signature vector than binary coding.

25.3.1 Arithmetic Coding

Since arithmetic coding is a well-established coding method, we refer to Rissanen (1976) and Langdon and Rissanen (1981) for details. This section briefly reviews the underlying concept of AC. In doing so, the best way is to use a simple example to illustrate how AC works. Suppose that X is a binary information source where $\{0,1\}$ is the source alphabet space and probabilities of 0 and 1 given by 0.4 and 0.6, respectively. Assume that a source message S is a binary string specified by $S = 01101$. The key idea of AC is to break up the unit interval [0,1) in accordance with probabilities assigned to each source alphabet. In our example, the interval of [0,1) is divided into [0,0.4) corresponding to the probability of 0, 0.4 and [0.4,1) corresponding to the probability of 1, 0.6. After seeing the first bit "0" of the S, we know the string S must lie in between 0 and 0.4 as shown in Figure 25.14(a). Then the interval [0,0.4) is further divided into two subintervals, [0,0.16) corresponding to 0 and [0.16,0.4) corresponding to 1. After seeing the second bit "1", the interval in which the string S lies is reduced to [0.16,0.4) with an increased decimal precision as shown in Figure 25.14(b). In order to encode the third bit in the string, the interval [0.16,0.4) is further divided into two subintervals [0.16,0.256) corresponding to 0 and [0.256,0.4) corresponding to 1 according to the probability of 0, 0.4 and the probability of 1, 0.6. The same procedure is continued until the end of the string S as shown in Figure 25.14(c)–(e).

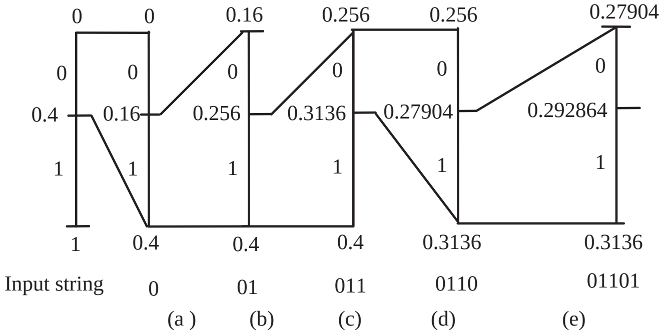


Figure 25.14 Illustrative diagram of encoding a binary string 01101 using AC.

In other words, the binary string $S = 01101$ can be encoded by a real value lying in an interval continuously broken up in accordance with the probabilities assigned to 0 and 1, which are 0.4 and 0.6. The length of the interval within which the string S lies is shrunk by one decimal precision every time a new bit is input from the string S. This can be illustrated as follows using Figure 25.14 as an example.

1. After seeing "0", $S \in [0, 0.4)$ in Figure 25.2(a).
2. After seeing "01", $S \in [0.16, 0.4)$ in Figure 25.2(b).
3. After seeing "011", $S \in [0.256, 0.4)$ in Figure 25.2(c).
4. After seeing "0110", $S \in [0.256, 0.3136)$ in Figure 25.2(d).
5. After seeing "01101", $S \in [0.27904, 0.3136)$ in Figure 25.2(e).

So, from the above step-by-step illustration the accuracy of encoding the string S is improved by one decimal precision via a shrinking process of the interval length by one-tenth. This implies that AC remembers all its past while encoding the string S. The more the past stored in its memory, the better the accuracy with which it encodes strings produced by the information source X. It should be noted that the example provided in this section only serves as an illustrative purpose. More sophisticated examples and its programming implementation can be found in Rissanen (1976), Welch (1984), Witten et al. (1987), Bell et al. (1990), and Gersho and Gray (1992).

25.3.2 Spectral Feature Probabilistic Coding

Next we develop an AC-based spectral feature probabilistic coding that implements AC to encode a signature vector, denoted by $\mathbf{s} = (s_1, s_2, \ldots, s_L)^T$ with the total number of spectral bands specified by L. Assume that AC-encoded code word of the signature vector $\mathbf{s}$ is denoted by $\mathbf{s}^{\mathrm{AC}}$, which is actually a real value between 0 and 1. SFPC can be implemented for each signature vector $\mathbf{s}_i = (s_{i1}, s_{i2}, \ldots, s_{iL})^T$ of $\{\mathbf{s}_i\}_{i=1}^N$ as follows.

SFPC algorithm for encoding $\mathbf{s}_i$

1. Determine the number of quantization levels q to be used for AC. This is equivalent to determining the number of bits b to be used for AC with $q = 2^b$.
2. Implement an optimal uniform q-level quantizer (Lloyd, 1982; Gersho and Gray, 1992) to quantize L components $\{s_{i1}, s_{i2}, \ldots, s_{iL}\}$ of the signature vector $\mathbf{s}_i$ into q levels, denoted by $a_1^{\mathbf{s}_i}, a_2^{\mathbf{s}_i}, \ldots, a_q^{\mathbf{s}_i}$ that is considered as a source code word space of the signature vector $\mathbf{s}_i$.
3. Find a histogram of the q quantization levels $a_1^{\mathbf{s}_i}, a_2^{\mathbf{s}_i}, \ldots, a_q^{\mathbf{s}_i}$ to produce the probabilities of $a_1^{\mathbf{s}_i}, a_2^{\mathbf{s}_i}, \ldots, a_q^{\mathbf{s}_i}$ for the signature vector $\mathbf{s}_i$ denoted by $p_1^{\mathbf{s}_i}, p_2^{\mathbf{s}_i}, \ldots, p_q^{\mathbf{s}_i}$ that will be used for AC.
4. Use AC described in Section 25.2.2 to encode $\mathbf{s}_i$ with source alphabet probabilities obtained in step 3, $p_1^{\mathbf{s}_i}, p_2^{\mathbf{s}_i}, \ldots, p_q^{\mathbf{s}_i}$.
5. Assume that the final interval at which AC is terminated is $[x, y]$. The value at the half way between x and y, that is, $(x + y)/2$, is then assigned to the signature vector $\mathbf{s}_i$ as its code word denoted by $\mathbf{s}_i^{\mathrm{AC}}$ that is a real value in [0,1).

In order for SFPC to perform spectral similarity, let two spectral signature vectors be given by $\mathbf{s}_1 = (s_{11}, s_{12}, \ldots, s_{1L})^T$ and $\mathbf{s}_2 = (s_{21}, s_{22}, \ldots, s_{2L})^T$ with their SFPC-encoded code words $\mathbf{s}_1^{\mathrm{AC}}$ and $\mathbf{s}_2^{\mathrm{AC}}$. An SFPC measure between these two signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ can be defined by

$$\mathrm{SFPC}(\mathbf{s}_1, \mathbf{s}_2) = |\mathbf{s}_1^{\mathrm{AC}} - \mathbf{s}_2^{\mathrm{AC}}| \tag{25.21}$$

that is the absolute difference between their two SFPC-encoded code words $\mathbf{s}_1^{\mathrm{AC}}$ and $\mathbf{s}_2^{\mathrm{AC}}$. Since AC is a memory coding method, its initial condition has significant impact on its performance. Therefore, the initial band to be encoded by AC plays the key role in SFPC measure defined by (25.21). To address this issue, the initial band is selected by the band denoted by l^* that yields the maximum spectral variation between two signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ over all spectral bands and defined by

$$l^* = \arg\{\max_l |\mathbf{s}_{1l} - \mathbf{s}_{2l}|\} \tag{25.22}$$

With the l^*th band as the initial band, AC encodes both signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ in two different ways. One is called circular-SFPC that starts off the first band, the l^*th band, followed by the (l^*+1)th band until it reaches the last band, Lth band, and then it goes back to the first band, the second band, and stops its encoding at the (l^*-1)th band. In other words, AC implemented in C-SFPC is performed in the following circular order:

$$l^*, l^*+1, l^*+2, \ldots, L-1, L, 1, 2, \ldots, l^*-2, l^*-1 \tag{25.23}$$

Another is called split-SFPC that uses the l^*th band to break up the entire bands into two band subsets, $\Omega_1 = \{1, 2, \ldots, l^*-1, l^*\}$ and $\Omega_2 = \{l^*, l^*+1, \ldots, L\}$. Then AC implemented in S-SFPC encodes both signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ in Ω_1 in the decreasing order of band numbers:

$$l^*, l^*-1, l^*-2, \ldots, 1 \tag{25.24}$$

with their respective AC-encoded words denoted by $\mathbf{s}_1^{\mathrm{AC},\Omega_1}$ and $\mathbf{s}_2^{\mathrm{AC},\Omega_1}$ and Ω_2 in the increasing order of band numbers

$$l^*, l^*+1, l^*+2, \ldots, L-1, L \tag{25.25}$$

with their respective AC-encoded words denoted by $\mathbf{s}_1^{\mathrm{AC},\Omega_2}$ and $\mathbf{s}_1^{\mathrm{AC},\Omega_2}$. Then the spectral similarity between $\mathbf{s}_1$ and $\mathbf{s}_2$ is measured by

$$\mathrm{SFPC}(\mathbf{s}_1, \mathbf{s}_2) = |\mathbf{s}_1^{\mathrm{AC},\Omega_1} - \mathbf{s}_2^{\mathrm{AC},\Omega_1}| + |\mathbf{s}_1^{\mathrm{AC},\Omega_2} - \mathbf{s}_2^{\mathrm{AC},\Omega_2}| \tag{25.26}$$

In what follows, we summarize how to perform SFPC as a discrimination measure according to two different ways described by (25.23) and (25.24) and (25.25).

SFPC measure for discrimination

1. Find the initial band l^* determined by (25.22) for any given two signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$.
2. Implement C-SFPC using (25.23) or S-SFPC by (25.24) and (25.25) to encode both signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$.
3. Measure the spectral similarity between $\mathbf{s}_1$ and $\mathbf{s}_2$ via (25.21) or (25.26).

Three remarks are worth noting:

1. The selection of the initial band by (25.22) has two significant advantages. First, it makes the SFPC invariant to the ordering of the spectral bands. Second, it avoids a dilemma that the performance SFPC can be affected by a drastic change between two adjacent spectral bands.
2. There is an alternative to calculate probabilities of $a_1^{\mathbf{s}_i}, a_2^{\mathbf{s}_i}, \ldots, a_q^{\mathbf{s}_i}$. If two signature vectors $\mathbf{s}_1 = (s_{11}, s_{12}, \ldots, s_{1L})^T$ and $\mathbf{s}_2 = (s_{21}, s_{22}, \ldots, s_{2L})^T$ are compared for discrimination, another way to calculate the probabilities of the q optimal levels $\{a_1, a_2, \ldots, a_q\}$ is to consider the $2L$ components of the two signature vectors, $\{s_{11}, s_{12}, \ldots, s_{1L}, s_{21}, s_{22}, \ldots, s_{2L}\}$, as the input of the q-level uniform optimal quantizer to generate a common probability distribution $p_1^{(\mathbf{s}_1,\mathbf{s}_2)}, p_2^{(\mathbf{s}_1,\mathbf{s}_2)}, \ldots, p_q^{(\mathbf{s}_1,\mathbf{s}_2)}$ for AC to encode both signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ instead of two individual probability distributions $p_1^{\mathbf{s}_1}, p_2^{\mathbf{s}_1}, \ldots, p_q^{\mathbf{s}_1}$ and $p_1^{\mathbf{s}_2}, p_2^{\mathbf{s}_2}, \ldots, p_q^{\mathbf{s}_2}$. However, according to our

experiments, there is no appreciable difference between these two. In this case, only the latter is used for experiments. It is our belief that the total number of bands L for data acquisition is sufficiently large for AC to capture spectral characterization.

3. Similarly, if a database $\Delta = \left\{\mathbf{s}_i = (s_{i1}, s_{i2}, \ldots, s_{iL})^T\right\}_{i=1}^{N}$, which consists of N signature vectors, is used for signature identification, the probability distribution of the q optimal levels $\left\{a_1, a_2, \ldots, a_q\right\}$ can also be calculated based on quantization of the NL components of the N signature vectors, $\{s_{11}, s_{12}, \ldots, s_{1L}, s_{21}, s_{22}, \ldots, s_{2L}, \ldots, s_{N1}, \ldots, s_{NL}\}$, into the q optimal levels $\left\{a_1, a_2, \ldots, a_q\right\}$ to generate a common probability distribution $p_1^{\Delta}, p_2^{\Delta}, \ldots, p_q^{\Delta}$ for the database that can be used by AC to encode all the N signature vectors $\Delta = \{\mathbf{s}_i\}_{i=1}^{N}$. As noted previously, this practice does not provide particular advantages in improving signature identification.

25.3.3 AVIRIS Data Experiments

The same AVIRIS laboratory data set that consists of five reflectance spectra, alunite (A), buddingtonite (B), calcite (C), kaolinite (K), and muscovite (M) shown in Figure 25.2, is used for experiments to explore many interesting insights into SPFC.

In order to compare 2-bit SPAM and 3-bit SFBC, SFPC algorithm was implemented with AC as 2-bit and 3-bit encoders where Max's uniform quantization (Max, 1960), also known as Lloyd's algorithm I (Lloyd, 1982), was used to quantize the spectral variations of the five spectral signatures. Figure 25.15 shows comparative graphical plots of spectral similarity values among A, B, C, K, and M produced by SPAM, SFBC, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit C-SFPC, and 3-bit S-SFPC. As seen from Figure 25.15, the best discrimination result was generated by 3-bit S-SFPC followed by 3-bit C-SFPC. The histograms are plotted such that for each measure the sums of distance have been normalized to unity.

To quantitatively measure discrimination powers of different coding methods, once again RSDPW$_m$($\mathbf{s}_1$,$\mathbf{s}_2$;$\mathbf{r}$) specified by (25.19) was used for quantitative assessment where the signature vector $\mathbf{s}_1$ in (25.19) was designated as the signature vector of interest while $\mathbf{s}_2$ was used to compare against $\mathbf{s}_1$ with respect to the reference signature vector $\mathbf{r}$. Apparently, the higher the RSDPW$_m$($\mathbf{s}_1$,$\mathbf{s}_2$;$\mathbf{r}$), the better the discriminatory power of measure m. Generally, there are two ways to select the reference signature vector r. One way is to select the average of the five mineral signature vectors in Figure 25.2, that is, designate signature vector $\mathbf{s}_1 = \mathbf{r}$. The other way is to select one of these five signature vectors in Figure 25.2 as a candidate for the reference signature vector. Figure 25.16(a)–(e) shows comparative graphic plots of spectral similarity values for reference signature vector $\mathbf{r} = $ A, B, C, K, and M produced by SPAM and SFBC.

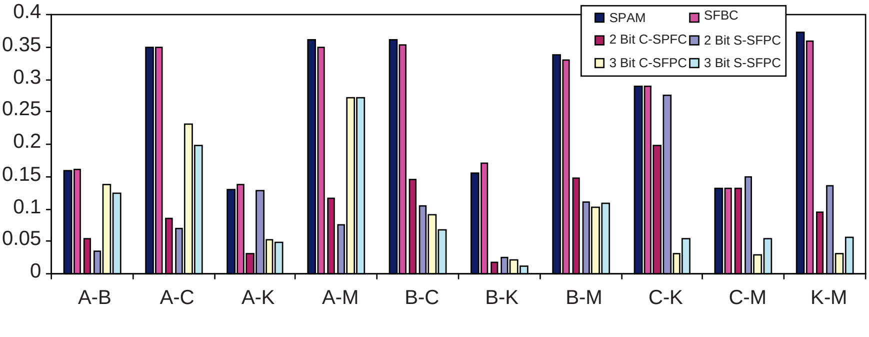
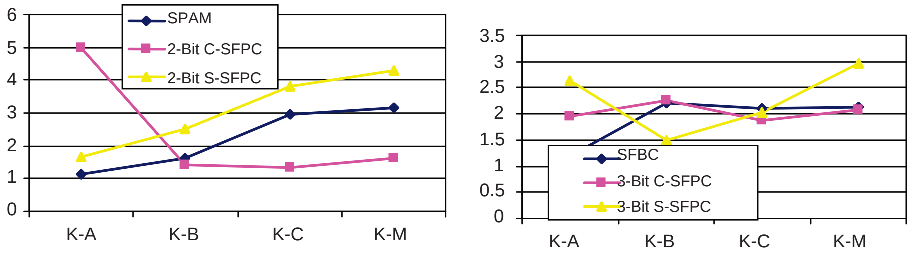


Figure 25.15 Spectral similarity values among A, B, C, K, and M produced by SPAM, SFBC, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit C-SFPC, and 3-bit S-SFPC.



(a) 2-bit SPAM, 2-bit C-SFPC, and 2-bit S-SFPC (b) 3-bit SFBC, 3-bit C-SFPC, and 3-bit S-SFPC

Figure 25.16 Comparative graphical plots of RSDPW values among the four mineral signatures A, B, C, and M produced by (a) 2-bit SPAM, 2-bit C-SFPC, and 2-bit S-SFPC and (b) 3-bit SFBC, 3-bit C-SFPC, and 3-bit S-SFPC using K as the signature of interest $\mathbf{s}_1$.

Experiment 25.3.3.1 (r = average of five signatures in Figure 25.2)

In this experiment, the K was assumed to be the signature vector of interest $\mathbf{s}_1$ and other signature vectors are chosen to be $\mathbf{s}_2$ to be compared against $\mathbf{s}_1$. The selection of K was arbitrary and any other signature can also be used to serve the same purpose. With $\mathbf{s}_1 = \text{K}$, Figure 25.16 plots RSDPW values calculated among the four signature vectors $\mathbf{s}_2 = \text{A, B, C, and M}$ for 2-bit SPAM, 2-bit C-SFPC, and 2-bit S-SFPC as a 2-bit group shown in Figure 25.16(a) and for SFBC, 3-bit C-SFPC, and 3-bit S-SFPC as another group shown in Figure 25.16(b) for comparative analysis.

According to (25.19), a measure is effective if it produces a higher value of the RSDPW if two signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ are more distinct and a lower value of the RSDPW if two signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ are more similar. On the other hand, a measure is ineffective if it produces similar values of the RSDPW among all signatures. In light of this interpretation, the results in Figure 25.16 show that SFPC performed significantly better than SPAM and SFBC in the sense of the same bit rate as well as the fact that the more distinct the two signature vectors, the higher their RSDPW values and the more similar the two signature vectors, the smaller their RSDPW values.

In order to further demonstrate the generalization capability of SFPC, comparative graphical plots of spectral similarity values of A, B, C, K, and M produced by the 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC are shown in Figure 25.17(a) and comparative graphical plots of RSDPW values among the four mineral signature vectors A, B, C, and M produced by

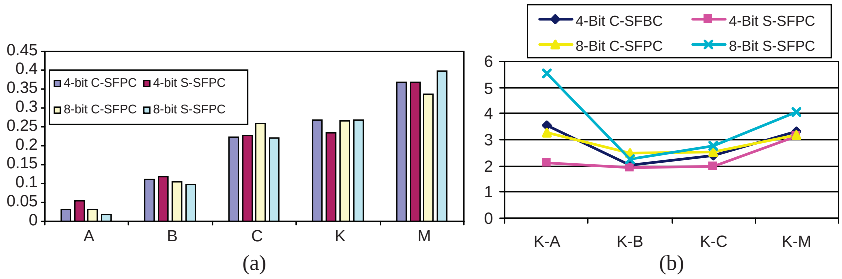


Figure 25.17 (a) Comparative plots of spectral similarity values produced by 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC. (b) Comparative graphical plots of RSDPW values among the four mineral signatures A, B, C, and M produced by 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC using the K as the signature of interest $\mathbf{s}_1$.

4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC, using the K as the signature vector of interest $\mathbf{s}_1$ are shown in Figure 25.17(b).

Experiment 25.3.3.2 ($\mathbf{r}$ = individual signature)

In this experiment, each of the five signature vectors in Figure 25.2 was selected as a reference signature vector $\mathbf{r}$. The same simulations conducted in Experiment 25.3.3.1 were also performed for the reference signature vector $\mathbf{r}$ = A, B, C, K, and M, respectively, for calculation of the RSDPW values via (25.19) where the signature vector of interest $\mathbf{s}_1$ was chosen to be same as the reference signature vector $\mathbf{r}$ for each case. Figure 25.18(a)–(e) shows comparative graphical plots of RSDPW values produced by 2-bit SPAM, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit SFBC, 3-bit C-SFPC, 3-bit S-SFPC, 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC, where one mineral signature vector is used as the reference signature vector $\mathbf{r}$ while the other four mineral signature vectors are used to calculate the RSDPW values.

Interestingly, according to the above experiments, 2-bit SFPC was still the best even though more bits were allowed for signature coding and the same conclusions made for Experiment 25.3.3.1 were also valid for Experiment 25.3.3.2. However, this was no longer true when the similar experiments were conducted for NIST gas data set in the following section. This is primarily due to the used AVIRIS data set that has only five signatures and 2-bit coders are sufficient to capture the complexity of their signature characterization.

25.3.4 NIST Gas Data Experiments

The spectral signature vector $\mathbf{s}_9$ is chosen as the signature vector of interest $\mathbf{s}_1$ in (25.19) and other signature vectors $\{\mathbf{s}_i\}_{i=2}^{8}$ were used as $\mathbf{s}_2$ in (25.19) to be compared against $\mathbf{s}_9$. This selection was random and any other signature vector could also be used as a reference signature vector. As discussed in Section 25.3.2, SFPC started coding from the band that yielded the maximum spectral variation between two signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ via (25.22). Since the range of reflectance values is widely spread, the natural logarithm function is used to replace $\mathbf{s}^{\mathrm{AC}}$ in (25.21) and (25.26) with $\ln \mathbf{s}^{\mathrm{AC}}$. It should be noted that the natural logarithm is a monotonically increasing function and will not have any effect on the results. By taking natural logarithm, equations (25.21) and (25.26) become

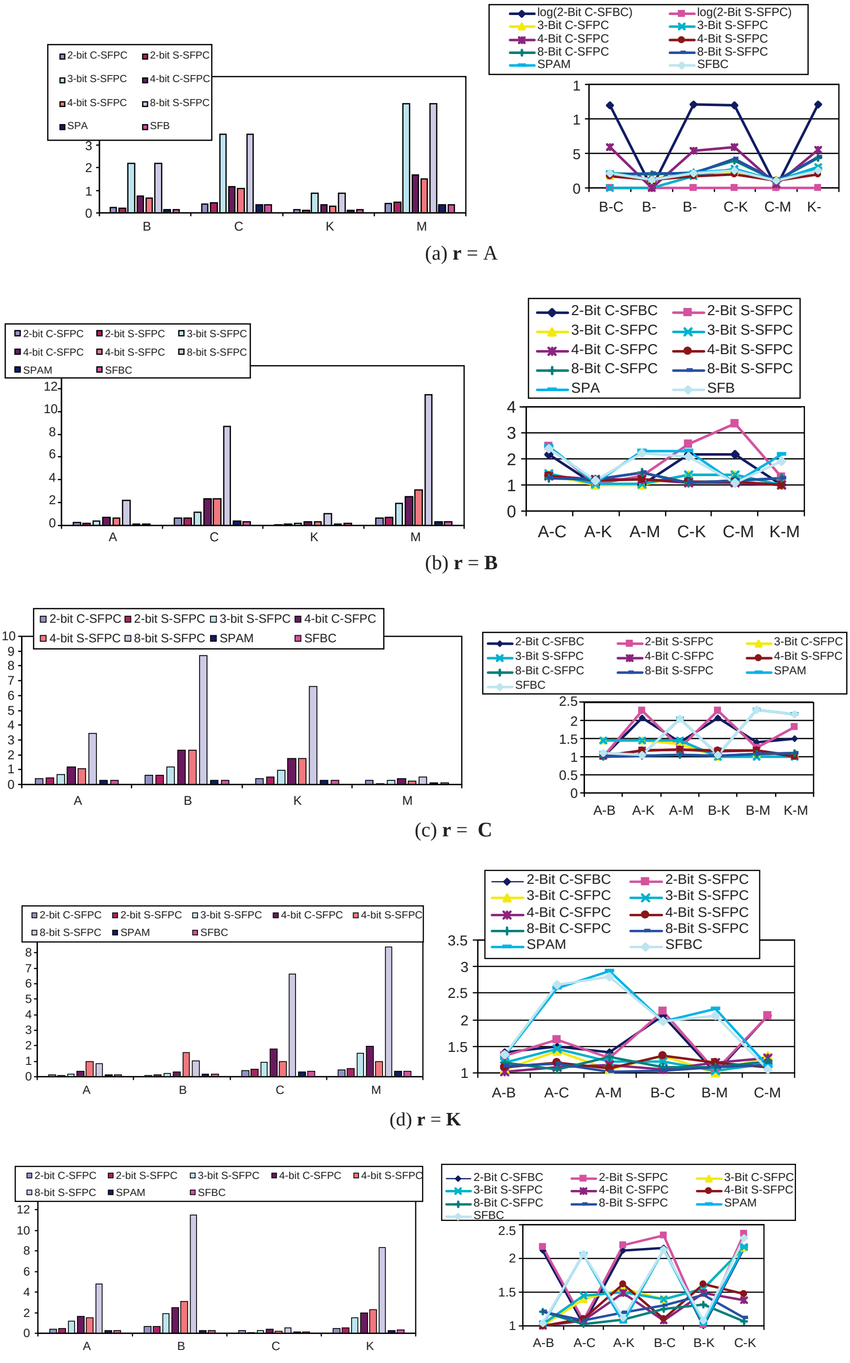
$$\mathrm{SFPC}(\mathbf{s}_1, \mathbf{s}_2) = |\ln(\mathbf{s}_1^{\mathrm{AC}}) - \ln(\mathbf{s}_2^{\mathrm{AC}})| \tag{25.27}$$

for C-SFPC and

$$\mathrm{SFPC}(\mathbf{s}_1, \mathbf{s}_2) = |\ln(\mathbf{s}_1^{\mathrm{AC},\Omega_1}) - \ln(\mathbf{s}_2^{\mathrm{AC},\Omega_1})| + |\ln(\mathbf{s}_1^{\mathrm{AC},\Omega_2}) - \ln(\mathbf{s}_2^{\mathrm{AC},\Omega_2})| \tag{25.28}$$

for S-SFPC.

The above two variations of SFPC were implemented along with 2-bit SPAM and 3-bit SFBC. Figure 25.19(a) plots the spectral similarity values among $\{\mathbf{s}_i\}_{i=2}^{9}$ produced by SPAM, SFBC, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit C-SFPC, and 3-bit S-SFPC, while Figure 25.19(b) is a spectral similarity value plot among $\{\mathbf{s}_i\}_{i=2}^{9}$ produced by SPAM, SFBC, 4-bit C-SFPC, and 4-bit S-SFPC (which gave the best discrimination performance among SFPC algorithms for this data set).



(a) **r** = A

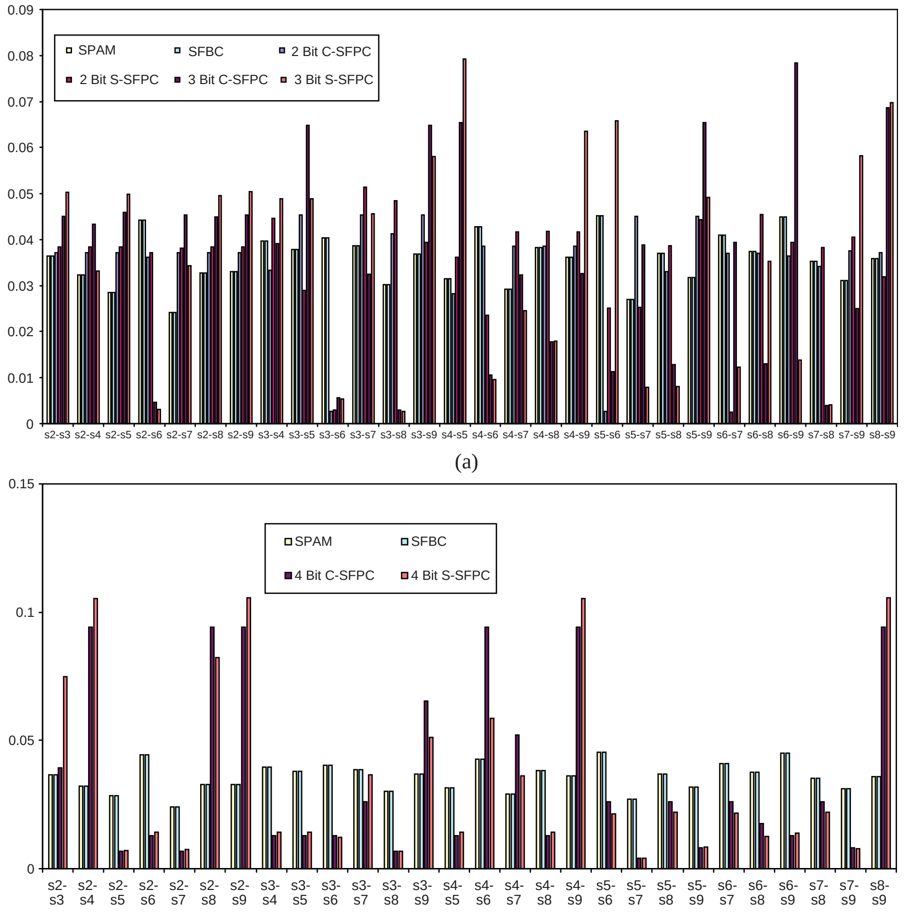
(b) **r** = **B**

(c) **r** = **C**

(d) **r** = **K**

(e) **r** = **M**

Figure 25.18 Comparative plots of spectral similarity values produced by SPAM, SFBC, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit C-SFPC, 3-bit S-SFPC, 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC along with their corresponding plots of RSDPW values.



(a)

(b)

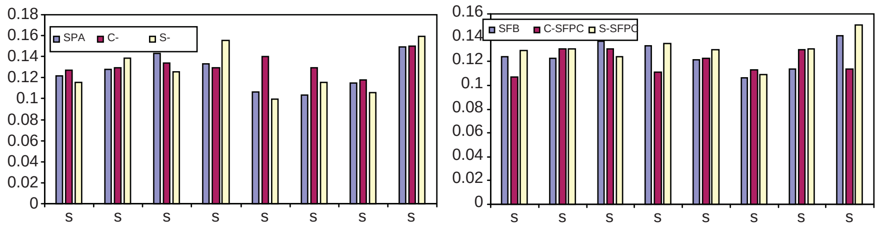
Figure 25.19 (a) Spectral similarity values among $\{\mathbf{s}_i\}_{i=2}^{9}$ produced by SPAM, SFBC, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit C-SFPC, and 3-bit S-SFPC. (b) Spectral similarity values among $\{\mathbf{s}_i\}_{i=2}^{9}$ produced by SPAM, SFBC, 4-bit C-SFPC, and 4-bit S-SFPC.

Similar to Experiments 25.3.3.1 and 25.3.3.2, two selections of the reference signature vector **r** were also conducted, that is, averaged signature vector and individual signature vectors.

Experiment 25.3.4.1 (r = average of $\{\mathbf{s}_i\}_{i=2}^{9}$)

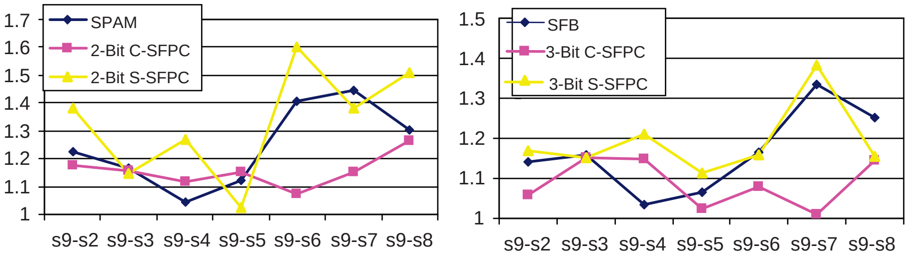
In this experiment, the average of the eight signature vectors was used as the reference signature vector while $\mathbf{s}_9$ was set as signature vector of interest. Figure 25.20 shows comparative graphical plots of spectral similarity values of $\mathbf{s}_2$, $\mathbf{s}_3$, $\mathbf{s}_4$, $\mathbf{s}_5$, $\mathbf{s}_6$, $\mathbf{s}_7$, $\mathbf{s}_8$, and $\mathbf{s}_9$ against the reference signature vector produced by 2-bit SPAM, 2-bit C-SFPC, and 2-bit S-SFPC and 3-bit SFBC, 3-bit C-SFPC, and 3-bit S-SFPC with their corresponding comparative plots of RSDPW values shown in Figure 25.21(a) and (b), respectively.

Figure 25.22(a) shows the spectral similarity values of $\mathbf{s}_2$, $\mathbf{s}_3$, $\mathbf{s}_4$, $\mathbf{s}_5$, $\mathbf{s}_6$, $\mathbf{s}_7$, $\mathbf{s}_8$, and $\mathbf{s}_9$ against the reference signature vector produced by the 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC and Figure 24.22(b) shows their respective RSDPW values.



(a) 2-bit SPAM, 2-bit C-SFPC, and 2-bit S-SFPC (b) 3-bit SFBC, 3-bit C-SFPC, and 3-bit S-SFPC

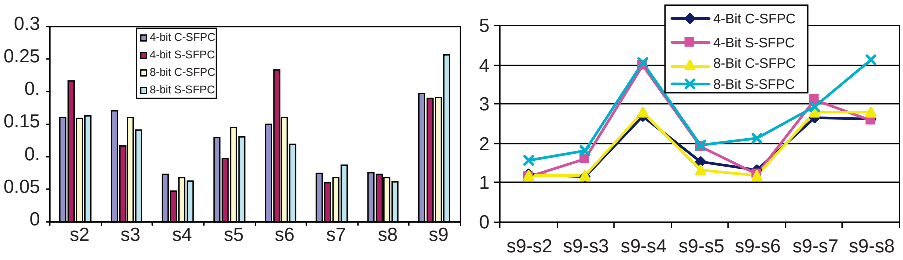
Figure 25.20 Comparative plots of spectral similarity values produced by 2-bit SPAM, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit SFBC, 3-bit C-SFPC, and 3-bit S-SFPC.



(a) 2-bit SPAM, 2-bit C-SFPC, and 2-bit S-SFPC (b) 3-bit SFBC, 3-bit C-SFPC, and 3-bit S-SFPC

Figure 25.21 Comparative graphical plots of RSDPW values among the signatures $\{\mathbf{s}_i\}_{i=2}^{8}$ produced by 2-bit SPAM, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit SFBC, 3-bit C-SFPC, and 3-bit S-SFPC using $\mathbf{s_9}$ as the signature of interest $\mathbf{s_1}$.

According to interpretation of (25.19), the higher the RSDPW value, the better the discrimination. So, from Figures 25.21 and 25.22(b), it is apparent that S-SFPC was the best, while C-SFPC was the worst and SPAM and SFBC were in between. Furthermore, unlike Experiments 25.3.2.1 and 25.3.2.2 where 2-bit SFPC outperforms high bit rates of SFPC, a comparison between Figures 25.21 and 25.22(b) shows that SFPC with higher bit rates generally



(a) (b)

Figure 25.22 (a) Comparative plots of spectral similarity values produced by 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC. (b) Comparative graphical plots of RSDPW values among the signatures $\{\mathbf{s}_i\}_{i=2}^{8}$ produced by 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC using $\mathbf{s_9}$ as the signature of interest $\mathbf{s_1}$.

performed better than SFPC with lower bit rates. This is primarily due to very high spectral dimensionality of the gas data that was nearly four times that of the AVIRIS data considered in Experiments 25.3.2.1 and 25.3.2.2.

Experiment 25.3.4.2 (r = individual signature)

Figure 25.23 shows comparative graphical plots of RSDPW values produced by 2-bit SPAM, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit SFBC, 3-bit C-SFPC, 3-bit S-SFPC, 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC, where signature vector $\mathbf{s}_9$ is used as the reference signature vector $\mathbf{r}$ while the signature vectors $\{\mathbf{s}_i\}_{i=2}^{8}$ are used to calculate their RSDPW values.

As seen from Figure 25.23(a) and (b), 2-bit coders performed very poorly compared to 3-bit coders in terms of spectral discrimination among all the eight signature vectors. Since this data set was more complicated and sophisticated than the five AVIRIS reflectance data considered in Figure 25.2, a coder with a bit rate higher than 3 may be desirable to improve the spectral discrimination performance. Because SPAM and SFBC did not have generalizability, only SFPC was implemented in Figure 25.23(c) with bit rates from 4 to 8 bits. As seen in Figure 25.23(c), at bit rates of 4 or higher, S-SFPC performed significantly better than the other spectral measures. Compared to the AVIRIS experiments conducted in Section 25.3.3, SFPC for gas data worked more effectively when the band number was increased.

25.4 Real Image Experiments

Two real image data sets, the Cuprite data of Figure 1.12 and the 15-panel HYDICE image scene in Figure 1.15, have been used for experiments in previous chapters. Since the experiments in Sections 25.2 and 25.3 have used the AIVRIS data of Figure 1.9 that are derived from the scene in Figure 1.12, this section will only focus on the 15 panels in Figure 1.15(b) and use the 5 panel signatures in Figure 1.16 for experiments. It should be noted that despite that the panel signatures are obtained from real image pixel vectors, the following experiments are performed on the five panel signatures as signature vectors not pixel vectors. Therefore, no sample spectral correlation among pixels is considered in signature coding.

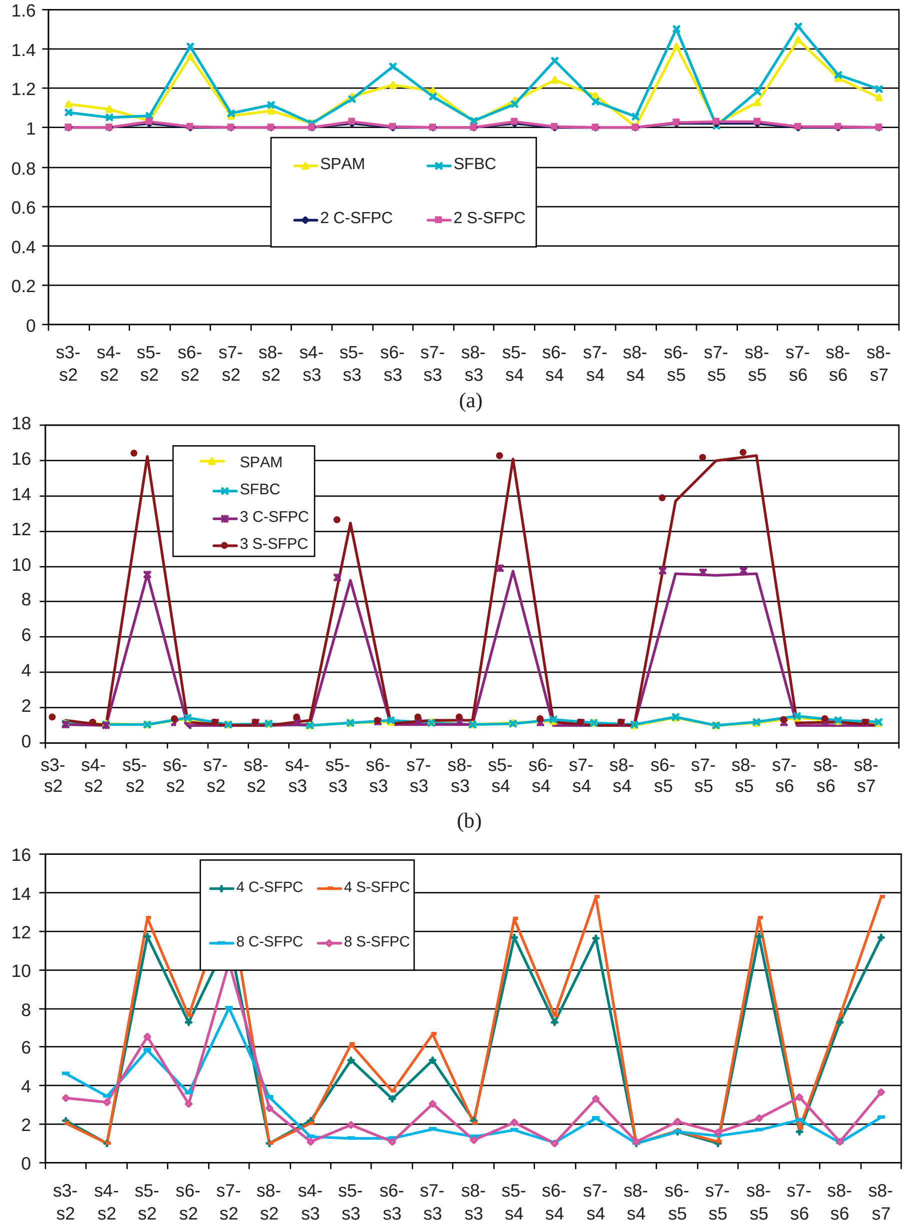
25.4.1 SDFC

Figure 25.24 shows the results of spectral similarity values produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC.

Figure 25.25(a) plots comparative results of spectral similarity values of $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ produced by the four different coding methods, SPAM, SFBC, HD-SDFC, and AVD-SDFC. Once again, there was no clear visual assessment to determine how one coding method performed better than another. RSDPW was calculated for this purpose to further evaluate discriminatory power.

Experiment 25.4.1.1 (r = average of five signatures in Figure 1.16)

Figure 25.25(b) plots RSDPW values calculated by (25.19) using $\mathbf{p}_{\text{ave}}$ obtained by averaging all the five panel signature vectors as the reference signature vector and $\mathbf{p}_i$ designated as the signature vector of interest $\mathbf{s}_1$ and other four signature vectors $\mathbf{p}_j$ with $j \neq i$ chosen to be $\mathbf{s}_2$ to be compared against $\mathbf{s}_1$. As can be seen from the experiments, SDFC also outperformed SPAM and SBFC in discriminatory power and AVD-SDFC seemed to perform slightly better than HD-SDFC in general. For other cases where signature $\mathbf{s}_1$ was selected to be a different panel signature, the conclusion was very similar. Their results are not included here.



(a)

(b)

(c)

Figure 25.23 Comparative graphical plots of RSDPW values produced by (a) 2-bit SPAM, 3-bit SFBC, 2-bit C-SFPC, and 2-bit S-SFPC, (b) 2-bit SPAM, 3-bit SFBC, 3-bit C-SFPC, and 3-bit S-SFPC, and (c) 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC, where $\mathbf{s}_9$ is the reference signature $\mathbf{r}$.

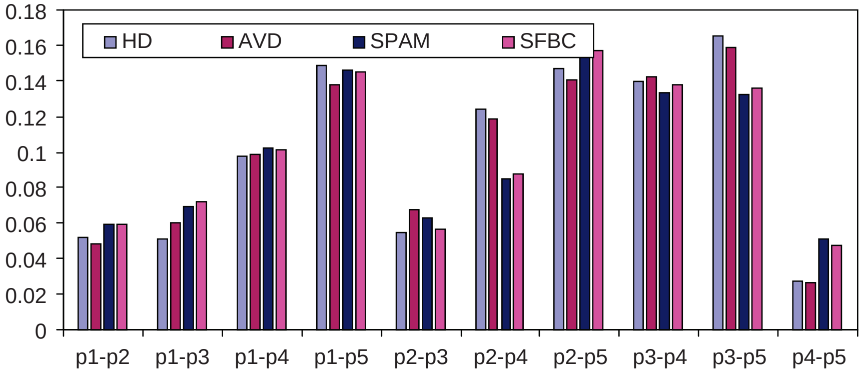


Figure 25.24 Spectral similarity values among $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC.

Experiment 25.4.1.2 ($\mathbf{r} = $ individual signature)

In this case, we choose each of the five signature vectors as a reference signature, that is, $\mathbf{r} = \mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$. In the mean time, these five panel signature vectors are also set to the signature vector of interest $\mathbf{s}_1 = \mathbf{r}$ for calculation of the RSDPW via (25.19). Figure 25.26(a)–(e) plots RSDPW values and shows the results of spectral similarity values between $\mathbf{s}_1 = \mathbf{r}$ and $\mathbf{s}_2 =$ other four signature vectors produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC along with plots of their corresponding RSDPW values.

According to Figure 25.26, both versions of SDFC also performed better than SPAM and SFBC in terms of RSDPW. As noted, unlike the simulation results where AVD-SDFC performed better than the HD-SDFC, the performance of HD-SDFC was slightly better than AVD-SDFC.

25.4.2 SFPC

Similar experriments conducted in Section 25.4.1 were also performed by SFPC on the five panel signature vectors. Figure 25.27 shows the spectral similarity values among $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ produced by SPAM, SFBC, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit C-SFPC, 3-bit S-SFPC,

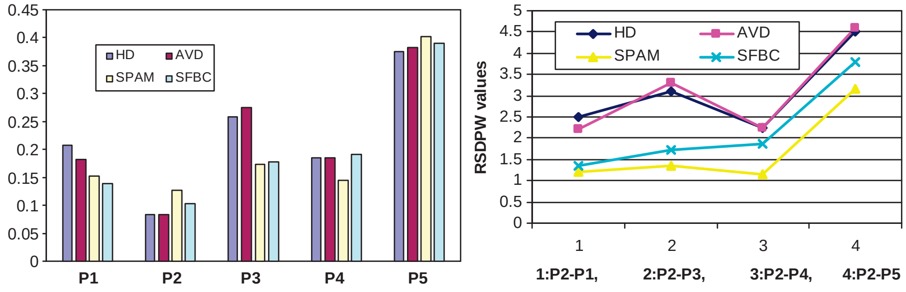
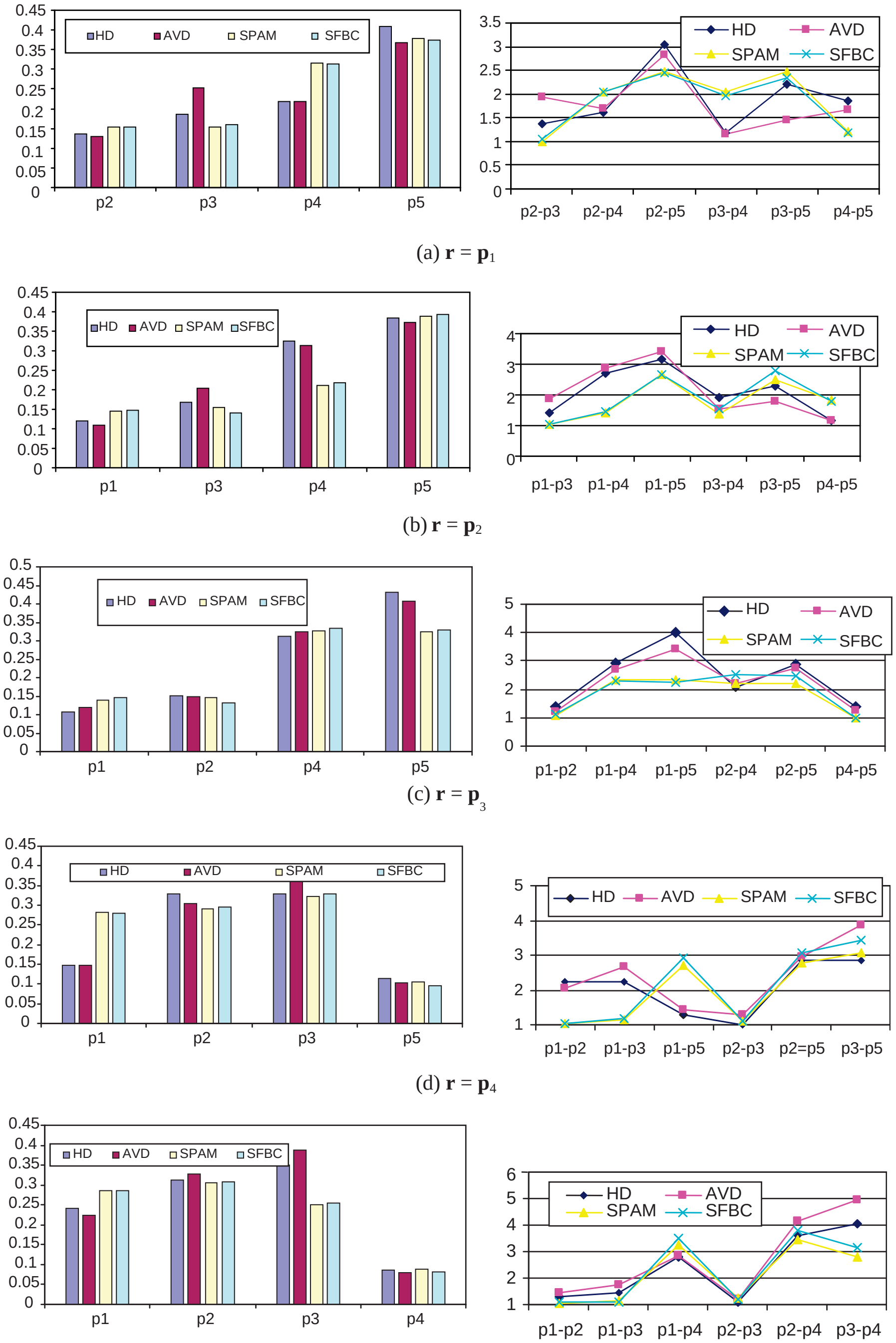


Figure 25.25 Comparative plots of spectral similarity values of the panel data produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC along with plots of RSDPW values.



(a) $\mathbf{r} = \mathbf{p}_1$

(b) $\mathbf{r} = \mathbf{p}_2$

(c) $\mathbf{r} = \mathbf{p}_3$

(d) $\mathbf{r} = \mathbf{p}_4$

(e) $\mathbf{r} = \mathbf{p}_5$

Figure 25.26 Comparative plots of spectral similarity values produced by SPAM, SFBC, HD-SDFC, and AVD-SDFC along with their corresponding plots of RSDPW values.

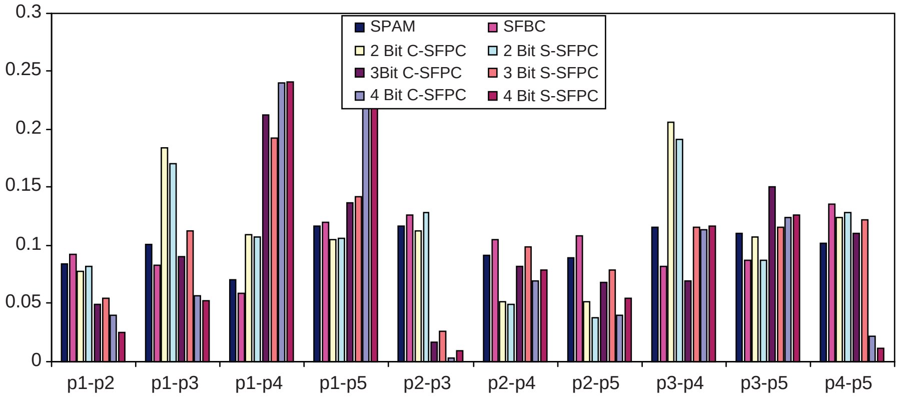


Figure 25.27 Spectral similarity values among $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ produced by SPAM, SFBC, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit C-SFPC, 3-bit S-SFPC, 4-bit C-SFPC, and 4-bit S-SFPC.

4-bit C-SFPC, and 4-bit S-SFPC, where the best performance seemed was produced by 4 bit S-SFPC followed by 4-bit C-SFPC.

In order to further quantitatively measure discriminatory powers of SPAM, SFBC, C-SFPC, and S-SFPC with variable bit rates, RSDPW via (25.19) was once again used for performance evaluation.

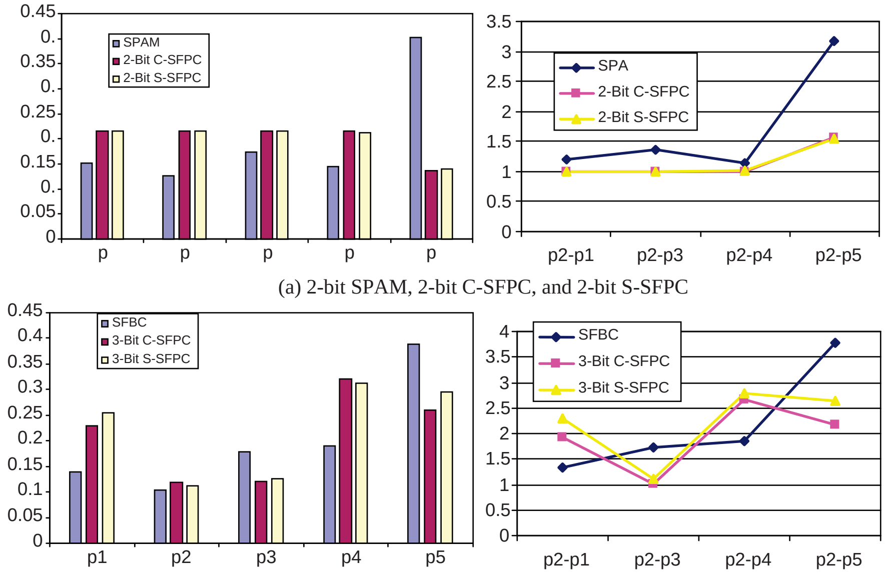
Experiment 25.4.2.1 (r = average of the five panel signatures in Figure 1.16)

Using the average of five signature vectors as the reference signature vector, Figure 25.28 shows comparative graphic plots of spectral similarity values of $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ produced by 2-bit SPAM, 2-bit C-SFPC, and 2-bit S-SFPC and comparative plots of spectral similarity values produced by 3-bit SFBC, 3-bit C-SFPC, and 3-bit S-SFPC. In the mean time, Figure 25.28 also plots their corresponding RSDPW values using $\mathbf{p}_{\text{ave}}$ obtained by averaging all the five panel signature vectors as the reference signature vector and each panel signature vector $\mathbf{p}_i$ designated as the signature vector of interest $\mathbf{s}_1$ and other four signature vectors $\mathbf{p}_j$ with $j \neq i$ chosen to be $\mathbf{s}_2$ to be compared against $\mathbf{s}_1$.

As shown in Figure 25.28(a), SFPC did not perform well in the 2-bit case due to the close similarity of the panel spectral signature vectors. However, the performance of SFPC was significantly improved when the bit rate was increased from 2 to 3 bits in Figure 25.28(b) and it performed better than the 3-bit SFBC except the case between $\mathbf{p}_4$ and $\mathbf{p}_5$. As the bit rate was increased to 4 and 8 bits, the performance of SFPC was improved as shown in Figure 25.29(a) and (b).

Experiment 25.4.2.2 (r = individual signature)

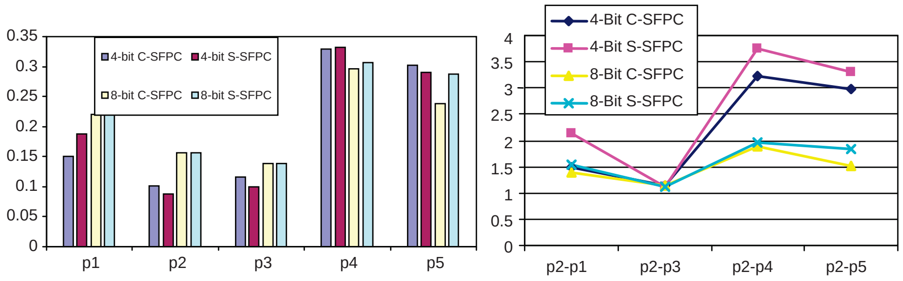
In this case, we chose any of five signature vectors as a reference signature vector where each of the five panel signature vectors $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ could be used as the signature vector of interest $\mathbf{s}_1$ for the calculation of the RSDPW via (25.19) while one of the remaining four panels signature vectors is designated as signature vector $\mathbf{s}_2$. Since the spectral similarity values of various coding methods are already plotted in Figure 25.29(a), Figure 25.30(a)–(e) only shows comparative graphical plots of RSDPW values produced by 2-bit SPAM, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit SFBC, 3-bit C-SFPC, 3-bit S-SFPC, 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC, where one panel signature vector was used as the reference signature vector **r** while the other four panel signature vectors were used to calculate RSDPW values.



(a) 2-bit SPAM, 2-bit C-SFPC, and 2-bit S-SFPC

(b) 3-bit SFBC, 3-bit C-SFPC, and 3-bit S-SFPC

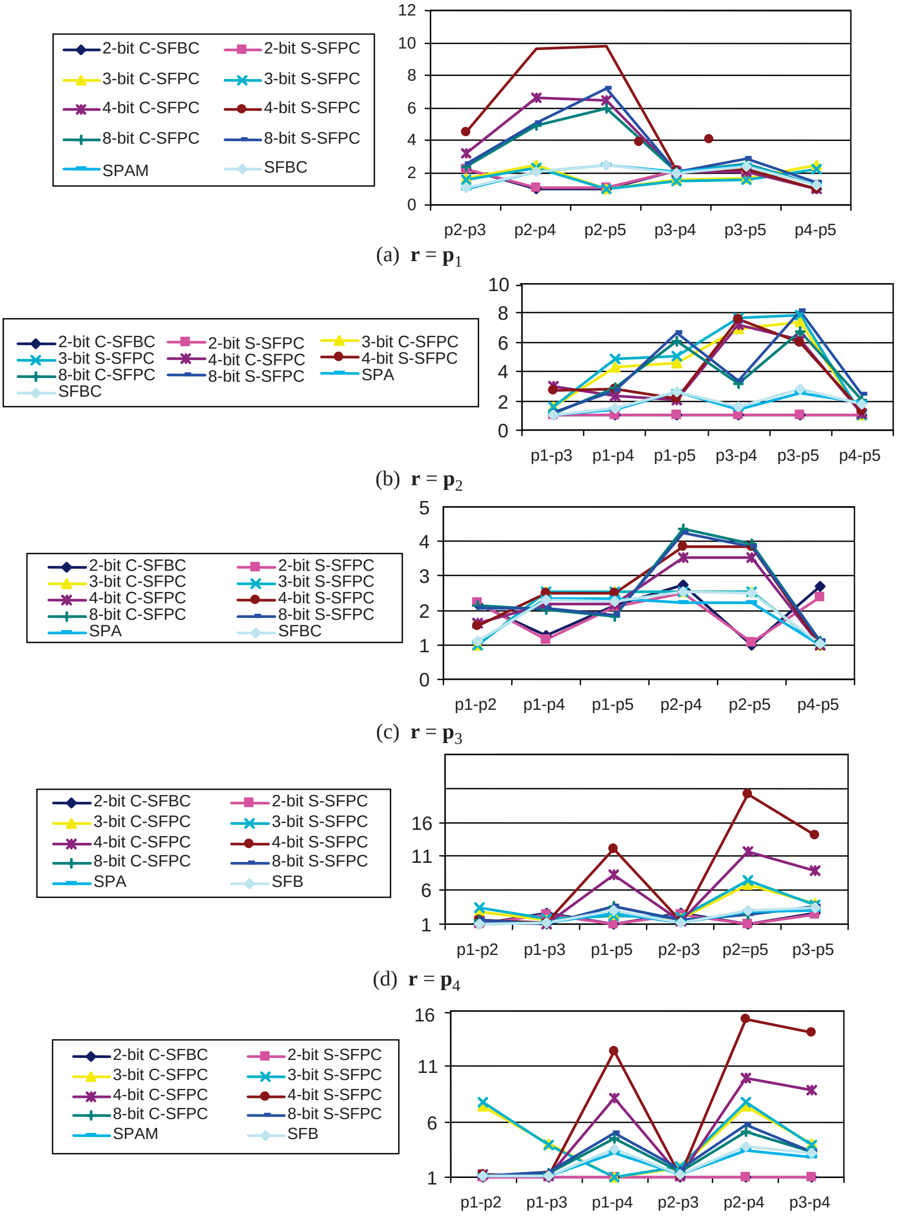
Figure 25.28 Comparative plots of spectral similarity values produced by 2-bit SPAM, 2-bit C-SFPC, 2-bit S-SFPC, 3-bit SFBC, 3-bit C-SFPC, and 3-bit S-SFPC along with their corresponding graphical plots of RSDPW values using $\mathbf{p}_{\text{ave}}$ obtained by averaging all the five panel signatures as the reference signature and each panel signature vector $\mathbf{p}_i$ designated as the signature vector of interest $\mathbf{s}_1$ and other four signature vectors $\mathbf{p}_j$ with $j \neq i$ chosen to be $\mathbf{s}_2$ to be compared against $\mathbf{s}_1$.



(a)

(b)

Figure 25.29 (a) Comparative plots of spectral similarity values produced by 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC. (b) Comparative graphical plots of RSDPW values among the four panel signature vectors $\mathbf{p}_1$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ produced by 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC using $\mathbf{p}_{\text{ave}}$ obtained by averaging all the five panel signature vectors as the reference signature and each panel signature $\mathbf{p}_i$ designated as the signature vector of interest $\mathbf{s}_1$ and other four signature vectors $\mathbf{p}_j$ with $j \neq i$ chosen to be $\mathbf{s}_2$ to be compared against $\mathbf{s}_1$.



(a) $\mathbf{r} = \mathbf{p}_1$

(b) $\mathbf{r} = \mathbf{p}_2$

(c) $\mathbf{r} = \mathbf{p}_3$

(d) $\mathbf{r} = \mathbf{p}_4$

(e) $\mathbf{r} = \mathbf{p}_5$

Figure 25.30 Comparative plots of RSDPW values produced by SPAM, SFBC, 2-bit C-SFPC. 2-bit S-SFPC, 3-bit C-SFPC, 3-bit S-SFPC, 4-bit C-SFPC, 4-bit S-SFPC, 8-bit C-SFPC, and 8-bit S-SFPC.

According to the above experiments, the best performance was produced by 4-bit S-SFPC. Several remarks are worth noting:

1. SFPC generally performed well and better than the SPAM and SFBC when spectral characteristics were sophisticated and required more bits for quantization such as the bit rate greater than

2 bits. On the other hand, it performed worse than SPAM and SFBC if the bit rate was low such as 1-bit or 2-bit coding. Therefore, SFPC generally performed better than SPAM and SFBC for signatures that were spectrally similar and required high bit rates for coding.
2. S-SFPC was usually better than C-SFPC because there was a lack of correlation between the first and last bands used in C-SFPC.
3. As for the reference signature vector, the following guideline is recommended. If the signature vectors to be encoded are relatively distinct in the sense of spectral similarity, the reference signature vector can be selected to their average. Otherwise, the reference signature vector can be selected as one of the signature vectors. This was demonstrated by the above experiments. For example, as $\mathbf{p}_1$, $\mathbf{p}_2$, and $\mathbf{p}_3$ are spectrally similar, using the averaged signature vector as a reference was less effective than using individual signature vectors as a reference signature vector.
4. Compared to SPAM and SFBC, SFPC has a generalization ability that can be extended to any arbitrary-bit coder, a major advantage that cannot be gained by the 2-bit coder, SPAM, and the 3-bit coder, SFBC.

25.5 Conclusions

This chapter develops two new vector coding techniques, called SDFC and SFPC, for signature coding, each of which can be considered as a binary encoder using variable-bit memory. With this new interpretation, two notable coding methods SPAM and SFBC can be viewed as binary encoders using 1-bit memory and 2-bit memory, respectively. A major difference between SDFC and SPAM/SFBC is that the former dictates gradient changes in spectral variation in terms of spectral textures occurred among three consecutive adjacent bands compared to the latter that only captures changes in spectral values among two adjacent bands. As a result, SDFC can be considered as a second-order coding method because it uses gradient changes, while SPAM and SFBC can be considered as first-order coding methods because they use changes only in spectral value between adjacent bands. Accordingly, experimental results show that SDFC performed more effectively in general than SPAM and SFBC in the characterization of spectral profiles of signatures. On the other hand, SFPC is developed based on bit allocation that is also completely different from SPAM and SFBC. More specifically, SFPC makes use of AC to keep track of between-band spectral variations and different bit rates to encode probabilistic behaviors of spectral changes across the entire spectral coverage. Consequently, SFPC can be implemented as an arbitrary-bit encoder, where the number of bits to be used is designed to capture subtle probabilistic changes resulting from spectral variations. In order to implement SFPC, two versions of SFPC are developed for spectral signature coding, which are circular-SFPC and split-SFPC. The conducted experimental results have demonstrated that SFPC also performed better than SPAM and SFBC.

26

Progressive Coding for Spectral Signatures

Spectral signature coding is an effective means of characterizing spectral features and is performed by encoding signature vectors sequentially. This chapter develops a rather different encoding concept called progressive signature coding (PSC) that encodes a signature vector in a progressive manner. More specifically, it progressively encodes a spectral signature vector in multiple stages, each of which captures different but disjoint spectral information contained in the spectral signature vector. As a result of such a progressive coding, a profile of progressive changes in spectral variation for a spectral signature vector can be generated for spectral characterization. The proposed idea is very simple and evolved from the pulse code modulation (PCM), which is a commonly used quantization technique in communications and signal processing. It expands PCM to multiple-stage PCM (MPCM) in the sense that a signature vector can be decomposed and quantized by PCM progressively in multiple stages for spectral characterization. In doing so, MPCM generates a priority code for a spectral signature vector so that its spectral information captured in different stages can be prioritized in accordance with significance of changes in spectral variation. Such a coding, referred to as MPCM-based progressive spectral signature coding (MPCM-PSSC), can be very useful in many applications in hyperspectral data exploitation.

26.1 Introduction

Spectral signature coding (SSC) is a scheme, a rule, or a mapping that transforms spectral values into a new set of symbols in a specific manner that a signature vector can be represented by the new symbols more effectively or efficiently. In hyperspectral data, each data sample is acquired by hundreds of spectral channels to form a column vector that can be used to diagnose subtle material substances based on their spectral characteristics. Therefore, taking advantage of such intraband spectral information (e.g., spectral information provided by spectral channels within a hyperspectral data sample vector) is one of the great benefits resulting from hyperspectral data. However, this also is traded off for a price that many unknown spectral signature vectors may be also extracted to further complicate spectral analysis. So, one of the major challenges in hyperspectral data exploitation is how to best utilize the spectral information provided by hyperspectral data to accomplish tasks such as detection, discrimination, classification, identification, while discarding undesired information caused by unwanted interference such as noise.

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

This chapter investigates a new approach to SSC, called progressive spectral signature coding (PSSC), where SSC is carried out in a progressive fashion rather than sequential coding by classical coding methods. It is a technique that can decompose a signature vector in multiple stages where each of these stages captures spectral changes across a range of wavelengths in a progressive manner. As a consequence, it provides a profile of progressive changes in spectral variation that describes the spectral behavior of a data sample vector in various stages. Accordingly, we can consider PSSC as "soft" coding in a progressive procedure as opposed to SSC that can be viewed as "hard" coding performed by classical coding techniques with binary decisions. One technique, called multistage pulse coding modulation (MPCM), is of particular interest and can be used for PSSC. It was previously developed for progressive image compression (Cheng and Chang, 1992). The success of progressive processing has been also demonstrated in image progressive reconstruction (Wang and Goldberg, 1988, 1989), progressive edge detection (Cheng and Chang, 1992; Cheng, 1993; Chang et al., 1992), and progressive text detection (Du et al., 2003, 2004). It is indeed a one-dimensional (1D) transform coding technique, which encodes a 1D signal progressively according to a priority assigned to each signal value. The signal priorities are then determined by changes between two successive signal values. In the above-mentioned applications, a 1D signal function to be processed for progressive coding is either a temporal function such as a speech signal whose signal values are temporal signals at different time instants or a set of image pixels arranged in a particular format such as raster format whose signal values are image pixels at particular spatial coordinates. Interestingly, since a hyperspectral data sample vector acquired by hundreds of contiguous spectral channels can be represented by a column vector, its spectrum can be considered as a 1D signature signal with each signal corresponding to a spectral value of a particular wavelength in spectral dimension. Using this interpretation, MPCM can be implemented to capture progressive changes of spectral variation occurred in spectral wavelengths that are used to acquire the spectral signature of the data sample vector, referred to as spectral signature vector.

One of the major advantages of using PSSC is characterization of a spectral signature vector in progressive changes across its spectral channels. This unique feature cannot be accomplished by any "hard" coding-based SSC methods. Another advantage is a spectral profile of progressive changes produced for a spectral signature vector that can be used for various applications such as discrimination, classification, and identification. It is often the case that two spectral signature vectors may be very similar in terms of a spectral signature vector direction measured by spectral angle mapper (SAM) (Schowengerdt, 1997; Chang, 2000; Chang, 2003a), but in fact, they do have very different spectral profiles of progressive changes in a range of spectral channels. PSSC provides such a progressive spectral profile for signature characterization. The third advantage is change detection that is a major issue in land-cover remote sensing image classification and has been generally performed by temporal processing. PSSC offers a different perspective in terms of change detection in spectral variation. The fourth advantage of PSSC is that it can be viewed as a progressive implementation of a sequence of binary coding where a sequence of bit plane coding (Gonzalez and Woods, 2002) is performed progressively with decreasing thresholds.

This chapter develops an MPCM-based PSSC (MPCM-PSSC) and provides a new look at how SSC can be accomplished progressively for signature characterization. The idea is derived from the success of MPCM in text detection for video images (Du et al., 2003) where the edges of a text were detected more effectively in a progressive manner. Such progressive edge detection seems to be very useful in hyperspectral signature characterization. Additionally, MPCM-PSSC has further advantages. It generates a priority code that keeps track of progressive changes in spectral variation. The larger the change in a spectral wavelength is, the higher the priority of this particular wavelength. Such MPCM-PSSC-generated priority codes provide fingerprints of a spectral signature vector via priority code words assigned to each of spectral wavelengths. Here, the term of "code" is referred to as a code book made up of "code words" that are used for encoding. Another important advantage resulting from MPCM-PSSC-generated priority codes are progressive decomposition of a spectral signature vector in accordance with the priority code words assigned to each of spectral wavelengths. The resulting progressive decomposition delineates a profile of progressive changes in spectral variation that can be used for discrimination and identification of a spectral signature, a feature that cannot be achieved by any spectral similarity measure. Furthermore, MPCM-PSSC-generated priority codes can progressively reconstruct a spectral signature vector literally by the priority code words assigned to spectral wavelengths. This progressive signature reconstruction enables users to see how spectral changes are updated in order to recover the original signature vector encoded by MPCM-generated priority codes. Most importantly, MPCM-PSSC priority codes can describe progressive transitions of spectral values from one spectral band to another via a simple coding scheme with a detailed profile of a spectral signature vector across spectral wavelengths in terms of progressive changes in spectral variation. This capability makes MPCM-PSSC unique. It distinguishes MPCM-PSSC from a spectral similarity measure that can only measure the closeness or similarity between two spectral signature vectors, but not progressive spectral signature similarity changes from band to band.

To facilitate our analysis, a distinction between discrimination and identification suggested in Chang (2003a) is also made in this chapter. The former is performed among a set of signatures where one signature vector is discerned from another compared to the latter carried out by verifying a signature vector via a database (spectral library). Consequently, algorithms designed for discrimination and identification are slightly different. In particular, a threshold is generally required for signature discrimination to discriminate one signature from another. On the other hand, signature identification via a database can be performed directly by finding the one in the database that most matches the signature vector to be identified. In MPCM-PSSC, the signature discrimination and signature matching are measured by the priority code words using the Hamming distance. Finally, simulations and real data experiments are conducted to demonstrate the utility of MPCM-PSSC in applications of signature discrimination and identification.

26.2 Multistage Pulse Code Modulation

In this section, we present a new concept called multistage pulse coding modulation (MPCM) that can be used for encoding spectral signatures in a progressive manner.

MPCM was originally developed for applications in image progressive transmission and reconstruction (Tzou, 1987; Wang and Goldberg, 1988, 1989). It can be viewed as a progressive version of a commonly used coding scheme in communications, pulse code modulation (PCM). It expands the hard-decision PCM-based quantizer to a soft-decision quantizer in such a fashion that it allows PCM to have a nondecision region that passes on its decisions to next stage progressively. As a result, a decision can be refined stage-by-stage so as to improve quantization results. The detailed idea of MPCM can be described as follows.

PCM is a quantizer, denoted by $Q(x)$ that requires a set of quantization levels $\{\Delta_k\}_{k=1}^{M}$ and a corresponding set of quantization thresholds $\{\tau_k\}_{k=1}^{M}$. It quantizes a signal function $x(n)$ according to

$$Q(x(n)) = \Delta_k \quad \text{if} \quad x(n) \in [\tau_{k-1}, \tau_k) \tag{26.1}$$

where τ_0 and τ_M are initial conditions determined by the domain of the signal function $x(n)$. It is a hard decision-based quantizer, referred to as a hard quantization because $Q(x(n))$ must make a decision on the input $x(n)$ via (26.1) by assigning the quantization level Δ_k to $x(n)$.

MPCM expands the above PCM to multiple stages in the sense that $x(n)$ in (26.1) is encoded by a sequence of M soft decision-based quantizers $\{Q_k(x(n))\}_{k=1}^{\mathrm{M}}$, referred to as soft quantizers, in a progressive manner compared to the hard decision made by one single value Δ_k in (26.1). Unlike the hard decision-based quantizer described in (26.1) that makes its binary decision on $x(n)$ by a single threshold interval $[\tau_{k-1}, \tau_k)$ for each quantization level Δ_k, $Q_k(x(n))$ makes its decision based on three threshold intervals, $(-\infty, -\Delta_k]$, $(-\Delta_k, \Delta_k)$ and $[\Delta_k, \infty)$, and its quantization level Δ_k where the interval $(-\Delta_k, \Delta_k)$ is designated as a no-decision threshold interval. More specifically, a soft quantizer $Q_k(x(n))$ derived from $Q(x(n))$ via the kth quantization level Δ_k is defined by

$$Q_k(x(n)) = \begin{cases} -\Delta_k; & \text{if} \quad x(n) \in (-\infty, -\Delta_k] \\ x(n); & \text{if} \quad x(n) \in (-\Delta_k, \Delta_k) \\ \Delta_k; & \text{if} \quad x(n) \in [\Delta_k, \infty) \end{cases} \tag{26.2}$$

where the soft quantizer $Q_k(x(n))$ passes its input $x(n)$ without making any decision when the input $x(n)$ falls in the region $x(n) \in (-\Delta_k, \Delta_k)$ as described by Figure 26.1. The decision maker for such a soft quantizer Q_k defined by (26.2) stretches a hard limiter that can be considered as a sign detector to a soft limiter shown in Figure 26.1. From Figure 26.1, the consequence of soft decisions comes from the inclusion of the no-decision interval, $(-\Delta_k, \Delta_k)$ in the quantizer $Q_k(x(n))$.

MPCM takes advantage of such a soft quantizer $Q_k(x(n))$ described by (26.2) to perform quantization progressively in multiple stages specified by $\{\Delta_k\}_{k=1}^{M}$, which will be referred to as stage levels in MPCM. Assume that $\{\Delta_k\}_{k=1}^{\mathrm{M}}$ are strictly decreasing quantization levels, that is, $\Delta_1 > \Delta_2 > \cdots > \Delta_{\mathrm{M}} > 0$. Therefore, the no decision-made outputs passed by the kth soft quantizer by $Q_k(x(n))$ at stage k are further processed by the follow-up $(k+1)$st soft quantizer $Q_{k+1}(x(n))$ in the next stage that uses a smaller quantization level, $\Delta_{k+1} < \Delta_k$ to refine its decision. In other words, instead of encoding $x(n)$ directly into Δ_k by (26.1), the $x(n)$ is actually encoded by M soft quantizers $\{Q_k(x(n))\}_{k=1}^{M}$ one at a time progressively using M-refined quantization levels. As a result of using a sequence of progressive soft quantizers $\{Q_k(x(n))\}_{k=1}^{M}$, $x(n)$ can be decomposed into a set of binary-valued stage components, denoted by $\{\hat{x}_k(n)\}_{k=1}^{M}$ where $\hat{x}_k(n) \in \{0, 1\}$ for $1 \leq k \leq M$ so that the $x(n)$ can be approximated by the estimate of $x(n)$, $\hat{x}(n)$ by

$$\hat{x}(n) = \hat{x}_1\Delta_1 + \hat{x}_2\Delta_2 + \cdots + \hat{x}_{\mathrm{M}}\Delta_{\mathrm{M}} = \sum\nolimits_{k=1}^{\mathrm{M}} \hat{x}_k(n)\Delta_k \tag{26.3}$$

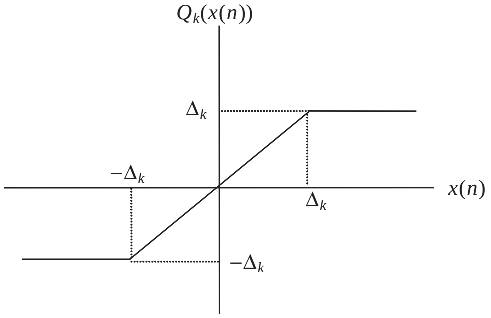


Figure 26.1 A soft quantizer $Q_k(x(n)$ described by (26.2).

The idea described by (26.3) is similar to representation of the $x(n)$ in terms of binary expansion with $\Delta_k = 2^{k-1}$. For a given set of stage levels $\{\Delta_k\}_{k=1}^{\mathrm{M}}$, $\hat{x}(n)$ can be represented by the value produced by (26.3) using the binary code word $(\hat{x}_1(n)\hat{x}_2(n)\ldots\hat{x}_{\mathrm{M}}(n))$ where the resulting approximation error given by $\varepsilon_{\mathrm{M}}(n) = x(n) - \hat{x}(n) = x(n) - \sum_{k=1}^{\mathrm{M}} \hat{x}_k(n)\Delta_k$. If $\varepsilon_{\mathrm{M}}(n) = 0$, $x(n)$ can be perfectly reconstructed by $\hat{x}(n) = \sum_{k=1}^{\mathrm{M}} \hat{x}_k(n)\Delta_k$. If $\varepsilon_{\mathrm{M}}(n) \neq 0$, it is required to encode both $\varepsilon_{\mathrm{M}}(n)$ and $(\hat{x}_1(n), \hat{x}_2(n), \ldots, \hat{x}_{\mathrm{M}}(n))$ to achieve the perfect reconstruction of $x(n)$. So, the above procedure can be regarded as an implementation of a sequence of M progressive binary encoders with a decreasing thresholds $\{\Delta_k\}_{k=1}^{M}$ to form a M-block length binary code word $(\hat{x}_1(n)\hat{x}_2(n)\ldots\hat{x}_{\mathrm{M}}(n))$ from which we can progressively reconstruct $x(n)$ by $\hat{x}_1(n)\Delta_1$,$\hat{x}_1(n)\Delta_1 + \hat{x}_2(n)\Delta_2$, $\hat{x}_1(n)\Delta_1 + \hat{x}_2(n)\Delta_2 + \hat{x}_3(n)\Delta_3$, etc., until the last stage M is reached, in which case $x(n)$ is approximated by (26.3). Such a progressive approximation is carried by the priority code word specified by the M-block length binary code word $(\hat{x}_1(n)\hat{x}_2(n)\ldots\hat{x}_{\mathrm{M}}(n))$. As an example for illustration, a real number $a \in (0, 1)$ can be approximated by an M-precision binary expansion $\hat{a} = \sum_{k=1}^{\mathrm{M}} \hat{a}_k 2^{-k}$ with the kth stage level Δ_k specified by 2^{-k} and the binary-valued coefficients of $\{\hat{a}_k\}_{k=1}^{\mathrm{M}}$. The M-block length binary code word is $(\hat{a}_1\hat{a}_2\ldots\hat{a}_{\mathrm{M}})$ is used to reconstruct and approximate the real value a progressively by $\hat{a}_1 2^{-1}$, $\hat{a}_1 2^{-1} + \hat{a}_2 2^{-2}$, $\hat{a}_1 2^{-1} + \hat{a}_2 2^{-2} + \hat{a}_3 2^{-3}$, and so on until the last stage, $\hat{a}_1 2^{-1} + \hat{a}_2 2^{-2} + \cdots + \hat{a}_{\mathrm{M}} 2^{-\mathrm{M}}$. The resulting approximation error $\varepsilon_{\mathrm{M}}(n)$ is then given by $\varepsilon_{\mathrm{M}} = a - \hat{a}$ with $\hat{a} = \sum_{k=1}^{M} a_k 2^{-k}$. The key issue is how to find a desired set of M soft binary quantizers, $\{Q_k(x(n))\}_{k=1}^{M}$ for a given set of quantization levels $\{\Delta_k\}_{k=1}^{M}$ to produce an optimal M-block length binary code for (26.3) in approximation. In doing so, the soft quantizer using the quantization level Δ_k defined by (26.2) can be used for the kth progressive soft quantizer in MPCM defined by

$$Q_k(\varepsilon_{k-1}(n)) = \begin{cases} -\Delta_k; & \text{if} \quad \varepsilon_{k-1}(n) \in (-\infty, -\Delta_k] \\ \varepsilon_{k-1}(n); & \text{if} \quad \varepsilon_{k-1}(n) \in (-\Delta_k, \Delta_k) \\ \Delta_k; & \text{if} \quad \varepsilon_{k-1}(n) \in [\Delta_k, \infty) \end{cases} \tag{26.4}$$

that takes the approximation error $\varepsilon_{k-1}(n) = x(n) - \sum_{j=1}^{k-1} \hat{x}_j(n)\Delta_j$ obtained at the $(k-1)$st stage as its input. It should be noted that $\varepsilon_{k-1}(n) = x(n) - \sum_{j=1}^{k-1} \hat{x}_j(n)\Delta_j$ in (26.4) is the approximation error obtained by a successive approximations using the binary code word $(\hat{x}_1(n)\hat{x}_2(n)\cdots\hat{x}_{\mathrm{M}}(n))$ up to the $(k-1)$st stage. The soft decision comes from the case that if $\varepsilon_{k-1}(n) \in (-\Delta_k, \Delta_k)$, $Q_k(\varepsilon_{k-1}(n)) = \varepsilon_{k-1}(n)$. A detailed implementation of MPCM is described as follows. A generalized version of MPCM can be found in Cheng (1993).

MPCM encoding algorithm for the nth signal value, x(n)

1. Initial condition

   Let $\{\Delta_k\}_{k=1}^{\mathrm{M}}$ be a set of M-stage levels that are used for MPCM and the initial approximation error $\varepsilon_0(n) = x(n) - \hat{x}(n-1)$ where $\hat{x}(n-1)$ is obtained by (26.3). Set $\hat{x}(0) = 0$ and $k = 1$.
2. At the kth stage, three cases are considered for the kth two-valued soft quantizer, Q_k defined by (26.4).

Case 1: If $\varepsilon_{k-1}(n) \geq \Delta_k$, then $Q_k(\varepsilon_{k-1}(n)) = \Delta_k$, $\hat{x}_k(n) = 1$ and set $\hat{x}_j(n) = 0$ for $k < j \leq \mathrm{M}$. In this case, the priority codeword $c(n)$ assigned to $x(n)$ is $c(n) = k$. Its diagram is depicted in Figure 26.2. Let $\varepsilon_{\mathrm{M}}(n) = \varepsilon_{k-1}(n) - \sum_{j=k}^{M} \hat{x}_j(n)\Delta_j = \varepsilon_{k-1}(n) - \Delta_k$. Go to step 4.

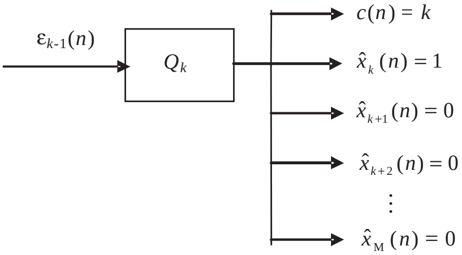


Figure 26.2 Case 1 for MPCM encoding algorithm.

Case 2: If $\varepsilon_{k-1}(n) \le -\Delta_k$, then $Q_k(\varepsilon_{k-1}(n)) = -\Delta_k$, $\hat{x}_k(n) = 0$ and set $\hat{x}_j(n) = 1$ for $k < j \le M$. In this case, the priority codeword $c(n)$ assigned to $x(n)$ is $c(n) = k$. Its diagram is depicted in Figure 26.3. Let $\varepsilon_{\mathrm{M}}(n) = \varepsilon_{k-1}(n) - \sum_{j=k}^{M} \hat{x}_j(n)\Delta_j = \varepsilon_{k-1}(n) - \sum_{j=k+1}^{M} \Delta_k$. Go to step 4.

Case 3: If $-\Delta_k < \varepsilon_{k-1}(n) < \Delta_k$, then $Q_k(\varepsilon_{k-1}(n)) = \varepsilon_{k-1}(n)$ and $\hat{x}_k(n) = \hat{x}_k(n-1)$. Its diagram is depicted in Figure 26.4. Go to step 3.

3. If $k < M$, let $k = k + 1$ and go to step 2. Otherwise, continue.
4. Go to the next sample, $(n+1)$st signal point, $x(n+1)$.

In the above MPCM encoding algorithm, a priority codeword is only assigned when a hard decision is made in a certain stage. When it occurs at stage k, the encoding for $x(n)$ is terminated and the priority code word for $x(n)$ is encoded as $c(n) = k$. In this case, the priority assigned to $x(n)$ is k, which indicates that there is a significant change in $x(n)$ at stage k. As a result, the higher the priority is, the greater the change and the smaller the index number of the stage is. According to MPCM, the set of quantization levels, $\{\Delta_k\}_{k=1}^{\mathrm{M}}$ are strictly decreasing quantization levels, that is, $\Delta_1 > \Delta_2 > \cdots > \Delta_{\mathrm{M}} > 0$. As a consequence, $c(n) = 1$ has the highest priority since there is a drastic change in stage 1 specified by the largest quantization level Δ_1. To the contrary, $c(n) = M$ indicates that there only has a small change in stage M because the quantization level Δ_M is the smallest quantization level. Interestingly, an immediate advantage resulting from MPCM encoding algorithm is that it allows one to decompose a signal sample $x(n)$ in multiple stages, that is, M stages and its priority code word indicates which stage the priority occurs where the signal sample makes a significant change.

Correspondingly, we also describe an MPCM decoding algorithm as follows. It decodes the nth signal sample $x(n)$ based on the encoded priority code word $c(n)$ along with the previous decoded $\hat{x}(n-1)$ that is an approximation of $x(n-1)$ via (26.3). In contrast to the MPCM encoding

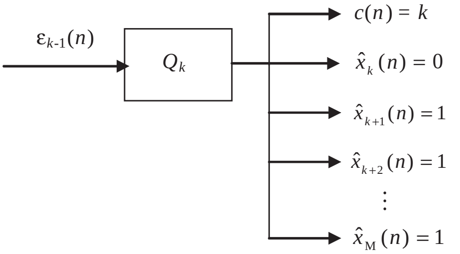
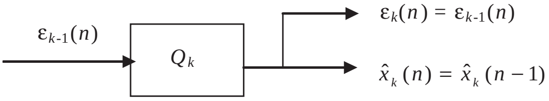


Figure 26.3 Case 2 for MPCM encoding algorithm.



$\varepsilon_{k-1}(n)$
Q_k
$\varepsilon_k(n) = \varepsilon_{k-1}(n)$
$\hat{x}_k(n) = \hat{x}_k(n-1)$

Figure 26.4 Case 3 for MPCM encoding algorithm.

algorithm that decomposes the nth signal sample stage-by-stage in M stages, MPCM decoding algorithm reconstructs the nth signal samples stage-by-stage based on its priority code word $c(n)$.

MPCM decoding algorithm for $x(n)$

1. *Initial condition*:

Let $\{\Delta_k\}_{k=1}^{\mathrm{M}}$ be a set of M-stage levels that are used for MPCM. Set the initial condition as $\hat{x}(0) = 0$, and $\hat{x}(n-1)$ is the reconstruction of $x(n)$ which is unknown and can be expressed by $\{\hat{x}_k(n-1)\}_{k=1}^{M}$ as

$$\hat{x}(n-1) = \hat{x}_1(n-1)\Delta_1 + \hat{x}_2(n-1)\Delta_2 + \cdots + \hat{x}_{\mathrm{M}}(n-1)\Delta_M \tag{26.5}$$

2. Input the encoded priority code word $c(n) = k$ for $x(n)$, in which case the priority of $x(n)$ occurs in stage k. Two cases are considered.

Case 1: if $\hat{x}_k(n-1) = 1$, then $\hat{x}_j(n) = \hat{x}_j(n-1)$ for $1 \leq j < k$, $\hat{x}_k(n) = 0$ and $\hat{x}_j(n) = 1$ for $k < j < \mathrm{M}$. In this case, $\hat{x}(n) = \sum_{j=1}^{k-1} \hat{x}_j(n-1)\Delta_j + \sum_{j=k+1}^{M} \Delta_j$.
Case 2: if $\hat{x}_k(n-1) = 0$, then $\hat{x}_j(n) = \hat{x}_j(n-1)$ for $1 \leq j < k$, $\hat{x}_k(n) = 1$ and $\hat{x}_j(n) = 0$ for $k < j < M$. In this case, $\hat{x}(n) = \sum_{j=1}^{k-1} \hat{x}_j(n-1)\Delta_j + \Delta_k$.

To apply MPCM to SSC we consider the spectrum of a signature vector $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ as a 1D signature vector with r_l being the spectral value of the lth band. Let $\Delta(\mathbf{r}) = \max_l\{r_l\} - \min_l\{r_l\}$. The number of stages, M is then obtained by

$$\mathrm{M} = [\log_2 \Delta(\mathbf{r})] + 1 \tag{26.6}$$

with $[x]$ defined by the largest integer less than or equal to x. So, the stage levels $\{\Delta_k\}_{k=1}^{\mathrm{M}}$ used in MPCM is defined by

$$\Delta_k(\mathbf{r}) = 2^{-k}\Delta(\mathbf{r}) \text{ for } k = 1, 2, \ldots, \mathrm{M}. \tag{26.7}$$

To demonstrate the utility of MPCM in SSC, two examples are provided for illustration. The first example shows a progressive MPCM-encoded signal of a one-dimensional gas spectral data, **r**, methyl salicylate in Figure 1.10 obtained from the website: webbook.nist.gov/chemistry and shown in Figure 26.5 and has 880 bands of spectral coverage 450–3966/cm.

For MPCM to operate on this signal, the number of stages required for MPCM encoding is calculated by (26.6) to be $M = 13$ stages. Since there are 13 stages, the stage levels obtained by (26.7) are $\Delta_k = \Delta(\mathbf{r})/2^k$ for $k = 1, 2, \ldots, 13$. With the initial condition assumed to be $x(0) = 0$ Figure 26.6 shows a graphical plot of the priority code words $c(n)$ for each of signal points $x(n)$ in Figure 26.5 produced by MPCM encoding algorithm with the x-axis and y-axis specified by signal points and their corresponding priority code words ranging from 1 to 13.

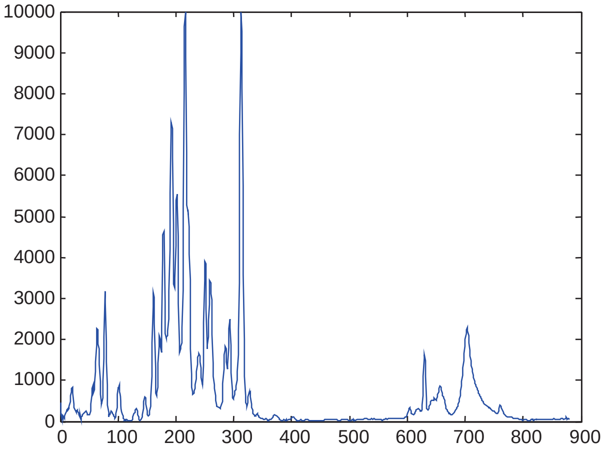


Figure 26.5 Spectral signature of methyl salicylate, r.

Using MPCM encoded priority code words provided by Figure 26.6, a 13-stage progressive signal components of the original signal in Figure 26.5 can be decomposed stage-by-stage in Figure 26.7.

As we can see from Figure 26.7, the MPCM encoding algorithm starts with the largest stage level, $\Delta_1 = \left(2^{13}/2\right)\left(\Delta(\mathbf{r})/2^{13}\right) = 4096 \times \left(\Delta(\mathbf{r})/2^{13}\right)$ in stage 1, then begins to reduce stage levels by half stage-by-stage to refine signal samples until it reaches the last stage that is stage 13 specified by stage level $\Delta_{13} = \Delta(\mathbf{r})/2^{13}$.

To decode the signal of methyl salicylate, the MPCM-encoded priority code words in Figure 26.6 is used as inputs and Figure 26.8 shows the 13 decoded signal components of methyl salicylate progressively stage by stage for signature reconstruction along with the approximation error $\varepsilon_{13}(n)$.

Since it may not be trivial to fully understand how MPCM works, the second example is provided by Table 26.1 for an illustrative purpose. It takes the first 20 signal points in Figure 26.5 to

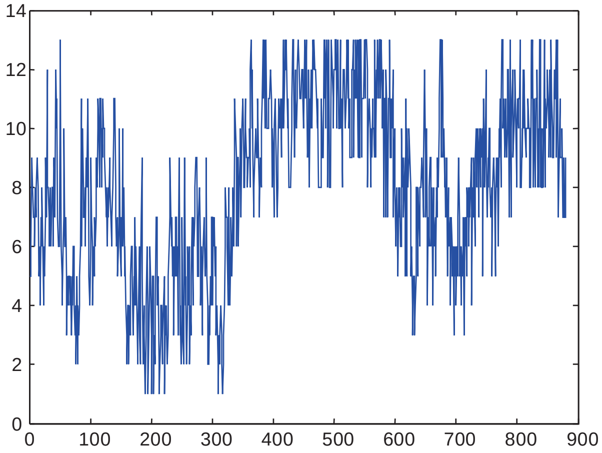
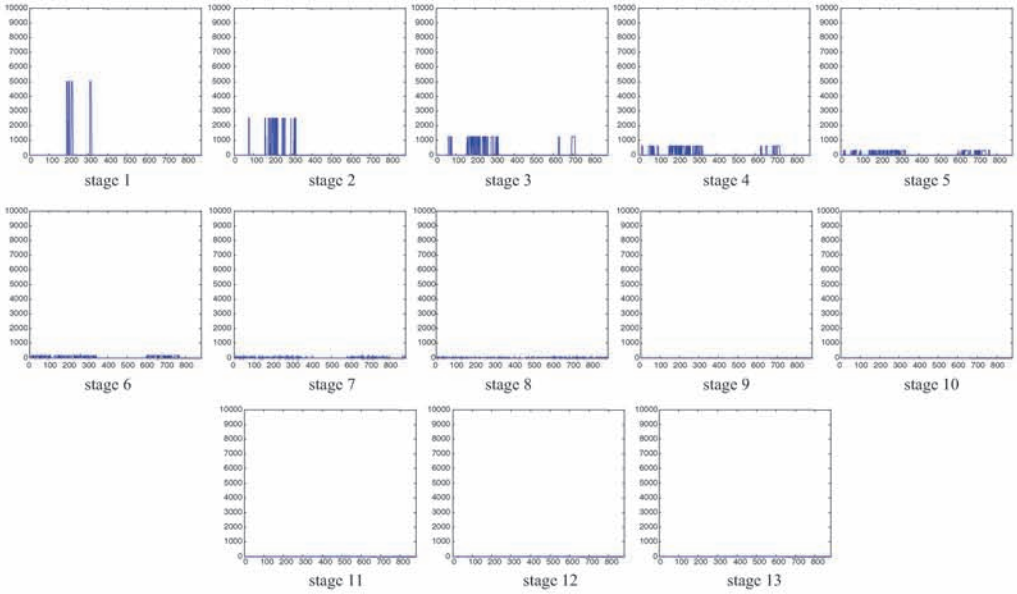


Figure 26.6 Graphical plot of priority code words for the signal of methyl salicylate in Figure 26. 5.

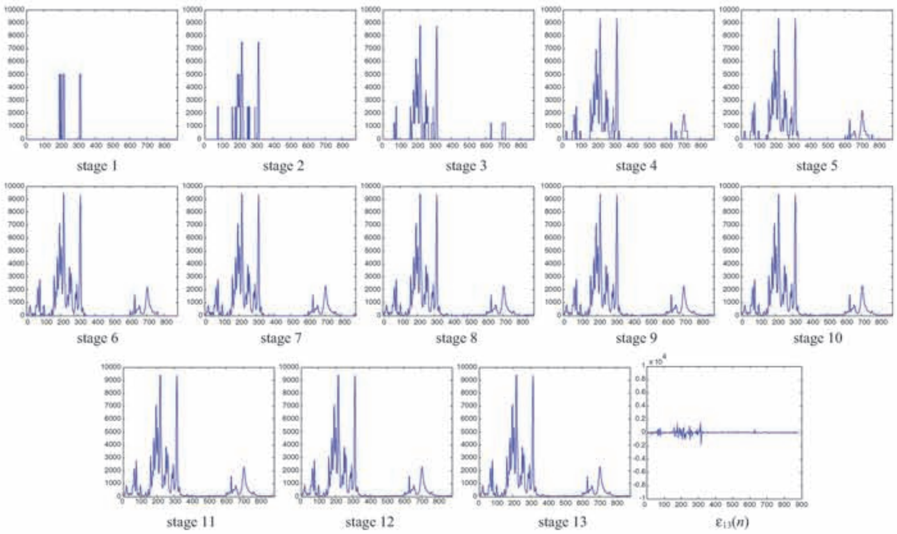


stage 1 stage 2 stage 3 stage 4 stage 5

stage 6 stage 7 stage 8 stage 9 stage 10

stage 11 stage 12 stage 13

Figure 26.7 MPCM-encoded progressive spectral signatures of methyl salicylate.



stage 1 stage 2 stage 3 stage 4 stage 5

stage 6 stage 7 stage 8 stage 9 stage 10

stage 11 stage 12 stage 13 $\varepsilon_{13}(n)$

Figure 26.8 A progressive stage-by stage decoded spectral signatures of methyl salicylate from the priority code words in Figure 6 along with the approximation error $\varepsilon_{13}(n)$.

Table 26.1 The first 20 MPCM-encoded signal samples in Figure 26.5 using 13 stages for signal decomposition

MPCM encoding algorithm																	
Input		Prediction		Signal components													Output
n	$x(n)$	$\hat{x}(n)$	$\varepsilon(n)$	$\hat{x}_1$	$\hat{x}_2$	$\hat{x}_3$	$\hat{x}_4$	$\hat{x}_5$	$\hat{x}_6$	$\hat{x}_7$	$\hat{x}_8$	$\hat{x}_9$	$\hat{x}_{10}$	$\hat{x}_{11}$	$\hat{x}_{12}$	$\hat{x}_{13}$	$c(n)$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	367	256	111	0	0	0	0	1	0	0	0	0	0	0	0	0	5
2	144	255	−111	0	0	0	0	0	1	1	1	1	1	1	1	1	5
3	33	127	−94	0	0	0	0	0	0	1	1	1	1	1	1	1	6
4	108	111	−3	0	0	0	0	0	0	1	1	0	1	1	1	1	9
5	70	95	−25	0	0	0	0	0	0	1	0	1	1	1	1	1	8
6	106	96	10	0	0	0	0	0	0	1	1	0	0	0	0	0	8
7	59	63	−4	0	0	0	0	0	0	0	1	1	1	1	1	1	7
8	119	64	55	0	0	0	0	0	0	1	0	0	0	0	0	0	7
9	157	128	29	0	0	0	0	0	1	0	0	0	0	0	0	0	6
10	162	160	2	0	0	0	0	0	1	0	1	0	0	0	0	0	8
11	198	192	6	0	0	0	0	0	1	1	0	0	0	0	0	0	7
12	233	224	9	0	0	0	0	0	1	1	1	0	0	0	0	0	8
13	240	240	0	0	0	0	0	0	1	1	1	1	0	0	0	0	9
14	223	223	0	0	0	0	0	0	1	1	0	1	1	1	1	1	8
15	245	224	21	0	0	0	0	0	1	1	1	0	0	0	0	0	8
16	290	256	34	0	0	0	0	1	0	0	0	0	0	0	0	0	5
17	388	384	4	0	0	0	0	1	1	0	0	0	0	0	0	0	6
18	516	512	4	0	0	0	1	0	0	0	0	0	0	0	0	0	4
19	591	576	15	0	0	0	1	0	0	1	0	0	0	0	0	0	7
20	665	640	25	0	0	0	1	0	1	0	0	0	0	0	0	0	6

Table 26.2 The first 20 MPCM decoded signal points for signal reconstruction in Figure 26.5 with 13 stages

MPCM decoding algorithm															
Input		Signal components													Output
n	$c(n)$	$\hat{x}_1$	$\hat{x}_2$	$\hat{x}_3$	$\hat{x}_4$	$\hat{x}_5$	$\hat{x}_6$	$\hat{x}_7$	$\hat{x}_8$	$\hat{x}_9$	$\hat{x}_{10}$	$\hat{x}_{11}$	$\hat{x}_{12}$	$\hat{x}_{13}$	$\hat{x}(n)$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	5	0	0	0	0	1	0	0	0	0	0	0	0	0	256
2	5	0	0	0	0	0	1	1	1	1	1	1	1	1	255
3	6	0	0	0	0	0	0	1	1	1	1	1	1	1	127
4	9	0	0	0	0	0	0	1	1	0	1	1	1	1	111
5	8	0	0	0	0	0	0	1	0	1	1	1	1	1	95
6	8	0	0	0	0	0	0	1	1	0	0	0	0	0	96
7	7	0	0	0	0	0	0	0	1	1	1	1	1	1	63
8	7	0	0	0	0	0	0	1	0	0	0	0	0	0	64
9	6	0	0	0	0	0	1	0	0	0	0	0	0	0	128
10	8	0	0	0	0	0	1	0	1	0	0	0	0	0	160
11	7	0	0	0	0	0	1	1	0	0	0	0	0	0	192
12	8	0	0	0	0	0	1	1	1	0	0	0	0	0	224
13	9	0	0	0	0	0	1	1	1	1	0	0	0	0	240
14	8	0	0	0	0	0	1	1	0	1	1	1	1	1	223
15	8	0	0	0	0	0	1	1	1	0	0	0	0	0	224
16	5	0	0	0	0	1	0	0	0	0	0	0	0	0	256
17	6	0	0	0	0	1	1	0	0	0	0	0	0	0	384
18	4	0	0	0	1	0	0	0	0	0	0	0	0	0	512
19	7	0	0	0	1	0	0	1	0	0	0	0	0	0	576
20	6	0	0	0	1	0	1	0	0	0	0	0	0	0	640

walk through detailed stage-by-stage implementations of the MPCM encoding and decoding algorithms. In Table 26.1, the first column lists the inputs specified by the first 20 signal values $\{x(n)\}_{n=1}^{20}$ with the initial condition specified by $x(0) = 0$. The second column lists the values of predicted $\hat{x}(n)$ and predicted error $\varepsilon(n)$. The third column lists all predicted values of signal components in 13 stages with stage levels specified by the largest stage level, $\Delta_1 = 2^{12} = 4096$ down to the smallest stage level, $\Delta_{13} = 2^0 = 1$. Finally, the last column produces the priority code words $\{c(n)\}_{n=1}^{20}$ for the first 20 signal points, $\{x(n)\}_{n=1}^{20}$.

Table 26.2 provides a state-by-stage decoding process for signal reconstruction of the 20 MPCM signal samples encoded in Table 26.1 where the first column takes the priority code words from the output in Table 26.1 as the input to MPCM decoder to decode the signal components in all the 13 stages in the second column. Finally, the last column of Table 26.2 outputs the predicted values of all the first 20 signal points of $\{x(n)\}_{n=1}^{20}$.

26.3 MPCM-Based Progressive Spectral Signature Coding

As recalled in the MPCM encoding algorithm, a signature vector $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ will be considered as a 1-D spectral signature where r_l is represented by one of the priority code words $\{c_k(\mathbf{r})\}_{k=1}^{\mathrm{M}}$ taking values in $\{1, 2, \ldots, \mathrm{M}\}$. For example, $c_l(\mathbf{r})$ indicates the priority of r_l in MPCM encoding and decoding. The smaller the number $c_l(\mathbf{r})$ is, the higher priority the $r_l(\mathbf{r})$ is for spectral encoding and decoding.

Next, we can further construct an M-dimensional priority unit vector associated with the priority code word $c_l(\mathbf{r})$ for MPCM-PSC as follows:

$$\mathbf{c}_l(\mathbf{r}) = (c_{l1}(\mathbf{r}), c_{l2}(\mathbf{r}), \ldots, c_{l\mathrm{M}}(\mathbf{r}))^T \tag{26.8}$$

with $c_{lk}(\mathbf{r}) \in \{0,1\}$ and $\sum_{k=1}^{M} c_{lk}(\mathbf{r}) = 1$. The condition that $\sum_{k=1}^{M} c_{lk}(\mathbf{r}) = 1$ in (26.8) implies that $c_l(\mathbf{r})$ has only one "1" in its component and all zeros in its remaining components. It should be noted that the priority code word $c_l(\mathbf{r})$ takes the value in $\{1,2,\ldots,M\}$. Instead of using the priority code word $c_l(\mathbf{r})$ itself, we use its corresponding M-dimensional priority unit vector $\mathbf{c}_l(\mathbf{r})$ defined by (26.8) where the boldface of $c_l(\mathbf{r})$, $\mathbf{c}_l(\mathbf{r})$ indicates that it is the priority unit vector of the original scalar priority code word $c_l(\mathbf{r})$. As an example, for $M=8$, $c_l(\mathbf{r})$ can take any of eight values, 1, 2, 3, 4, 5, 6, 7, 8. In this case, the following eight-dimensional priority unit vectors derived from (26.8) can be used for spectral signature coding:

$$\begin{array}{ll} c_l(\mathbf{r}) = 3 \Leftrightarrow \mathbf{c}_l(\mathbf{r}) = (0,0,1,0,0,0,0,0); & c_l(\mathbf{r}) = 4 \Leftrightarrow \mathbf{c}_l(\mathbf{r}) = (0,0,0,1,0,0,0,0) \\ c_l(\mathbf{r}) = 5 \Leftrightarrow \mathbf{c}_l(\mathbf{r}) = (0,0,0,0,1,0,0,0); & c_l(\mathbf{r}) = 6 \Leftrightarrow \mathbf{c}_l(\mathbf{r}) = (0,0,0,0,0,1,0,0) \\ c_l(\mathbf{r}) = 7 \Leftrightarrow \mathbf{c}_l(\mathbf{r}) = (0,0,0,0,0,0,1,0); & c_l(\mathbf{r}) = 8 \Leftrightarrow \mathbf{c}_l(\mathbf{r}) = (0,0,0,0,0,0,0,1) \end{array} \tag{26.9}$$

More specifically, if the priority code word $c_l(\mathbf{r})$ resulting from r_l is the priority, k, its M-dimensional priority unit vector $\mathbf{c}_l(\mathbf{r})$ is then specified by

$$\mathbf{c}_l(\mathbf{r}) = \Big(\underbrace{0}_{1}, \underbrace{0}_{2}, \ldots, \underbrace{0}_{k-1}, \underbrace{1}_{k}, \underbrace{0}_{k+1}, \ldots, \underbrace{0}_{M-1}, \underbrace{0}_{M}\Big)^T \tag{26.10}$$

where only one "1" occurs in the kth component and represents its priority specified by the kth stage. The advantage of using the M-dimensional priority unit vector, the position of "one" in (26.10) indicates the significance of its priority in the same manner that the bit position indicates the precision of the bit in a binary representation. Most importantly, we can use (26.10) and the Hamming distance to define a distance measure between two signature vectors $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ and $\mathbf{s} = (s_1, s_2, \ldots, s_L)^T$ at the kth stage via their corresponding M-dimensional priority unit vectors, $\mathbf{c}_l(\mathbf{r})$ and $\mathbf{c}_l(\mathbf{s})$ by

$$D_k(\mathbf{r},\mathbf{s}) = \sum\nolimits_{l=1}^{L} (c_{lk}(\mathbf{r}) \oplus c_{lk}(\mathbf{s})) \tag{26.11}$$

26.3.1 Spectral Discrimination

By virtue of (26.11), the similarity between two signature vectors **r** and **s** can be measured progressively. In other words, two signature vectors **r** and **s** are first measured by (26.11) in stage 1 via a prescribed stage threshold, say τ_1. If the distance $D_1(\mathbf{r},\mathbf{s})$ is greater than τ_1, **r** and **s** will be declared to be distinct. Otherwise, the comparison between **r** and **s** is continued to proceed at stage 2. If the distance $D_2(\mathbf{r},\mathbf{s})$ is greater than a prescribed stage threshold τ_2, **r** and **s** will be considered to be distinct signatures. Otherwise, a further comparison between **r** and **s** is continued on at stage 3, etc. The implementation of MPCM-based progressive spectral coding for target discrimination can be summarized as follows.

MPCM-PSSC spectral discri mination algorithm

1. Let **r** and **s** be two spectral signature vectors to be discriminated.
2. Specify the number of stages, M, needed to be processed. If two signatures produce different stage numbers M_1 and M_2, M is chosen as the minimum of M_1 and M_2.

3. Determine the stage thresholds $\{\tau_k\}_{k=1}^{M}$ to be used for discrimination in each of M stages.
4. Apply MPCM to **r** and **s** to generate their priority code words as described in (26.8) and expressed by (26.10).
5. Use (26.11) to measure the similarity between **r** and **s** progressively. For each stage k, we calculate the distance $D_k(\mathbf{r},\mathbf{s})$ and compare it against the kth stage threshold, τ_k. If $D_k(\mathbf{r},\mathbf{s}) > \tau_k$, then two pixel vectors **r** and **s** are declared to be distinct, and the process is terminated. Otherwise, repeat the same procedure until it reaches the last stage M. In this case, we check if $D_M(\mathbf{r},\mathbf{s}) > \tau_M$.
6. If $D_M(\mathbf{r},\mathbf{s}) > \tau_M$, then two pixel vectors **r** and **s** are declared to be distinct, and the process is terminated.
7. If $D_M(\mathbf{r},\mathbf{s}) \leq \tau_M$, then the process is also terminated and output "no discrimination," which declares **r** and **s** to be the same signature.

A key issue in implementing the above MPCM-PSSC discrimination algorithm is determination of an appropriate set of M-stage thresholds for a signature vector. In doing so a simulated white Gaussian noise is added to the signature vector to achieve a certain level of signal-to-noise ratio (SNR). This SNR is determined by how much sensitivity we would like to have for a signature vector responding to its spectral variations.

26.3.2 Spectral Identification

The spectral identification studied in this section is different from spectral discrimination in Section 26.3.1. While spectral discrimination only discriminates one signature vector from another without performing any additional task such as detection, classification and identification, spectral identification uses a given database (spectral library) Δ to identify an unknown signature vector **t**, referred to as target signature vector. Unlike spectral discrimination, the proposed spectral identification does not require stage thresholds.

MPCM-PSSC spectral identification algorithm 1

1. Let Δ be a given database (spectral library) that is made up of p spectral signatures, $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p$, that is, $\Delta = \{\mathbf{s}_h\}_{h=1}^{p}$ and **t** be target spectral signature vector to be identified via the database Δ.
2. Specify the number of stages, M via (26.6), needed to be processed. For each signature $\mathbf{s}_h$, let M_h be the associated stage number. M is chosen as the minimum among $M_1, M_2, \ldots, M_p$.
3. Determine stage thresholds for all M stages, $\{\tau_k\}_{k=1}^{M}$ for $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p$.
4. Apply MPCM to the target signature **t** to generate its priority code. Set $k = 1$.
5. At the kth stage, calculate the distance between **t** and $\mathbf{s}_h$, $D_k(\mathbf{t},\mathbf{s}_h)$ at stage k for $1 \leq h \leq p$ using (26.11). The **t** is identified by $\mathbf{s}_{h^*}$ with $h^* = \min_{1 \leq h \leq p}\{D_k(\mathbf{t},\mathbf{s}_h)\}$. If there is a tie, then the process is continued with those signatures that yield $\min_{1 \leq h \leq p}\{D_k(\mathbf{t},\mathbf{s}_h)\}$ and continue.
6. If $k < M$, then let $k \leftarrow k+1$ and go to step 5. Otherwise, continue.
7. In this case, we reach the last stage M. The **t** is identified by $\mathbf{s}_{h^*}$ with $h^* = \min_{1 \leq h \leq p}\{D_M(\mathbf{t},\mathbf{s}_h)\}$. If there is a tie at this final stage, the algorithm declares either "no match" or identifies **t** as one of tied signatures.

The steps 5–7 in the above algorithm calculates the distance between **t** and $\mathbf{s}_h$, $D_k(\mathbf{t},\mathbf{s}_h)$ for each $1 \leq h \leq p$ stage by stage and makes progressive decisions to determine if there is a match between **t** and $\mathbf{s}_{h^*}$ for some h^*. There is no need of implementing stage thresholds as the way carried out by spectral discrimination. As an alternative, we can also replace the steps 5–7 to derive a second version of MPCM-PSSC spectral identification that postpones the decision until the last stage M

by calculating the sum of stage distances between $\mathbf{t}$ and $\mathbf{s}_h$ in all M stages. In this case, the identification is to find the signature vector that yields the smallest sum.

MPCM-PSSC spectral identification algorithm 2

1–4. The same first four steps used in MPCM-PSSC target identification algorithm 1.
5′. Compute $\text{SUM}_h = \sum_{k=1}^{\text{M}} D_k(\mathbf{t}, \mathbf{s}_h)$ and identify $\mathbf{t}$ by $\mathbf{s}_{h*}$ with $h^* = \arg\{\min_{1 \leq h \leq p}\{\text{SUM}_h\}\}$, the signature that yields the smallest SUM_h. If there is a tie at this final stage, then the algorithm declares either “no match” or identifies $\mathbf{t}$ as one of tied signatures.

It should be noted that step 5′ does not make its decision progressively. Instead, it makes its decision at the final stage, stage M, on the sum of all stage distances. Nevertheless, it does take advantage of progressive spectral signature changes occurred at each stage, each of which contributes its change to the sum.

26.4 NIST-GAS Data Experiments

The ability of MPCM-PSSC in progressive signature decomposition and progressive signature reconstruction are demonstrated in Figures 26.7 and 26.8 and Tables 26.1 and 26.2. This and the following sections further demonstrate versatility of MPCM-PSSC in other applications, spectral discrimination and identification. Two sets of data are used for experiments, laboratory data and real hyperspectral images. The laboratory data to be used in this section are gas spectral data shown in Figure 1.10 available online at National Institute of Standards Technology (NIST)’s website (webbook.nist.gov/chemistry) The data set has five 880-band chemical/biological spectral signatures shown in Figure 26.9, which are methyl salicylate, pentanedione, propanoic acid, thiodiglycol, thriethyl phosphate, and heptanol. Since the selected data set for experiments was empirical and all the experiments conducted for this data set could be also applied to other data sets.

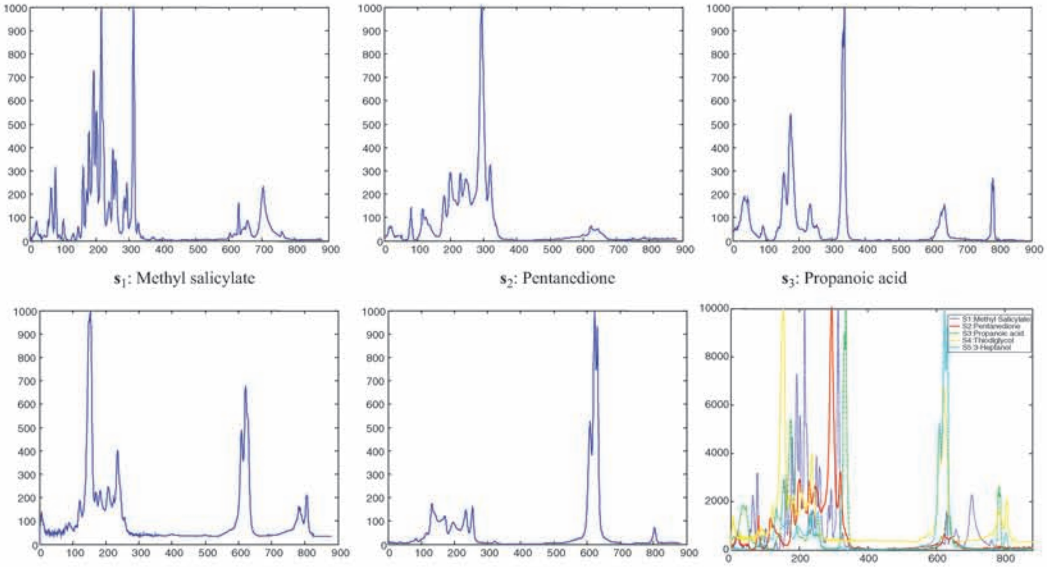
There are two reasons to select this data set. One is to demonstrate that MPCM-PSSC has an application in chemical/biological defense. The other is to demonstrate that MPCM-PSSC can be also used for ultraspectral signature characterization with thousands of spectral channels. There are also some other applications such as hyperspectral laboratory data experiments that can be found in Chang et al. (2003).

EXAMPLE 26.1

(spectral discrimination)

To perform spectral discrimination using MPCM-PSSC, we needed to determine appropriate thresholds for each stage that are implemented by MPCM-PSSC stage by stage. For each signature we created a noise-corrupted signature with SNR 30:1 where the SNR is defined in Harsanyi and Chang (1994) as the ratio of 50% reflectance to noise standard deviation. Using the methyl salicylate in Figure 26.5 as an example, the spectral signature represented by the methyl salicylate is denoted by $\mathbf{r} = (r_1, r_2, \ldots, r_{880})^T$. Then a noise corrupted methyl salicylate signature denoted by $\tilde{\mathbf{r}} = (\tilde{r}_1, \tilde{r}_2, \ldots, \tilde{r}_{880})^T$ was obtained by adding a white Gaussian noise to each band to achieve the SNR = 30:1. Finally, MPCM was applied to both the pure methyl salicylate signature with no noise and the 30:1 SNR noise corrupted methyl salicylate signature to obtain their respective MPCM priority code word for band l, $\mathbf{c}_l = (c_{l1}, c_{l2}, \ldots, c_{lM})^T$ and $\tilde{\mathbf{c}}_l = (\tilde{c}_{l1}, \tilde{c}_{l2}, \ldots, \tilde{c}_{lM})^T$ with $M = 13$. Then the kth stage threshold τ_k was obtained by

$$\tau_k = \sum_{l=1}^{880} c_{lk} \oplus \tilde{c}_{lk} \tag{26.12}$$



$\mathbf{s}_1$: Methyl salicylate

$\mathbf{s}_2$: Pentanedione

$\mathbf{s}_3$: Propanoic acid

$\mathbf{s}_4$: Thiodiglycol

$\mathbf{s}_5$: 3-Heptanol

5 signatures

Figure 26.9 Five spectral signatures of chemical data from NIST.

Table 26.3 13 stage thresholds for five signature vectors in Figure 26.9 with SNR 30:1

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13
$\mathbf{s}_1$	2	3	7	16	23	28	40	47	58	58	54	48	47
$\mathbf{s}_2$	1	9	6	19	29	40	61	61	83	96	94	83	81
$\mathbf{s}_3$	2	3	6	10	21	27	38	49	50	57	70	66	64
$\mathbf{s}_4$	1	2	9	26	27	52	92	110	150	150	140	80	130
$\mathbf{s}_5$	0	0	1	5	8	18	20	30	37	55	55	58	87

Table 26.3 tabulates all the stage thresholds $\{\tau_k\}_{k=1}^{\mathrm{M}}$ for each of five signature vectors, methyl salicylate, pentanedione, propanoic acid, thiodiglycol, thriethyl phosphate, and heptanol denoted by $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$, $\mathbf{s}_4$, and $\mathbf{s}_5$.

It should be noted that the total number of stages, $M = 13$ was determined by (26.6). As long as $\{\tau_k\}_{k=1}^{M}$ were determined, the discrimination process started with the stage threshold in stage 1. If the distance between two signatures in stage 1 was greater than the threshold, the two signatures were declared to be distinct and discrimination process is terminated. Otherwise, it implied that two signatures could not be discriminated in stage 1 and the discrimination process was then passed on to stage 2 where the distance between two signatures in stage 2 was calculated and compared to the threshold at stage 2. If the distance at stage 2 was greater than the threshold, the process was terminated. Otherwise, the same procedure was repeated again until the last stage was reached.

Since the stage thresholds produced by one signature vector generally were different from those produced by another signature vectors, the discrimination threshold was then determined by the minimum of the two different stage thresholds, that is, $\min\{\tau_i(\text{signature 1}), \tau_i(\text{signature 2})\}$. Table 26.4 shows the results where the stage thresholds in Table 26.3 were used for discrimination and all the five signature vectors could be discriminated in stage 1.

EXAMPLE 26.2

(spectral identification)

In this example, we further demonstrate the utility of MPCM-PSSC in spectral identification via a database (spectral library) Δ that consisted of the five signature vectors in Figure 26.9. For each target signature vector $\mathbf{t}$, 60% abundance fraction was simulated while the other four signature vectors sharing the remaining 40% abundance fraction with each of 10% abundance fraction. Five different admixtures were generated by a fixed

Table 26.4 Discrimination among five signature vectors in Figure 26.9 using the stage thresholds in Table 26.3

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13
$\mathbf{s}_1 - \mathbf{s}_2$	10	32	49	80	94	120	170	190	190	200	160	110	120
$\mathbf{s}_1 - \mathbf{s}_3$	12	32	53	72	100	120	150	180	180	180	190	120	140
$\mathbf{s}_1 - \mathbf{s}_4$	12	30	57	98	100	150	200	160	180	170	150	87	150
$\mathbf{s}_1 - \mathbf{s}_5$	10	28	48	69	88	120	140	160	180	180	170	130	260
$\mathbf{s}_2 - \mathbf{s}_3$	6	22	36	54	88	110	140	160	180	200	180	140	170
$\mathbf{s}_2 - \mathbf{s}_4$	6	20	42	78	82	140	180	160	190	190	160	110	170
$\mathbf{s}_2 - \mathbf{s}_5$	4	18	27	45	66	100	140	130	180	210	160	130	280
$\mathbf{s}_3 - \mathbf{s}_4$	8	16	44	78	88	130	160	170	160	180	170	130	170
$\mathbf{s}_3 - \mathbf{s}_5$	6	14	35	39	80	98	96	140	160	170	180	170	270
$\mathbf{s}_4 - \mathbf{s}_5$	6	14	39	67	76	120	150	130	170	180	160	110	280

Table 26.5 Spectral identification for a mixed signature, **s** with $t = \mathbf{s}_1$

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{s}_1$	6	24	46	80	110	120	140	170	140	130	130	70	68	1234
$\mathbf{s}_2$	12	34	47	88	100	130	170	170	190	190	150	99	120	1500
$\mathbf{s}_3$	14	34	55	92	110	130	150	170	170	170	180	130	150	1555
$\mathbf{s}_4$	14	30	61	110	110	170	200	160	170	160	140	79	160	1564
$\mathbf{s}_5$	12	30	46	77	95	130	140	150	180	170	160	120	270	1580

Table 26.6 Spectral identification for a mixed signature, $\mathbf{s} = 0.1\mathbf{s}_1 + 0.6\mathbf{t} + 0.1\mathbf{s}_3 + 0.1\mathbf{s}_4 + 0.1\mathbf{s}_5$ with $\mathbf{t} = \mathbf{s}_2$

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{s}_1$	10	32	53	87	120	150	180	190	170	170	160	96	98	1516
$\mathbf{s}_2$	0	16	22	59	98	130	140	160	150	190	150	110	100	1325
$\mathbf{s}_3$	6	20	42	65	100	130	150	190	160	180	170	130	160	1503
$\mathbf{s}_4$	6	18	48	83	110	160	190	180	170	160	150	95	150	1520
$\mathbf{s}_5$	4	16	31	54	96	120	130	150	170	180	170	120	290	1531

mixing composition (0.6, 0.1, 0.1, 0.1, 0.1) of the five signature vectors. When one of $\mathbf{s}_1$, $\mathbf{s}_2$, $\mathbf{s}_3$, $\mathbf{s}_4$, and $\mathbf{s}_5$ was designated as a target signature vector, say $\mathbf{s}_1$, a mixed signature vector **s** is then generated by mixing 0.6 of $\mathbf{s}_1$ with abundance fraction of 0.1 from each of the other four signature vectors $\mathbf{s}_2$, $\mathbf{s}_3$, $\mathbf{s}_4$, and $\mathbf{s}_5$. Table 26.5 tabulates a progressive spectral identification process for such a mixed signature vector **s** where the signature vector **s** was quickly identified by the target signature vector $\mathbf{t} = \mathbf{s}_1$ immediately by Algorithm 1 in the first stage as well as by Algorithm 2 correctly.

Similar experiments were also performed by changing the designated target signature vector **t** from $\mathbf{s}_1$ to $\mathbf{s}_2$, $\mathbf{s}_3$, $\mathbf{s}_4$, and $\mathbf{s}_5$ for two spectral identification algorithms. Tables 26.6–26.9 tabulate their respective spectral progressive identification results. All the four mixed signature vectors were correctly identified by both Algorithm 1 and Algorithm 2.

The above experiment indicates that $\mathbf{s}_4$ as $\mathbf{s}_5$ were very similar to each other in terms of spectral variation. Algorithm 1 has difficulty with identification until stage 2.

As a concluding remark, the abundance fraction of the target signature vector **t** has impact on the performance of MPCM-PSSC in identification. If the abundance fraction was greater than 60%, MPCM-PSSC improved significantly its performance. Otherwise, its performance deteriorated as the abundance fraction gradually diminished. In the following real image experiments, we will further demonstrate that MPCM-PSSC can still perform effectively when the estimated abundance fraction of a subpixel target is above 40%.

Table 26.7 Spectral identification for a mixed signature, $\mathbf{s} = 0.1\mathbf{s}_1 + 0.1\mathbf{s}_2 + 0.6\mathbf{t} + 0.1\mathbf{s}_4 + 0.1\mathbf{s}_5$ with $\mathbf{t} = \mathbf{s}_3$

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{s}_1$	14	34	57	93	100	160	160	180	170	170	140	110	100	1488
$\mathbf{s}_2$	8	30	40	79	92	120	170	160	200	190	160	130	130	1509
$\mathbf{s}_3$	6	18	24	59	84	130	130	150	160	160	150	140	120	1331
$\mathbf{s}_4$	10	24	46	95	92	150	180	160	170	170	140	110	150	1497
$\mathbf{s}_5$	8	24	39	58	78	120	130	140	170	180	140	140	260	1487

Table 26.8 Spectral identification for a mixed signature, $\mathbf{s} = 0.1\mathbf{s}_1 + 0.1\mathbf{s}_2 + 0.1\mathbf{s}_3 + 0.6\mathbf{t} + 0.1\mathbf{s}_5$ with $\mathbf{t} = \mathbf{s}_4$

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{s}_1$	12	34	55	110	130	140	190	180	180	150	150	95	130	1556
$\mathbf{s}_2$	6	24	36	83	110	130	190	160	190	190	160	110	130	1519
$\mathbf{s}_3$	8	22	40	85	110	130	160	160	170	160	170	120	170	1505
$\mathbf{s}_4$	6	14	42	87	110	150	170	130	150	140	130	72	96	1297
$\mathbf{s}_5$	6	20	31	74	110	110	140	130	150	170	160	130	270	1501

Table 26.9 Spectral identification for a mixed signature, $\mathbf{s} = 0.1\mathbf{s}_1 + 0.1\mathbf{s}_2 + 0.1\mathbf{s}_3 + 0.1\mathbf{s}_4 + 0.6\mathbf{t}$ with $\mathbf{t} = \mathbf{s}_5$

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{s}_1$	10	26	54	72	120	150	180	170	190	160	160	120	140	1552
$\mathbf{s}_2$	4	20	35	56	92	130	180	160	180	210	170	130	170	1537
$\mathbf{s}_3$	6	16	41	58	94	120	150	160	180	170	170	150	160	1475
$\mathbf{s}_4$	6	16	47	72	94	150	190	140	170	160	150	98	170	1463
$\mathbf{s}_5$	2	6	28	37	88	110	130	130	150	150	150	130	230	1341

26.5 Real Image Hyperspectral Experiments

The second data set used for experiments was the 15-panel HYDICE image shown in Figure 1.15 (a). Two scenarios were conducted for experiments based on this 15-panel HYDICE scene. One was spectral discrimination among the five panel signatures, $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$. The other was to identify the 15 panels unsupervisedly using only knowledge obtained directly from the data.

EXAMPLE 26.3

(spectral discrimination)

Like Example 26.1, spectral discrimination was performed by MPCM-PSSC where the number of stages required for MPCM-PSSC was calculated by (26.6) to be $M = 13$ and the stage levels $\{\Delta_k\}_{k=1}^{13}$ were obtained by (26.7). To implement MPCM-PSSC algorithm, we also needed to determine an appropriate set of stage thresholds.

Using the same way conducted in Example 26.1, the desired set of stage thresholds $\{\tau_k\}_{k=1}^{13}$ were obtained in Table 26.10 by (26.12) using noise-corrupted signatures with SNR set to 30:1 as variation of signature tolerance.

Table 26.10 Stage thresholds for five panel signatures with SNR 30:1

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13
$\mathbf{p}_1$	2	1	5	10	14	24	23	16	16	16	9	4	4
$\mathbf{p}_2$	2	2	14	9	13	23	24	15	13	11	10	4	4
$\mathbf{p}_3$	1	4	6	8	16	19	21	13	11	11	7	6	5
$\mathbf{p}_4$	3	6	3	11	15	21	23	20	16	12	7	7	3
$\mathbf{p}_5$	3	6	4	10	16	19	21	18	13	10	7	4	3

Table 26.11 Discrimination among five panel signature vectors using the stage thresholds in Table 26.10

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13
$\mathbf{p}_1$–$\mathbf{p}_2$	2	2	10	13	20	32	30	24	29	21	16	4	5
$\mathbf{p}_1$–$\mathbf{p}_3$	4	4	17	23	28	29	31	26	24	22	11	6	5
$\mathbf{p}_1$–$\mathbf{p}_4$	4	8	8	22	26	37	32	26	22	23	10	6	4
$\mathbf{p}_1$–$\mathbf{p}_5$	4	8	11	22	32	34	34	21	13	16	12	2	5
$\mathbf{p}_2$–$\mathbf{p}_3$	2	2	13	12	24	35	31	22	21	9	11	6	4
$\mathbf{p}_2$–$\mathbf{p}_4$	4	8	14	23	28	33	36	22	21	10	10	8	3
$\mathbf{p}_2$–$\mathbf{p}_5$	4	8	17	23	28	30	30	25	28	13	8	4	4
$\mathbf{p}_3$–$\mathbf{p}_4$	6	10	19	27	30	34	31	12	20	13	9	10	5
$\mathbf{p}_3$–$\mathbf{p}_5$	6	10	22	29	34	35	29	29	17	16	7	6	6
$\mathbf{p}_4$–$\mathbf{p}_5$	4	8	3	6	16	25	24	29	21	13	4	6	1

Table 26.11 tabulates discrimination results obtained by MPCM-PSSC among the five panel signature vectors $\{\mathbf{p}_i\}_{i=1}^{5}$ in Figure 1.16.

As shown in Table 26.11, $\mathbf{p}_1$ and $\mathbf{p}_2$ were more similar each other than other three panel signature vectors since the discrimination could be accomplished in stage 2 in terms of spectral variation compared to other signature discrimination that was already discriminated in stage 1.

EXAMPLE 26.4

(spectral identification)

The experiments conducted in this example were very interesting and offered several intriguing results and observations. It was designed to identify the 19 R panel pixels, p_{ij} in Figure 1.15(b) by MPCM-PSSC. Since the panel pixels p_{13}, p_{23}, p_{33}, p_{43}, p_{53} have size of 1 m × 1 m that is smaller than the pixel size, their abundance fractions present in single pixels can be at most $1/(1.56)^2 = 0.4109$ that can be interpreted as approximately 50% of the pixel size. As a result, the performance in identification of these subpixel panels can be expected to be very challenging and difficult. On the other hand, due to its very high spatial and spectral resolution the spectral variations of image pixels in this HYDICE scene can be very subtle and sensitive. Therefore, using the five panel signatures $\{\mathbf{p}_i\}_{i=1}^{5}$ in Figure 1.16 as a data base may not be appropriate. Instead, a more effective data base must be obtained in an unsupervised means directly form data. In doing so, the result of the 34 target pixels generated directly from the scene by an unsupervised fully constrained least squares (UFCLS) method developed in Heinz and Chang (2001) and Chang (2003a) were used to form a desired data base Δ. Among these 34 generated target pixels there were five panel pixels identified to correspond to the five distinct panel signatures $\{\mathbf{p}_i\}_{i=1}^{5}$. Table 26.12 tabulates the results produced by MPCM-PSSC using Algorithm 1 and Algorithm 2 for spectral identification along with the abundance fractions of the 19 R pixels estimated by FCLS where an identification error was highlighted by shade.

According to Table 26.12, Algorithm 1 yielded the best performance in the sense that it only missed identification when the panels pixel, p_{13}, p_{212}, p_{33}, p_{412}, p_{43}, p_{53} with estimated abundance fractions less than 0.3821. Algorithm 2 also made six identification errors, but it seemed that these misidentifications had no clear tie to the abundance fractions as Algorithm 1 did. For example, it correctly identifies p_{212} whose abundance was only 0.3141, but it misidentified the p_{32} whose abundance was 0.5343. Compared to Algorithm 2, SAM and SID not only made the same 6 identification errors as did Algorithm 2, but also made two more additional errors, which were panel pixels p_{511}, p_{52} with abundance fractions, 0.7203 and 0.7789.

This experiment shows that MPCM-PSSC performed more effectively than a pixel-based spectral similarity measure such as SAM and SID in Table 26.12. It should be noted that real target panel pixels in Table 26.12 are compared against the five panel signature vectors $\{\mathbf{p}_i\}_{i=1}^{5}$ for analysis.

It is interesting to note that if the five panel signature vectors $\{\mathbf{p}_i\}_{i=1}^{5}$ in Figure 1.16 were directly used for identification, the results are reported in Chang (2003a) and are not as good as the results in Table 26.12 that

Table 26.12 Identification of 19 R panel pixels in Figure 1.5(a)

Panel pixels	Algorithm 1	Algorithm 2	SAM/SID	Abundance fractions estimated by FCLS
p_{11}	$\mathbf{p}_1$	$\mathbf{p}_1$	$\mathbf{p}_1$	1
p_{12}	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_2$	0.4098
p_{13}	$\mathbf{p}_3$	$\mathbf{p}_2$	$\mathbf{p}_2$	0.0499
p_{211}	$\mathbf{p}_2$	$\mathbf{p}_2$	$\mathbf{p}_2$	0.5255
p_{221}	$\mathbf{p}_3$	$\mathbf{p}_2$	$\mathbf{p}_2$	0.3141
p_{22}	$\mathbf{p}_2$	$\mathbf{p}_2$	$\mathbf{p}_2$	0.6917
p_{23}	$\mathbf{p}_2$	$\mathbf{p}_2$	$\mathbf{p}_2$	0.4221
p_{311}	$\mathbf{p}_3$	$\mathbf{p}_3$	$\mathbf{p}_3$	0.8647
p_{312}	$\mathbf{p}_3$	$\mathbf{p}_3$	$\mathbf{p}_3$	1
p_{32}	$\mathbf{p}_3$	$\mathbf{p}_2$	$\mathbf{p}_2$	0.5343
p_{33}	$\mathbf{p}_2$	$\mathbf{p}_2$	$\mathbf{p}_2$	0.3285
p_{411}	$\mathbf{p}_4$	$\mathbf{p}_4$	$\mathbf{p}_4$	1
p_{412}	$\mathbf{p}_5$	$\mathbf{p}_4$	$\mathbf{p}_4$	0.3821
p_{42}	$\mathbf{p}_4$	$\mathbf{p}_4$	$\mathbf{p}_4$	0.7034
p_{43}	$\mathbf{p}_2$	$\mathbf{p}_2$	$\mathbf{p}_2$	0.2242
p_{511}	$\mathbf{p}_5$	$\mathbf{p}_5$	$\mathbf{p}_4$	0.7203
p_{521}	$\mathbf{p}_5$	$\mathbf{p}_5$	$\mathbf{p}_5$	1
p_{52}	$\mathbf{p}_5$	$\mathbf{p}_5$	$\mathbf{p}_4$	0.7789
p_{53}	$\mathbf{p}_2$	$\mathbf{p}_2$	$\mathbf{p}_2$	0.1466

were produced by using the real target panel pixels in Table 26.12. This is primarily due to the fact that the panel signature vectors $\{\mathbf{p}_i\}_{i=1}^{5}$ obtained by averaging R panel pixels are not real pixels. As a result, the signature variations of real target panel pixel vectors were compromised. MPCM-PSSC seemed to remedy such deficiency by capturing subtle spectral variations in multiple stages that were able to dictate changes in subtle difference encountered in real data as shown in Table 26.13.

As a final comment, it should be noted that the 34 target pixels used in this experiment were obtained according to Heinz and Chang (2001) that have shown to be sufficiently enough to represent the five distinct panel spectral signature vectors. However, it did not imply that it required at least 34 target pixel vectors to do

Table 26.13(a) Identification of p_{11}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\mathbf{p}_2$	2	6	12	28	27	35	39	31	20	12	4	3	7	226
$\mathbf{p}_3$	6	8	19	29	25	37	37	31	19	14	8	4	5	242
$\mathbf{p}_4$	4	12	4	18	27	29	39	34	21	10	8	3	1	210
$\mathbf{p}_5$	2	8	3	21	29	30	36	27	17	10	10	4	5	202

Table 26.13(b) Identification of p_{12}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	2	8	4	20	29	42	39	31	15	12	7	4	5	218
$\mathbf{p}_2$	4	2	10	14	18	29	22	22	17	18	9	3	10	178
$\mathbf{p}_3$	4	8	21	27	28	39	24	28	18	18	9	8	6	238
$\mathbf{p}_4$	4	10	8	20	28	39	36	27	22	14	13	3	4	228
$\mathbf{p}_5$	4	10	7	25	28	38	31	26	24	16	13	4	4	230

Table 26.13(c) Identification of p_{13}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	4	6	15	26	28	35	38	36	23	18	9	6	6	250
$\mathbf{p}_2$	6	0	19	12	13	32	29	21	15	18	9	5	5	184
$\mathbf{p}_3$	2	6	12	17	25	24	21	23	20	22	9	6	3	190
$\mathbf{p}_4$	6	8	19	24	29	40	33	26	14	12	9	5	5	230
$\mathbf{p}_5$	6	8	18	29	29	41	28	29	22	20	15	6	5	256

Table 26.13(d) Identification of p_{211}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	4	8	12	28	28	37	33	37	25	10	5	4	3	234
$\mathbf{p}_2$	2	2	6	6	11	26	24	14	13	10	3	3	6	126
$\mathbf{p}_3$	2	8	15	23	29	32	32	24	20	12	9	6	4	216
$\mathbf{p}_4$	4	10	14	28	27	38	36	27	16	4	7	3	4	218
$\mathbf{p}_5$	4	10	13	31	31	31	39	26	22	10	7	4	8	236

Table 26.13(e) Identification of p_{221}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	6	8	16	31	30	39	34	38	21	15	9	6	5	258
$\mathbf{p}_2$	4	2	10	9	15	32	21	15	13	15	9	3	6	154
$\mathbf{p}_3$	0	8	13	20	29	30	29	25	16	19	9	8	6	212
$\mathbf{p}_4$	6	10	18	27	29	42	31	30	14	11	11	3	4	236
$\mathbf{p}_5$	6	10	17	32	33	33	38	25	18	17	9	4	6	248

Table 26.13(f) Identification of p_{22}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	4	6	12	25	32	40	33	30	22	14	11	2	3	234
$\mathbf{p}_2$	2	0	10	9	13	27	30	17	22	14	11	1	6	162
$\mathbf{p}_3$	2	6	15	24	31	33	24	21	19	14	7	4	2	202
$\mathbf{p}_4$	4	8	14	27	31	33	36	28	21	12	13	1	2	230
$\mathbf{p}_5$	4	8	13	28	33	40	33	23	29	14	15	2	4	246

Table 26.13(g) Identification of p_{23}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	4	6	16	27	24	29	36	28	22	14	5	7	4	222
$\mathbf{p}_2$	2	0	16	13	17	28	29	17	20	16	5	6	7	176
$\mathbf{p}_3$	2	6	19	22	27	34	29	17	21	22	9	7	5	220
$\mathbf{p}_4$	4	8	18	29	31	44	39	24	19	10	9	6	3	244
$\mathbf{p}_5$	4	8	17	30	35	35	34	23	21	18	11	7	7	250

Table 26.13(h) Identification of p_{311}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	6	12	18	34	20	29	45	34	27	12	7	5	3	252
$\mathbf{p}_2$	4	6	16	24	21	30	38	21	19	12	7	4	8	210
$\mathbf{p}_3$	0	4	3	11	11	28	22	19	26	12	7	7	4	154
$\mathbf{p}_4$	6	12	20	30	23	42	36	22	20	10	9	4	4	238
$\mathbf{p}_5$	6	14	19	37	25	37	33	23	26	16	13	5	6	260

Table 26.13(i) Identification of p_{312}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	6	8	19	29	25	37	37	31	19	14	8	4	5	242
$\mathbf{p}_2$	4	6	19	23	24	32	28	22	19	16	8	5	6	212
$\mathbf{p}_3$	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\mathbf{p}_4$	6	10	19	27	24	42	32	29	16	16	12	5	4	242
$\mathbf{p}_5$	6	12	20	34	24	45	29	26	18	16	12	6	6	254

Table 26.13(j) Identification of p_{32}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	6	6	10	21	29	34	31	34	20	12	7	4	4	218
$\mathbf{p}_2$	6	0	16	11	12	23	24	13	16	14	7	5	11	158
$\mathbf{p}_3$	2	6	21	26	26	31	30	15	17	14	9	6	7	210
$\mathbf{p}_4$	6	8	14	25	28	35	38	28	21	14	9	5	5	236
$\mathbf{p}_5$	6	8	13	28	26	38	35	21	25	20	13	6	7	246

Table 26.13(k) Identification of p_{33}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	4	6	14	36	24	32	38	33	26	10	4	6	1	234
$\mathbf{p}_2$	4	0	12	20	17	39	25	24	16	16	4	5	8	190
$\mathbf{p}_3$	4	6	11	27	27	29	19	16	19	14	8	6	6	192
$\mathbf{p}_4$	4	8	16	38	29	45	35	27	13	10	8	5	2	240
$\mathbf{p}_5$	4	8	15	43	35	38	32	28	23	16	10	6	6	264

Table 26.13(l) Identification of p_{411}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	4	12	4	18	27	29	39	34	21	10	8	3	1	210
$\mathbf{p}_2$	4	8	14	28	26	38	38	29	19	10	8	0	6	228
$\mathbf{p}_3$	6	10	19	27	24	42	32	29	16	16	12	5	4	242
$\mathbf{p}_4$	0	0	0	0	0	0	0	0	0	0	0	0	0	0
$\mathbf{p}_5$	2	8	1	9	24	27	35	27	18	12	10	1	4	178

Table 26.13(m) Identification of p_{412}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	2	10	3	14	27	30	36	33	23	16	5	4	5	208
$\mathbf{p}_2$	2	6	13	26	28	39	33	26	17	16	3	3	10	222
$\mathbf{p}_3$	6	10	20	29	24	35	25	22	22	20	7	4	6	230
$\mathbf{p}_4$	2	6	1	10	18	21	29	23	22	14	7	3	4	160
$\mathbf{p}_5$	0	8	0	15	24	24	28	26	20	16	11	4	4	180

Table 26.13(n) Identification of p_{42}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	6	8	7	18	23	31	38	35	25	14	8	4	3	220
$\mathbf{p}_2$	6	8	13	26	24	34	39	26	17	16	8	3	6	226
$\mathbf{p}_3$	6	12	20	27	26	34	27	26	20	22	10	6	6	242
$\mathbf{p}_4$	4	12	5	10	24	26	25	13	14	14	8	3	2	160
$\mathbf{p}_5$	6	8	4	15	28	25	26	22	20	16	8	4	6	188

Table 26.13(o) Identification of p_{43}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	6	8	11	28	23	29	35	30	18	11	6	3	6	214
$\mathbf{p}_2$	4	2	15	22	24	36	32	19	16	17	6	2	9	204
$\mathbf{p}_3$	4	6	22	27	26	40	34	23	19	21	8	5	5	240
$\mathbf{p}_4$	4	6	7	16	30	36	42	22	19	13	10	2	5	212
$\mathbf{p}_5$	6	10	8	25	34	31	45	19	23	17	12	3	7	240

Table 26.13(p) Identification of p_{511}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	2	10	7	13	24	24	32	25	18	10	11	6	2	184
$\mathbf{p}_2$	2	6	13	25	31	41	37	24	22	12	11	3	7	234
$\mathbf{p}_3$	6	10	20	32	29	41	29	26	17	18	11	8	3	250
$\mathbf{p}_4$	4	6	5	13	19	19	31	31	17	14	13	3	1	176
$\mathbf{p}_5$	2	10	4	12	29	22	26	14	17	10	13	2	3	164

Table 26.13(q) Identification of p_{521}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	2	8	3	21	29	30	36	27	17	10	10	4	5	202
$\mathbf{p}_2$	2	8	13	31	30	39	41	22	19	16	10	1	10	242
$\mathbf{p}_3$	6	12	20	34	24	45	29	26	18	16	12	6	6	254
$\mathbf{p}_4$	2	8	1	9	24	27	35	27	18	12	10	1	4	178
$\mathbf{p}_5$	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table 26.13(r) Identification of p_{52}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	4	10	5	22	31	25	40	27	18	12	7	3	2	206
$\mathbf{p}_2$	4	8	11	26	30	40	45	24	20	14	5	2	7	236
$\mathbf{p}_3$	6	12	18	29	28	38	33	24	17	20	9	5	5	244
$\mathbf{p}_4$	4	8	3	14	20	28	25	25	17	12	9	2	1	168
$\mathbf{p}_5$	2	8	2	11	26	23	26	16	17	10	11	3	3	158

Table 26.13(s) Identification of p_{53}

Stage	1	2	3	4	5	6	7	8	9	10	11	12	13	SUM
$\mathbf{p}_1$	6	8	9	24	23	35	40	30	15	13	10	4	5	222
$\mathbf{p}_2$	4	2	13	20	26	38	29	17	13	17	10	3	10	202
$\mathbf{p}_3$	4	6	22	29	24	28	29	19	16	15	12	6	6	216
$\mathbf{p}_4$	4	6	7	16	28	38	37	24	12	11	8	3	4	198
$\mathbf{p}_5$	6	10	6	21	30	29	40	21	18	17	12	4	6	220

so. There may have some unsupervised target detection and classification algorithms such as those developed in Chapter 17 that can generate a fewer number of target pixel vectors than 34 but still include pixels that can represent all the desired five panel signature vectors. In this case, these generated target pixel vectors can be used as a database as well. As expected, the conclusion drawn from Table 26.13 will remain unchanged.

26.6 Conclusions

This chapter introduces a new concept of PSSC for hyperspectral signature characterization. It is derived from a technique called MPCM that was previously developed for progressive image reconstruction and edge detection. Unlike the commonly used SSC that performs coding with hard decision, PSSC characterizes a hyperspectral signature in a sequence of soft decisions in multiple stages to produce a spectral profile of progressive changes in spectral variation of a spectral signature vector. The idea of MPCM-based PSSC (MPCM-PSSC) is to use a sequence of soft decision-based quantizers to generate priority codes for a hyperspectral signature vector that can be used to prioritize signature values across its spectral range whose priorities are specified by stage levels implemented in various stages. Such priority codes allow users to decompose and reconstruct a hyperspectral signature vector progressively in accordance with the priorities assigned to spectral signature values specified by various wavelengths. As a result, a spectral profile of progressive changes in spectral variation can be generated for a hyperspectral signature vector and can be further used to dictate subtle differences in spectral characterization. To substantiate the utility of MPCM-PSSC in applications spectral discrimination and identification are used for illustration. Experiments are also conducted to demonstrate unique features of MPCM-PSSC in hyperspectral signature characterization such as progressive spectral changes, progressive signature decomposition, and progressive signature reconstruction that cannot be achieved by any spectral signature coding.

VII

Hyperspectral Signal Characterization

The hyperspectral signal coding in Part VI is designed to produce credible discrete versions of hyperspectral signals as fingerprints so that these fingerprints provide sufficient information of their own identities. But such signal coding does not necessarily tell you what a real signal looks like and how it behaves. In other words, a signal identity and its fingerprint is one-to-one correspondence relationship such as one-to-one identification between a person and his unique nickname where the nickname does not have to describe the person in detail. Using a more specific example for illustration we consider a set of signals, $\{\mathbf{s}_j\}_{j=1}^{p}$ to be used for data transmission. When one of these signals is selected for transmission, it is its subscript instead of the signal itself being transmitted. According to information theory, if a fixed length coding is used, only log$p$ bits required to derive a set of p code words corresponding to fingerprints of the p signals, $\{\mathbf{s}_j\}_{j=1}^{p}$ for signal transmission without actually transmitting these signals themselves. This is because a signal can be identified by its subscript through its code word used as its fingerprint. By means of these p code words signal coding is generally considered as discrete signal processing and used for hard-decision-made applications such as signal detection, discrimination, classification, and identification, but it certainly cannot be used for continuous signal processing-based applications, which generally requires signal characterization rather than signal coding using only a discrete set of code words. Therefore, Part VII is developed as a companion part of discrete-value hyperspectral signal coding to deal with continuous-value hyperspectral signal characterization, which can be considered as 1D continuous signal processing as opposed to hyperspectral signal coding to be treated as 1D discrete signal processing.

Due to very high spectral resolution provided by hyperspectral imaging sensors the spectral information among spectral bands is expected to be highly correlated, in which case a certain level of redundant information can be removed. To address this issue, three topics are of major interest in Part VII, which correspond to hyperspectral variable band selection for feature characterization in Chapter 27, hyperspectral signal estimation in Chapter 28, and hyperspectral signal representation in Chapter 29, respectively. Chapter 27 develops an approach to band selection that allows users to select variable number variable bands according to spectral characteristics of a

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

hyperspectral signal. In other words, each hyperspectral signal requires its own bands to characterize its spectral profile. As a result, two different hyperspectral signals must have two different sets of bands, each of which requires a different number of bands to be selected. This idea is derived from variable length coding commonly used by source coding in information theory where each source alphabet requires a code word with a different coding length derived from its probability. A similar idea called dynamic dimensionality allocation (DDA) for progressive band selection (PBS) was also explored in Chapter 23. As an alternative to variable number variable band selection to remove interband spectral correlation, a widely used technique in signal processing and communications, Kalman filtering is further investigated in Chapter 28 for hyperspectral signal estimation where three Kalman filter-based techniques, Kalman filter-based spectral signature estimator (KFSSE), Kalman filter-based spectral signature identifier (KFSSI), and Kalman filter-based spectral signature quantifier (KFSSQ), are developed to characterize hyperspectral signals. A third approach to removing redundant spectral information is to represent a hyperspectral signal in a more effective manner. One such representation is wavelets that make use of low-pass and high-pass filters to retain different levels of details for signal approximation. Chapter 29 takes up this approach to derive a wavelet-based signature characterization algorithm (WSCA) for hyperspectral signal representation that can be used for signature discrimination, classification, and identification.

27

Variable-Number Variable-Band Selection for Hyperspectral Signals

This chapter presents a novel band selection-based feature characterization technique for a hyperspectral signature vector, referred to as variable-number variable-band selection (VNVBS). Since a hyperspectral signature vector is generally characterized by its spectral profile, its feature characterization can be achieved by selecting appropriate bands from the original set of spectral bands, and the number of bands to be selected is totally determined by its original spectral profile. As a result, two hyperspectral signature vectors may require different sets of bands for spectral feature characterization. Therefore, VNVBS allows users to select a different number of variable bands in accordance with spectral characteristics of a hyperspectral signature vector. In order for VNVBS to select an appropriate subset of bands for a hyperspectral signature vector, a new band prioritization criterion, referred to as orthogonal subspace projector-based band prioritization criterion (OSP-BPC), is derived. It assigns a different priority score to each spectral band of a hyperspectral signature vector such that various features can be captured by VNVBS. Accordingly, VNVBS can be interpreted as a spectral band selection-based feature extraction technique for hyperspectral signature characterization.

27.1 Introduction

Hyperspectral data are collected by hundreds of contiguous and highly correlated spectral bands. Consequently, the same spectral bands used to acquire two different hyperspectral signatures may not provide the same level of signature information. Furthermore, recent advances in sensor technology have made it possible for sensor data to be acquired by more than hundreds or thousands of spectral channels, for example, hyperspectral or ultraspectral data. Of particular interest is chemical/biological (CB) defense for bioterrorism where CB data available for analysis are generally spectral data rather than image data. However, it also comes at a price that such wealthy spectral information is highly correlated. As a result, using all the hundreds or thousands of spectral channels might not be a good choice for preserving spectral information since a significant and crucial piece of information of interest may only be provided by a very narrow range of spectral coverage and could be overwhelmed by other dominated spectral channels. For example, the crucial information of chemical data is provided by the thermal range, and biological data are determined by their distinct protein spectral profiles,

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

which are usually very small and can only be captured by very narrow diagnostic spectral channels. Therefore, the information provided by their spectral profiles in signature characterization becomes vital, and band selection (BS)-based spectral signature analysis and characterization seem to be the most effective means to address this issue.

BS has been widely used in remote sensing image analysis for various application (Mausel et al., 1990; Conese and Maselli 1993; Stearns et al., 1993; Chang et al., 1999; Huang and He, 2005). It is discussed extensively in Chapters 21–23. However, most existing BS techniques are developed for images where the number of bands is fixed, and the bands selected for each image pixel vector are the same and identical. Unfortunately, a direct application of such image-based BS to single spectral signature analysis and characterization is not feasible due to the following two reasons. First, no sample spectral correlation among pixel vectors, which has been used by image-based BS, is available for a single hyperspectral signature vector. Second a different hyperspectral signature vector generally requires a different number of bands as well as different spectral bands to characterize its spectral profile. More specifically, in order to characterize a spectral signature vector effectively, variable-band numbers and variable bands should be selected for different spectral signature vectors for signature analysis. This chapter presents a new concept, referred to as variable-number variable-band selection (VNVBS), that can be used to effectively characterize the spectral profile of a single hyperspectral signature vector rather than a hyperspectral image pixel vector.

It is general understanding that commonly used BS techniques in remote sensing image processing are developed to explore correlation among spectral images rather than spectral correlation within a single image pixel vector. Therefore, they can be viewed as image-based approaches. By contrast, VNVBS is developed to only deal with single hyperspectral signature vectors, which may come from a database or a spectral library or non-image sensors not from an image cube. Therefore, there has no correlation among image pixel vectors that can be used by VNVBS. Instead, the only available information that can be used by VNVBS is the spectral band-to-band correlation within a single hyperspectral signature vector. Compared to the image-based BS, which considers a three-dimensional image cube as a whole, VNVBS actually operates on one-dimensional hyperspectral signature vectors, and thus it can be considered as one-dimensional signature-based BS. Additionally, both the band numbers and bands to be selected for each individual image pixel vector by the conventional image-based BS are always fixed and identical. To the contrary, VNVBS selects a variable-band number and variable bands for any given hyperspectral signature vector. In doing so, two key issues must be addressed: "How many bands are needed for VNVBS to perform analysis on a hyperspectral signature vector?" and "What bands are crucial for this particular hyperspectral signature vector?" Both of these issues can be simultaneously addressed by a new approach, referred to as orthogonal subspace projector-based band prioritization criterion (OSP-BPC) along with a so-called reference signature vector. OSP-BPC decomposes a hyperspectral signature vector $\mathbf{s}$ to be processed into two OSP components with respect to a reference signature vector from which a score for each particular band of the signature vector can be derived for prioritization. By virtue of OSP-BPC, the original band set Ω of a hyperspectral signature vector can be rearranged and divided into two disjoint sets, denoted by $\Omega_{\mathbf{s}}^{\perp}$ and $\Omega_{\mathbf{s}}$, in accordance with OSP-BPC assigned priority score for each band. Only those bands in $\Omega_{\mathbf{s}}^{\perp}$ have higher priorities than those in $\Omega_{\mathbf{s}}$. Since the bands from $\Omega_{\mathbf{s}}^{\perp}$ varies with the hyperspectral signature vector $\mathbf{s}$, two different hyperspectral signature vectors may result in different sets of $\Omega_{\mathbf{s}}^{\perp}$ because the number of bands to be selected is different, and the selected bands are also different as well.

One interesting finding is noteworthy. VNVBS can be used as a feature selection method. Compared to other traditional spectral similarity measures such as Euclidean distance (ED), spectral angle mapper (SAM) (Schowengerdt, 1997; Chang 2003a), spectral information divergence (SID) (Chang 2000, 2003a), which utilize the full band information, VNVBS judiciously selects bands that can best describe the spectral characterization in the sense of OSP. As a consequence, it can remove redundant spectral information, while retaining vital and crucial information so as to improve performance in spectral signature analysis and characterization. This is a task more than just spectral similarity.

27.2 Orthogonal Subspace Projection-Based Band Prioritization Criterion

Orthogonal subspace projector (OSP) approach has been widely used for hyperspectral target detection and classification (Harsanyi and Chang, 1994). It extends a standard signal detection model by dividing signature vectors of interest into two types of signals, called desired target signature vector, **d**, to be detected and undesired target signature vectors to be eliminated. The performance in detecting **d** can be improved by eliminating the undesired signature vectors prior to detection of **d**. This section extends the OSP concept to a band prioritization criterion.

Assume that **U** is an undesired signature matrix formed by placing all undesired target signature vectors as its column vectors. To eliminate all the signature vectors in **U**, an orthogonal subspace projector specified by (2.86) can be used for this purpose and defined by

$$P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\mathbf{U}^{\#} \tag{27.1}$$

where $\mathbf{U}^{\#} = \left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T$ is the pseudoinverse of **U** and **I** is an identity matrix. Applying $P_{\mathbf{U}}^{\perp}$ to a hyperspectral signature vector **s** leads to a new signature vector denoted by $\hat{\mathbf{s}}$ defined as follows:

$$\hat{\mathbf{s}} = P_{\mathbf{U}}^{\perp}\mathbf{s} = \left(\mathbf{I} - \mathbf{U}\mathbf{U}^{\#}\right)\mathbf{s} \tag{27.2}$$

where the undesired signature vectors in **U** have been eliminated from original signature vector **s** and $\hat{\mathbf{s}}$ is the signature vector resulting from projecting **s** onto the subspace $P_{\mathbf{U}}^{\perp}$. The geometric relationship between **s**, $P_{\mathbf{U}}^{\perp}$, and $\hat{\mathbf{s}}$ can be described in Figure 27.1.

By taking advantage of the OSP concept outlined by (27.1) and (27.2), a hyperspectral signature vector can be decomposed into two orthogonal projection components where the spectral bands of a hyperspectral signature vector can be ranked and selected according to their priorities measured by an OSP-based criterion, referred to as OSP-BPC presented in details in the following.

Assume that $\mathbf{s} = (s_1, s_2, \ldots, s_L)^T$ is a hyperspectral signature vector to be processed and $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ is the so-called reference signature vector against which the **s** is compared, where

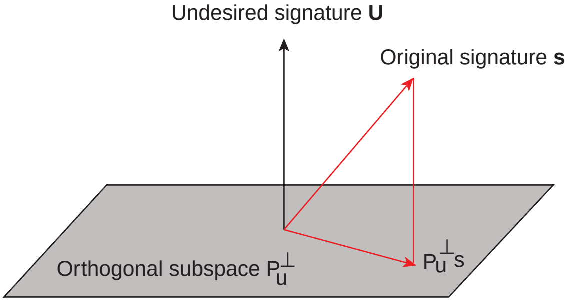


Figure 27.1 Geometric interpretation of between **s**, $P_{\mathbf{U}}^{\perp}$, and $\hat{\mathbf{s}}$.

L denotes the total number of bands used to acquire the signature vector $\mathbf{s}$. Two orthogonal projectors based on the reference signature vector $\mathbf{r}$, $P_{\mathbf{r}}$ and $P_{\mathbf{r}}^{\perp}$, can be further be defined by

$$P_{\mathbf{r}} = \mathbf{r}\mathbf{r}^{\#} \tag{27.3}$$

$$P_{\mathbf{r}}^{\perp} = \mathbf{I} - P_{\mathbf{r}} \tag{27.4}$$

where $\mathbf{r}^{\#} = \left(\mathbf{r}^T\mathbf{r}\right)^{-1}\mathbf{r}^T$ is the pseudoinverse of $\mathbf{r}$. By means of (27.3) and (27.4), a hyperspectral signature vector $\mathbf{s}$ can be projected onto two orthogonal subspaces $P_{\mathbf{r}}^{\perp}$ and $P_{\mathbf{r}}$ and decomposed into two orthogonal projection components $\mathbf{s}_{\mathbf{r}}^{\perp}$ and $\mathbf{s}_{\mathbf{r}}$, respectively, defined by

$$\begin{aligned} \mathbf{s}_{\mathbf{r}}^{\perp} &= P_{\mathbf{r}}^{\perp}\mathbf{s} = \left(\mathbf{I} - \mathbf{r}\mathbf{r}^{\#}\right)\mathbf{s} \\ \mathbf{s}_{\mathbf{r}} &= P_{\mathbf{r}}\mathbf{s} = \left(\mathbf{r}\mathbf{r}^{\#}\right)\mathbf{s} \end{aligned} \tag{27.5}$$

In general, the reference signature vector $\mathbf{r}$ is selected in such a way that it shares some information with the signature vector $\mathbf{s}$ to be processed in order to avoid $\mathbf{s}_{\mathbf{r}}^{\perp}$ being empty when $\mathbf{s}$ is projected onto $P_{\mathbf{r}}^{\perp}$ and $P_{\mathbf{r}}$ via (27.5). However, it should be noted that on some occasions, this may not be the case where the selected reference signature vector $\mathbf{r}$ may turn out to be either parallel to the signature vector $\mathbf{s}$, which results in $\mathbf{s}_{\mathbf{r}}^{\perp} = \varnothing$, or orthogonal to the signature vector $\mathbf{s}$, which results in $\mathbf{s}_{\mathbf{r}} = \varnothing$.

OSP-BPC algorithm

1. Preprocess the signature vector $\mathbf{s}$ by expanding the $\mathbf{s}$ into $(L+4)$-dimensional column vector by adding two zeros to both ends of $\mathbf{s}$ into the form of $(0, 0, s_1, s_2, \ldots, s_L, 0, 0)$.
2. Assume that the spectral value on the lth band of the signature vector $\mathbf{s}$ is s_l, $l = 1, 2 \ldots L$. For the lth band, signature s_l group its four neighboring bands, s_{l-2}, s_{l-1}, s_{l+1}, s_{l+2} from $(0, 0, s_1, s_2, \ldots, s_L, 0, 0)$ to form a vector centered at s_l defined by

$$\mathbf{s}_l^5 = \left(s_{l-2}, s_{l-1}, s_l, s_{l+1}, s_{l+2}\right)^T \tag{27.6}$$

Similarly, for the l-band of two orthogonal components, $\mathbf{s}_{\mathbf{r}}^{\perp}$, $\mathbf{s}_{\mathbf{r}}$, denoted by $\mathbf{s}_{\mathbf{r}_l}^{\perp}$ and $\mathbf{s}_{\mathbf{r}_l}$, we can also define $\left(\mathbf{s}_{\mathbf{r}}^{\perp}\right)_l^5$ and $\left(\mathbf{s}_{\mathbf{r}}\right)_l^5$ as follows:

$$\begin{aligned} \left(\mathbf{s}_{\mathbf{r}}^{\perp}\right)_l^5 &= \left(s_{\mathbf{r}_{l-2}}^{\perp}, s_{\mathbf{r}_{l-1}}^{\perp}, s_{\mathbf{r}_l}^{\perp}, s_{\mathbf{r}_{l+1}}^{\perp}, s_{\mathbf{r}_{l+2}}^{\perp}\right)^T \\ \left(\mathbf{s}_{\mathbf{r}}\right)_l^5 &= \left(s_{\mathbf{r}_{l-2}}, s_{\mathbf{r}_{l-1}}, s_{\mathbf{r}_l}, s_{\mathbf{r}_{l+1}}, s_{\mathbf{r}_{l+2}}\right)^T \end{aligned} \tag{27.7}$$

3. Calculate inner products between vector $\mathbf{s}_l^5$ and the two vectors $\left(\mathbf{s}_{\mathbf{r}}^{\perp}\right)_l^5$ and $\left(\mathbf{s}_{\mathbf{r}}\right)_l^5$ defined in (27.7) by

$$\begin{aligned} \left\langle \mathbf{s}_l^5, \left(\mathbf{s}_{\mathbf{r}}^{\perp}\right)_l^5 \right\rangle &= \left(\mathbf{s}_l^5\right)^T \left(\mathbf{s}_{\mathbf{r}}^{\perp}\right)_l^5 \\ \left\langle \mathbf{s}_l^5, \left(\mathbf{s}_{\mathbf{r}}\right)_l^5 \right\rangle &= \left(\mathbf{s}_l^5\right)^T \left(\mathbf{s}_{\mathbf{r}}\right)_l^5 \end{aligned} \tag{27.8}$$

4. For the lth band signature s_l, there is a pair of priority scores associated with it, defined by

$$\left(\left\langle \mathbf{s}_l^5, \left(\mathbf{s}_\mathbf{r}^\perp\right)_l^5 \right\rangle, \left\langle \mathbf{s}_l^5, (\mathbf{s}_\mathbf{r})_l^5 \right\rangle\right) \tag{27.9}$$

5. According to (27.9), the original band set $\Omega = \{1, 2, \ldots, L\}$ can be divided into two disjoint subsets, denoted by $\Omega_\mathbf{s}^\perp$ and $\Omega_\mathbf{s}$, and defined by

$$\begin{aligned} \Omega_\mathbf{s}^\perp &= \left\{l : \left\langle \mathbf{s}_l^5, \left(\mathbf{s}_\mathbf{r}^\perp\right)_l^5 \right\rangle \rangle \left\langle \mathbf{s}_l^5, (\mathbf{s}_\mathbf{r})_l^5 \right\rangle, \qquad l = 1, 2, \ldots L\right\} \\ \Omega_\mathbf{s} &= \left\{l : \left\langle \mathbf{s}_l^5, \left(\mathbf{s}_\mathbf{r}^\perp\right)_l^5 \right\rangle \langle \left\langle \mathbf{s}_l^5, (\mathbf{s}_\mathbf{r})_l^5 \right\rangle, \qquad l = 1, 2, \ldots L\right\} \end{aligned} \tag{27.10}$$

such that $\Omega_\mathbf{s}^\perp$ collects those bands that contain more information in $P_\mathbf{r}^\perp$ than that in $P_\mathbf{r}$ in the sense of OSP, while the set $\Omega_\mathbf{s}$ does oppositely.

The motivation of using the four adjacent neighboring bands in (27.6) comes from the image processing which uses four-neighbor connectivity and eight-neighbor connectivity to account for inter-pixel spatial correlation within a 3 × 3 window (Gonzales and Woods, 2002). This idea is extended to the spectral domain to capture inter-band correlation within a 5-band 1 × 5 window. Of course, the same idea can also be applied to 7-band, 9-band windows, etc. However, according to our experiments, using 5-band 1 × 5 window seems to be the best compromise because using a window of a band number greater than 5 has little improvement on performance at the expense of computational complexity, while the performance using 5-band window indeed improves significantly over that using 3-band window.

27.3 Variable-Number Variable-Band Selection

The pair of $\left\langle \mathbf{s}_l^5, \left(\mathbf{s}_\mathbf{r}^\perp\right)_l^5 \right\rangle$ and $\left\langle \mathbf{s}_l^5, (\mathbf{s}_\mathbf{r})_l^5 \right\rangle$ defined in (27.8) associated with the lth band signature s_l is considered as two correlated prioritization scores of the lth band, which are essentially two pieces of information contained in two orthogonal subspaces, $P_\mathbf{r}^\perp$ and $P_\mathbf{r}$. Due to the principle of orthogonality (Poor, 1994), the information in $P_\mathbf{r}^\perp$ generally provides innovations information about the signature vector $\mathbf{s}$ with respect to reference signature vector $\mathbf{r}$. Therefore, only the $\left\langle \mathbf{s}_l^5, \left(\mathbf{s}_\mathbf{r}^\perp\right)_l^5 \right\rangle$ in (27.8) is used to rank the bands in Ω according to descending order of its magnitude. The ranked band set resulting from using $\left\langle \mathbf{s}_l^5, \left(\mathbf{s}_\mathbf{r}^\perp\right)_l^5 \right\rangle$ priority can be further broken up into two disjoint sets, $\Omega_\mathbf{s}^\perp$ and $\Omega_\mathbf{s}$, determined by (27.10). By realigning $\Omega_\mathbf{s}^\perp$ and $\Omega_\mathbf{s}$, a new priority-ranked band set will be generated for further band selection, denoted by $\Omega_\mathbf{s}^*$

$$\Omega_\mathbf{s}^* = \left(\Omega_\mathbf{s}^\perp, \Omega_\mathbf{s}\right) \tag{27.11}$$

In what follows, VNVBS is developed by only selecting those bands in the set $\Omega_\mathbf{s}^\perp$, based on the pair of priority scores defined by (27.9). Since the size of the set $\Omega_\mathbf{s}^\perp$ varies with the signature vector $\mathbf{s}$ to be prioritized, the number of bands to be selected for various hyperspectral signatures is not fixed *a priori*, but rather determined by comparison between $\left\langle \mathbf{s}_l^5, \left(\mathbf{s}_\mathbf{r}^\perp\right)_l^5 \right\rangle$ and $\left\langle \mathbf{s}_l^5, (\mathbf{s}_\mathbf{r})_l^5 \right\rangle$ via (27.10).

In hyperspectral signature characterization, a frequently encountered application is signature discrimination, which involves two different signature vectors, $\mathbf{s}_1$ and $\mathbf{s}_2$. In this case, VNVBS is implemented using the following steps, referred to as VNVBS-based hyperspectral signature discrimination (VNVBS-HSD).

VNVBS-HSD

1. Given two signature vectors to be discriminated, $\mathbf{s}_1$ and $\mathbf{s}_2$, select a reference signature vector $\mathbf{r}$ onto which both $\mathbf{s}_1$ and $\mathbf{s}_2$ are orthogonally projected. The issue of how to select the $\mathbf{r}$ will be discussed in detail at the end of Section 27.5.
2. Obtain the priority-ranked band sets for the $\mathbf{s}_1$ and $\mathbf{s}_2$ according to (27.10), denoted by $\Omega^*_{\mathbf{s}_1} = \left(\Omega^\perp_{\mathbf{s}_1}, \Omega_{\mathbf{s}_1}\right)$ and $\Omega^*_{\mathbf{s}_2} = \left(\Omega^\perp_{\mathbf{s}_2}, \Omega_{\mathbf{s}_2}\right)$, respectively. A geometric interpretation of the relationship between $\Omega^*_{\mathbf{s}_1} = \left(\Omega^\perp_{\mathbf{s}_1}, \Omega_{\mathbf{s}_1}\right)$ and $\Omega^*_{\mathbf{s}_2} = \left(\Omega^\perp_{\mathbf{s}_2}, \Omega_{\mathbf{s}_2}\right)$ is depicted in Figure 27.2 as follows.
3. Find a new band set, denoted by $\Omega^\perp(\mathbf{s}_1, \mathbf{s}_2)$ which is generated by the $\Omega^\perp_{\mathbf{s}_1}$ and $\Omega^\perp_{\mathbf{s}_2}$ as follows:

$$\Omega^\perp(\mathbf{s}_1, \mathbf{s}_2) = \begin{cases} \Omega^\perp_{\mathbf{s}_1} \cap \Omega^\perp_{\mathbf{s}_2} & \textit{if} \quad \Omega^\perp_{\mathbf{s}_1} \cap \Omega^\perp_{\mathbf{s}_2} \neq \varnothing \\ \Omega^\perp_{\mathbf{s}_1} \cup \Omega^\perp_{\mathbf{s}_2} & \textit{if} \quad \Omega^\perp_{\mathbf{s}_1} \cap \Omega^\perp_{\mathbf{s}_2} = \varnothing \end{cases} \tag{27.12}$$

   In other words, the $\Omega^\perp(\mathbf{s}_1, \mathbf{s}_2)$ is considered as those bands that can be used for discrimination between signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ in the sense that $\Omega^\perp(\mathbf{s}_1, \mathbf{s}_2)$ can best preserve the information required by $\mathbf{s}_1$ and $\mathbf{s}_2$ for signature discrimination.
4. Generate two new signatures vector $\mathbf{s}^*_1$ and $\mathbf{s}^*_2$ for both signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$, respectively, based on the bands selected by (27.12).
5. Use a spectral similarity measure such as SAM, SID, or ED to perform signature discrimination on the signature vectors $\mathbf{s}^*_1$ and $\mathbf{s}^*_2$ obtained in step 4.

Two notes on the criterion specified by (27.12) are worthwhile.

1. For the case of nonempty-intersection, that is, $\Omega^\perp_{\mathbf{s}_1} \cap \Omega^\perp_{\mathbf{s}_2} \neq \varnothing$.

   The $\Omega^\perp_{\mathbf{s}_1} \cap \Omega^\perp_{\mathbf{s}_2}$ is chosen due to the fact that it carries more critical spectral information than $\Omega^\perp_{\mathbf{s}_1} \cup \Omega^\perp_{\mathbf{s}_2}$ in discrimination between $\mathbf{s}_1$ and $\mathbf{s}_2$ based on the following two reasons.

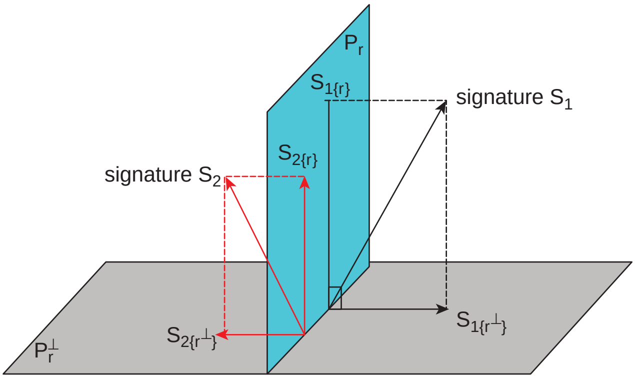


Figure 27.2 Geometry interpretation of the relationship between $\Omega^*_{\mathbf{s}_1} = \left(\Omega^\perp_{\mathbf{s}_1}, \Omega_{\mathbf{s}_1}\right)$ and $\Omega^*_{\mathbf{s}_2} = \left(\Omega^\perp_{\mathbf{s}_2}, \Omega_{\mathbf{s}_2}\right)$.

a. Since $\Omega_{\mathbf{s}_1}^{\perp}$ and $\Omega_{\mathbf{s}_2}^{\perp}$ retain more crucial local spectral information of $\mathbf{s}_1$ and $\mathbf{s}_2$ in $P_{\mathbf{r}}^{\perp}$ than that in $P_{\mathbf{r}}$ in terms of discriminating spectral signatures $\mathbf{s}_1$ and $\mathbf{s}_2$ via orthogonal projection, the bands in $\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp}$ is the smallest band set that can achieve the best possible spectral discrimination between $\mathbf{s}_1$ and $\mathbf{s}_2$.

b. On the other hand, if the $\Omega_{\mathbf{s}_1}^{\perp} \cup \Omega_{\mathbf{s}_2}^{\perp}$ is chosen instead of $\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp}$, the bands in $([\Omega_{\mathbf{s}_1}^{\perp} \cup \Omega_{\mathbf{s}_2}^{\perp}] - [\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp}])$ may potentially reduce discrimination power between $\mathbf{s}_1$ and $\mathbf{s}_2$ due to the fact that the bands in $\Omega_{\mathbf{s}_1}^{\perp} - (\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp})$ and $\Omega_{\mathbf{s}_2}^{\perp} - (\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp})$ are either spectrally representative for $\mathbf{s}_1$ or $\mathbf{s}_2$, but not for both. For example, using the bands from $\Omega_{\mathbf{s}_1}^{\perp} - (\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp})$ may be effective in better differentiating $\mathbf{s}_1$ from $\mathbf{s}_2$, but not necessarily the other way around because the bands in $\Omega_{\mathbf{s}_1}^{\perp} - (\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp})$ are not part of $\Omega_{\mathbf{s}_2}^{\perp}$ and cannot effectively discriminate $\mathbf{s}_2$ from $\mathbf{s}_1$ as those bands in $\Omega_{\mathbf{s}_2}^{\perp}$ do. Similarly, it is also true for the bands in $\Omega_{\mathbf{s}_2}^{\perp} - (\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp})$ which have better discrimination of $\mathbf{s}_2$ from $\mathbf{s}_1$, but do not necessarily have better discrimination of $\mathbf{s}_1$ from $\mathbf{s}_2$. Therefore, if bands in both band sets $\Omega_{\mathbf{s}_1}^{\perp} - (\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp})$ and $\Omega_{\mathbf{s}_2}^{\perp} - (\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp})$ are selected, these bands may be able to discriminate one from another but may be also very likely to deteriorate discrimination between $\mathbf{s}_2$ and $\mathbf{s}_1$. More specifically, an improvement upon discrimination of $\mathbf{s}_1$ from $\mathbf{s}_2$ using the bands from $\Omega_{\mathbf{s}_1}^{\perp} - (\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp})$ may further impair the discrimination of $\mathbf{s}_2$ from $\mathbf{s}_1$, and vice versa. A comprehensive study on comparison between using $\Omega_{\mathbf{s}_1}^{\perp} \cup \Omega_{\mathbf{s}_2}^{\perp}$ and $\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp}$ demonstrated that selecting bands from $\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp}$ outperformed those selected from $\Omega_{\mathbf{s}_1}^{\perp} \cup \Omega_{\mathbf{s}_2}^{\perp}$ in discrimination between $\mathbf{s}_1$ and $\mathbf{s}_2$ in both ways, that is, discrimination of $\mathbf{s}_1$ from $\mathbf{s}_2$ as well as discrimination of $\mathbf{s}_2$ from $\mathbf{s}_1$.

2. For the case of empty-intersection, that is, $\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp}$.

   Since no bands are in common, the best way to achieve both discrimination of $\mathbf{s}_1$ from $\mathbf{s}_2$ and discrimination of $\mathbf{s}_2$ from $\mathbf{s}_1$ is to select all the bands in $\Omega_{\mathbf{s}_1}^{\perp} \cup \Omega_{\mathbf{s}_2}^{\perp}$ for $\Omega^{\perp}(\mathbf{s}_1, \mathbf{s}_2)$, which turns out to be the smallest band set in discrimination between $\mathbf{s}_1$ and $\mathbf{s}_2$.

So, to have better discrimination between $\mathbf{s}_1$ and $\mathbf{s}_2$, both the better discrimination of $\mathbf{s}_2$ from $\mathbf{s}_1$ and the better discrimination of $\mathbf{s}_2$ from $\mathbf{s}_1$ must be achieved. Taking the intersection $\Omega_{\mathbf{s}_1}^{\perp} \cap \Omega_{\mathbf{s}_2}^{\perp}$ is the only way to accomplish the task while retaining the smallest possible band set.

Another note is also worth being mentioned. Despite the fact that VNVBS-HSD is developed to discriminate one signature from another, its functionality is not limited to signature discrimination. For example, if a signature is known to be detected, VNVBS-HSD turns out to be a detector for this particular signature. On the other hand, if a group of signatures of interest are available for classification, then VNVBS can be used as a signature classifier. Furthermore, if there is a database to be used for signature identification, VNVBS can become a signature identifier.

Finally, we summarize the differences of VNVBS from commonly used image-based BS techniques described in Section 6.7 and Chapters 21–23 which are generally performed via various criteria such as variance, signal-to-noise ratio (SNR), and information divergence.

1. VNVBS only involves the spectral correlation among individual bands within a hyperspectral signature vector as opposed to conventional image-based BS techniques, which consider each individual band image as a whole. Therefore, VNVBS must rely only on inter-band spectral correlation to select bands comparing to conventional image-based BS techniques which make use of spectral correlation among image pixels to select desired bands. As a result, the number of bands to be selected by VNVBS varies with the signature vector to be processed, while the number of bands chosen by the conventional image-based BS techniques are fixed for all image pixels,

2. The band prioritization criterion used in VNVBS is OSP-BPC, which is easy and simple to implement. It also deviates from commonly used BS criteria such as variance, SNR, and information divergence.

Interestingly, VNVBS can also be performed on hyperspectral signature vectors acquired by different numbers of bands such as gas data whose band numbers are different. In doing so, two approaches are suggested. One is to select their common bands to yield the same number of bands. The other is to consider all bands while performing zero-interpolation if bands in one signature are missing in the other signature. As a result, two hyperspectral signature vectors acquired by different numbers of spectral bands can be prioritized by OSP-BPC and characterized by VNVBS via their spectral features. Although these two strategies are applicable to any spectral similarity measure, there is no follow-up prioritization as what OSP-BPC does that can be developed for any spectral similarity measure in further spectral feature characterization. This benefit cannot be gained by any band selection technique developed for images.

27.4 Experiments

To demonstrate the utility of VNVBS in hyperspectral signature characterization, two completely different data sets were used for experiments and two particular applications, signature discrimination and mixed signature classification/identification were of interest and further considered for comparative analysis with SAM and SID used as spectral similarity measures. Nevertheless, other applications can also be explored for VNVBS.

27.4.1 Hyperspectral Data

The hyperspectral data used for computer simulations presented in this section were the five AVIRIS reflectance spectral signature vectors, blackbrush, creosote leaves, dry grass, red soil, and sagebrush, shown in Figure 1.8. Each of these five spectral signature vectors has 158 bands after water bands were removed and can be considered as a 158-dimensional hyperspectral signature vector where each signature component is specified by a particular spectral wavelength. According to Chapter 2 in Chang (2003a), the spectral profiles of blackbrush, creosote leaves, and sagebrush were close to each other. In particular, the creosote leaves and sagebrush even have very close spectral values. A detailed quantitative analysis among these three signature vectors is provided in Chapter 2 in Chang (2003a). In this section, these five signature vectors constitute a spectral library or database to be used to evaluate the performance of VNVBS in three different applications, signature discrimination, classification, and identification.

27.4.1.1 Signature Discrimination

Three similar signatures blackbrush, creosote leaves, and sagebrush were used for discrimination and the reference vector **r** specified by (27.3) and (27.4) was chosen to be a signature vector obtained by averaging these three signature vectors, $(\mathit{blackbrush} + \mathit{creosote\ leaves} + \mathit{sagebrush})/3$. Table 27.1 lists the bands prioritized by VNVBS according to (27.9), where the original set of bands was divided into two subsets of bands, $\Omega_{\mathbf{s}}^{\perp}$ and $\Omega_{\mathbf{s}}$ via (27.10).

As noted in Table 27.1, the $\Omega_{\mathbf{s}}^{\perp}$ obtained for creosote leaves from VNVBS via orthogonal subspace decomposition is empty. This case occured when either $\left\langle \mathbf{s}_l^5, (\mathbf{s}_{\mathbf{r}}^{\perp})_l^5 \right\rangle \rangle \left\langle \mathbf{s}_l^5, (\mathbf{s}_{\mathbf{r}})_l^5 \right\rangle$ for all $l \in \{1, 2, \ldots, L\}$ or $\left\langle \mathbf{s}_l^5, (\mathbf{s}_{\mathbf{r}}^{\perp})_l^5 \right\rangle \langle \left\langle \mathbf{s}_l^5, (\mathbf{s}_{\mathbf{r}})_l^5 \right\rangle$ for all $l \in \{1, 2, \ldots, L\}$. However, it should be noted that whether or not a particular band was removed or preserved was completely determined by how a

Table 27.1 $\Omega_{\mathbf{s}}^{\perp}$ and $\Omega_{\mathbf{s}}$ for blackbrush, creosote leaves, and sagebrush

S	$\Omega_{\mathbf{s}}^{\perp}$	$\Omega_{\mathbf{s}}$
Blackbrush	21–25, 91–98, 118–158	1–20, 26–90, 99–117
Creosote leaves	Empty	1–158
Sagebrush	1–23, 92–100, 121–158	24–91, 101–120

reference signature vector was selected and how the local spectral correlation described by (27.10) was involved among its four neighboring bands.

It should be noted that VNVBS grouped four-band neighbors of a spectral band to generate a five-band vector to capture its local spectral shape features rather than the spectral global tendency of a hyperspectral signature vector across the entire wavelengths. If we carefully compare the flat regions and edges of blackbrush, creosote leave, and sagebrush, the local spectral shapes captured by flat wavelengths of these three signature vectors were very close to each other, while the edges were much more different from each other. This finding explained why VNVBS favored local edges over global flat regions. According to OSP-BPC criterion, global flat regions seemed to contain redundant information which had no benefit to discrimination compared to local edges which contained spectral characteristics that generally improved distance among three signatures in terms of spectral similarity.

Table 27.2 tabulates the values of SAM and SID applied to the three VNVBS-generated new signatures of blackbrush, creosote leaves, and sagebrush by (27.12) where the upper and lower values were obtained by the original full band set and VNVBS, respectively, and the least discrimination results are highlighted by shade.

It should be pointed out that the discriminatory power of a spectral measure was not determined by the magnitude of its spectral similarity value, but rather by the relative magnitude of one spectral value to another value. So, even though the values of SAM and SID obtained by full bands were slightly larger than those obtained by VNVBS, the ratio of one SAM (or SID) value to another SAM (or SID) value using VNVBS was greatly increased compared to that using full bands. To see this more clearly, we normalized the least discrimination results, which were SAM or SID values between blackbrush and sagebrush, to 1, and then a new table could be generated in Table 27.3 from Table 27.2 where it clearly showed that the relative discrimination between two signatures was greatly improved by VNVBS.

The above simple experiment demonstrated that the relative discrimination between these three signature vectors could be significantly increased if VNVBS is used. This insight provided

Table 27.2 Discrimination among blackbrush, creosote leaves, and sagebrush using (27.12) and Table 27.1

full bands VNVBS	(blackbrush, creosote)	(blackbrush, sagebrush)	(creosote, sagebrush)
SAM	0.1767 0.1234	0.0681 0.0317	0.1289 0.1145
SID	0.0497 0.0161	0.0063 0.0011	0.0303 0.0140

Table 27.3 Rescaled discrimination among blackbrush, creosote leaves, and sagebrush using from Table 27.2

full bands VNVBS	(blackbrush, creosote)	(blackbrush, sagebrush)	(creosote, sagebrush)
SAM	2.59 3.89	1 1	1.89 3.61
SID	7.89 14.64	1 1	4.81 12.73

evidence that in order to measure the effectiveness of VNVBS relative to full bands, a direct comparison using Table 27.2 may not be appropriate. To address this issue, a measure suggested in Chang (2000) and Chang (2003a), called the relative spectral discriminatory power (RSDPW) seems to fit our need and was be used for performance evaluation.

Assume that m is any given hyperspectral measure, and $\mathbf{s}_1$, $\mathbf{s}_2$ are a pair of two spectral signature vectors to be measured. Let $\boldsymbol{\beta}$ be a third arbitrary signature vector with respect to which the two signature vectors $\mathbf{s}_1$, $\mathbf{s}_2$ are compared against. The RSDPW of m, denoted by $\text{RSDPW}_m(\mathbf{s}_1,\mathbf{s}_2;\boldsymbol{\beta})$, is defined in Chang (2000) and Chang (2003a) by

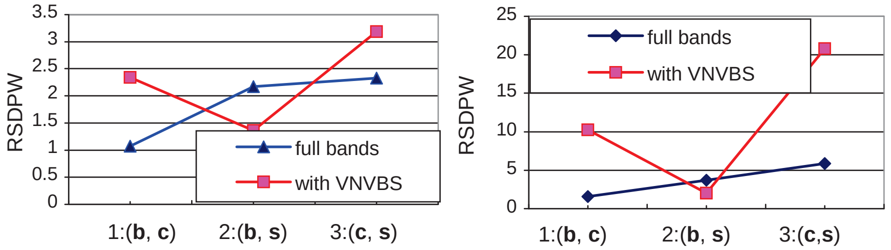
$$\text{RSDPW}_m(\mathbf{s}_1,\mathbf{s}_2;\boldsymbol{\beta}) = \max\{m(\mathbf{s}_1,\boldsymbol{\beta})/m(\mathbf{s}_2,\boldsymbol{\beta}), m(\mathbf{s}_2,\boldsymbol{\beta})/m(\mathbf{s}_1,\boldsymbol{\beta})\} \tag{27.13}$$

which measures the discriminatory power of the measure m by finding the maximum of two ratios, ratio of $m(\mathbf{s}_1,\boldsymbol{\beta})$ to $m(\mathbf{s}_2,\boldsymbol{\beta})$ and ratio of $m(\mathbf{s}_2,\boldsymbol{\beta})$ to $m(\mathbf{s}_1,\boldsymbol{\beta})$. The $\text{RSDPW}_m(\mathbf{s}_1,\mathbf{s}_2;\boldsymbol{\beta})$ defined by (27.13) provides a quantitative index of spectral discrimination capability of a specific hyperspectral measure m between two spectral signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ with respect to a third signature vector $\boldsymbol{\beta}$. Therefore, the higher the $\text{RSDPW}_m(\mathbf{s}_1,\mathbf{s}_2;\boldsymbol{\beta})$ is, the better discriminatory power the m is. In addition, $\text{RSDPW}_m(\mathbf{s}_1,\mathbf{s}_2;\boldsymbol{\beta})$ is symmetric and bounded below by one which is achieved by equality if and only if $\mathbf{s}_1 = \mathbf{s}_2$. It should be noted that the reference signature vector $\mathbf{r}$ used in OSP-BPC for VNVBS is similar to the concept of using a third signature vector $\boldsymbol{\beta}$ in RSDPW.

Using (27.13), Table 27.4 tabulates RSDPW values obtained by full bands and the VNVBS using the SAM and SID as the measures m in (27.13) with the signature vector $\boldsymbol{\beta}$ chosen to be the reference signature vector $\mathbf{r}$ which was the averaged signature vector over blackbrush, creosote leave, and sagebrush, denoted by $(\mathit{blackbrush} + \mathit{creosote\ leaves} + \mathit{sagebrush})/3$.

Table 27.4 RSDPW values of SAM and SID with and without VNVBS

full bands VNVBS	(blackbrush, creosote)	(blackbrush, sagebrush)	(creosote, sagebrush)
SAM	1.0715 2.3422	2.1693 1.3615	2.3244 3.1890
SID	1.5793 10.2230	3.7215 2.0333	5.8772 20.7859



(a) SAM (b) SID

Figure 27.3 Comparison between RSDPW of VNVBS and full bands using SAM and SID.

The experimental results in Table 27.4 demonstrated that the relative discriminatory powers were significantly increased by VNVBS in the sense that the higher the RSDPW value, the better the discrimination between two signature vectors. On the other hand, the smaller the RSDPW value is, the more difficult the discrimination between two signature vectors is.

To have better visual assessment, Figure 27.3(a) and (b) also shows the graphical representations of Table 27.4 where RSDPW was graphically plotted as the y-axis against a pair of signature vectors ($\mathbf{s}_1$,$\mathbf{s}_2$) along the x-axis with the reference signature vector $\mathbf{r}$ specified by $\boldsymbol{\beta}$. For example, 1:(**b**,**c**) in Figure 27.3(a) represents $\text{RSDPW}_{\text{SAM}}(\mathbf{s}_1,\mathbf{s}_2;\boldsymbol{\beta})$ of SAM comparing $\mathbf{s}_1 =$ '**blackbrush**' against $\mathbf{s}_2 =$ '**creosote leaves**' with $\boldsymbol{\beta} = \mathbf{r} =$ '$(\textit{blackbrush} + \textit{creosote leaves} + \textit{sagebrush})/3$.

To Figure 27.3(a) and (b), a tremendous improvement of RSDPW values produced by VNVBS over using full bands was visually apparent because RSDPW values between blackbrush (**b**) and sagebrush (**s**) was reduced, and the other two pairs, (**b**,**c**) and (**c**,**s**) were greatly increased. More specifically, the contrast between similarity and dissimilarity of two signature vectors in terms of RSDPW has been significantly enhanced and increased by VNVBS. Furthermore, comparing Figure 27.3(a) with Figure 27.3(b), SID was also shown to outperform SAM in discrimination between blackbrush, creosote leave, and sagebrush because RSDPW values of SID was much higher than that of SAM in terms of the contrast between signature similarity and signature dissimilarity. Because of that, only SID would be used for study and analysis in the following experiments.

27.4.1.2 Signature Classification/Identification

In this section, we further assumed that there was a class of signature vectors, blackbrush (**b**), creosote leaves (**c**), dry grass (**d**), and sagebrush (**s**) of interest for classification. We then simulated a mixed signature vector $\mathbf{t}^{\text{mix}}$ by uniformly mixing ¼ **b**, ¼ **c**, ¼ **d**, and ¼ **s** as follows:

$$\mathbf{t}^{\text{mix}} = 0.25 \cdot \mathbf{b} + 0.25 \cdot \mathbf{c} + 0.25 \cdot \mathbf{d} + 0.25 \cdot \mathbf{s} \tag{27.14}$$

According to the results in Chang (2000) and Chang (2003a), the spectral signature profile of blackbrush was more close to the spectral signature profile of sagebrush than to the spectral signature profile of creosote leaves. Similarly, the spectral signature profile of the creosote leaves was more close to the spectral signature profile of sagebrush than to the spectral signature profile of blackbrush. Therefore, both blackbrush and creosote leaves were considered as mutated signature vectors of sagebrush. In this case, the total amount of abundance fractions contributed by blackbrush, creosote leaves, and sagebrush is 75% compared to only 25%

Table 27.5 $\Omega_{\mathbf{s}}^{\perp}$ and $\Omega_{\mathbf{s}}$ for blackbrush, creosote leaves, and sagebrush

	$\Omega_{\mathbf{s}}^{\perp}$	$\Omega_{\mathbf{s}}$
Blackbrush	25,26	1–24, 27–158
Creosote leaves	47,48,49	1–46, 50–158
Sagebrush	43–48	1–42, 49–158
Drygrass	13–23, 92–98, 119–158	1–12, 24–91, 99–118,
$\mathbf{t}^{\text{mix}}$	None	1–158

contributed by drygrass. As a result, it is natural to conclude that the mixture signature vector $\mathbf{t}^{\text{mix}}$ in (27.14) should be classified as sagebrush because sagebrush contributed more to the mixed signature vector than any other three signature vectors. This interesting scenario sheds some light on the impact of BS on signature analysis. In this case, the reference signature vector $\mathbf{r}$ for VNVBS was simply chosen to be the $\mathbf{t}^{\text{mix}}$, Table 27.5 lists the bands prioritized by VNVBS according to (27.9), where the original set of bands was divided into two subsets of bands, $\Omega_{\mathbf{s}}^{\perp}$ and $\Omega_{\mathbf{s}}$ via (27.10).

Like Table 27.1, $\Omega_{\mathbf{s}}^{\perp}$ is empty for $\mathbf{t}^{\text{mix}}$. This was very obvious because the reference signature vector $\mathbf{r}$ specified by (27.3)–(27.4) was equal to the signature vector $\mathbf{s}$, both of which were $\mathbf{t}^{\text{mix}}$. Table 27.6 tabulates the spectral similarity values measured by SID between VNVBS-generated new signatures of $\mathbf{t}^{\text{mix}}$ and each of its components, $\mathbf{b}$, $\mathbf{c}$, $\mathbf{d}$, and $\mathbf{s}$, according to the bands selected by Table 27.5 where the upper and lower values in Table 27.6 were obtained by the full band set and the priority-ranked band set $\Omega_{\mathbf{s}}^{\perp}$, respectively. In particular, a smallest SID value across a row was highlighted by shade for classification. The rescaled SID values, as well as the corresponding RSDPW values with $\boldsymbol{\beta}$ specified by $\mathbf{t}^{\text{mix}}$, were also shown in Table 27.6 for direct comparisons.

The results in Table 27.6 demonstrated that SID using VNVBS greatly improved the performance for mixed signature classification over SID using full bands because the former could correctly classify the mixed signature into sagebrush while the latter could not. In addition, the contrast between different discrimination powers provided by SID was also increased by VNVBS compared to that by full bands as shown by the rescaled SID values in the second row

Table 27.6 SID, rescaled SID, and RSDPW values between $\mathbf{t}^{\text{mix}}$ and each component using bands obtained by (27.12) and Table 27.5

full bands VNVBS	$(\mathbf{t}^{\text{mix}},\mathbf{b})$	$(\mathbf{t}^{\text{mix}},\mathbf{s})$	$(\mathbf{t}^{\text{mix}},\mathbf{c})$	$(\mathbf{t}^{\text{mix}},\mathbf{d})$
SID	0.0052 5e-005	0.0081 1e-006	0.0661 2e-005	0.0500 0.0039
Rescaled SID	1 50	1.56 1	12.71 20	9.62 3900
RSDPW	192.2 20096	123.3 935720	15.123 45974	20.00 254.71

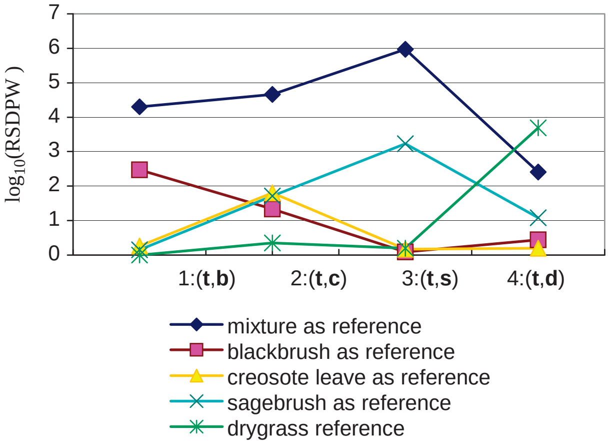


Figure 27.4 RSDPW curves corresponding to five different selections of reference signatures of using VNVBS with similarity measured by SID.

of Table 27.6. For example, the maximum ratio between these two pairs using full bands was only 12.71 as opposed to 3900 using VNVBS.

To further investigate the impact of the reference signature vector used by the VNVBS, experiments with different choices of the reference signature vector **r** were used, blackbrush, creosote leaves, sagebrush, and drygrass, respectively. As a result, five RSDPW$_{\text{SID}}$(**t**,**x**;$\boldsymbol{\beta}$) plots were generated, each of which was produced by one particular choice of reference signature vector **r** as shown in Figure 27.4 with SID used as a spectral similarity measure where '**t**' was $\mathbf{t}^{\text{mix}}$, and "**x**" could be any one of the four signatures, '**b**', '**c**', '**s**', and '**d**' which represent, blackbrush, creosote leaves, sagebrush, and drygrass with $\boldsymbol{\beta} = \mathbf{t}^{\text{mix}}$. Also the interpretation of RSDPW between two signature vectors in Figure 27.4 was the same as that made in Figure 27.3(a) and (b).

As shown in Figure 27.4, using $\mathbf{t}^{\text{mix}}$ as the reference signature vector seemed to outperform any other selected reference signature vector.

The aforementioned experiments were also for mixed signature identification if a class of signature vectors to be classified was replaced by a database or spectral library. In this case, the mixed signature $\mathbf{t}^{\text{mix}}$ would be identified as sagebrush via the assumed database $\Delta = \{\mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{s}\}$.

27.4.1.3 Noise Effect on VNVBS

Finally, this section concludes with an investigation of noise effects on the performance of VNVBS. Adding a Gaussian noise **n** to (27.13) yielded a noisy mixed signature as follows:

$$\tilde{\mathbf{t}}^{\text{mix}} = 0.25 \cdot \mathbf{b} + 0.25 \cdot \mathbf{c} + 0.25 \cdot \mathbf{d} + 0.25 \cdot \mathbf{s} + \mathbf{n} \qquad (27.15)$$

Since it is shown in Section 27.4.1.2 that $\mathbf{t}^{\text{mix}}$ was a better option for the reference signature vector, $\tilde{\mathbf{t}}^{\text{mix}}$ was also selected as the reference signature vector **r**.

Table 27.7 tabulates the SID-measured spectral similarity values between $\tilde{\mathbf{t}}^{\text{mix}}$ and each component, **b**, **c**, **d**, and **s**, respectively, where a Gaussian noise was used to produce three different SNRs, 10 dB, 20 dB, and 30 dB as defined in Harsanyi and Chang (1994). The upper and lower entries in

Table 27.7 SID values between $\tilde{\mathbf{t}}^{\text{mix}}$ and each of its components, **b**, **c**, **d**, **s** with three SNRs, 10 dB, 20 dB, and 30 dB

full VNVBS	$(\tilde{\mathbf{t}}^{\text{mix}},\mathbf{b})$	$(\tilde{\mathbf{t}}^{\text{mix}},\mathbf{c})$	$(\tilde{\mathbf{t}}^{\text{mix}},\mathbf{s})$	$(\tilde{\mathbf{t}}^{\text{mix}},\mathbf{d})$
10dB	0.0074	0.0674	0.0100	0.0532
	0.0052	0.0062	0.0068	0.0073
20dB	0.0052	0.0662	0.0081	0.0500
	0.0125	0.0351	0.0029	0.0058
30dB	0.0052	0.0661	0.0081	0.0500
	0.00005	0.00002	0	0.0039

Table 27.7 are the SID-values obtained using VNVBS and full bands, respectively, where a smallest SID value was highlighted by shade to indicate the classification of $\tilde{\mathbf{t}}^{\text{mix}}$.

As a comparison, the RSDPW values between $\tilde{\mathbf{t}}^{\text{mix}}$ and each component with three SNRs were shown in Table 27.8. Similarly, the upper and lower entries in Table 27.8 were the RSDPW values obtained using VNVBS and full bands, respectively.

An interesting finding can be observed from Tables 27.7 and 27.8. The higher the SNR value is, the lower the noise level relative to the signal, and vice versa. Therefore, according to Tables 27.7 and 27.8, when SNR = 10 dB which indicated that the noise level was high and the signal was overwhelmed by noise, the SID values calculated by both VNVBS and full bands classified $\tilde{\mathbf{t}}^{\text{mix}}$ as blackbrush instead of sagebrush as the noise-free case did in Table 27.6 since the spectral profile, especially the local spectral profile suffered from significant distortion due to noise effects. However, when SNR = 20 dB or 30 dB which implied that signal began to show its dominance compared to the case of SNR = 10 dB. As a result, the SID values using the VNVBS correctly classified $\tilde{\mathbf{t}}^{\text{mix}}$ as sagebrush compared to the SID values using full bands which still classified $\tilde{\mathbf{t}}^{\text{mix}}$ as blackbrush. This experiment further showed an advantage of using VNVBS over using full bands.

Table 27.8 RSDPW values between $\tilde{\mathbf{t}}^{\text{mix}}$ and each of its components, **b**, **c**, **d**, **s** with three SNRs, 10, 20, and 30 dB

full VNVBS	$(\tilde{\mathbf{t}}^{\text{mix}},\mathbf{b})$	$(\tilde{\mathbf{t}}^{\text{mix}},\mathbf{c})$	$(\tilde{\mathbf{t}}^{\text{mix}},\mathbf{s})$	$(\tilde{\mathbf{t}}^{\text{mix}},\mathbf{d})$
10dB	137.41	15.05	107.52	18.55
	14.62	4.82	11.28	8.34
20dB	194.71	15.13	123.59	19.94
	12.03	4.78	17.96	10.00
30dB	192.30	15.12	123.37	19.99
	12.12	4.77	18.05	10.04

27.4.2 NIST-Gas Data

In this section, the NIST-gas data set Δ shown in Figure 1.10 was used for experiments. It is the same data set considered in Kwan et al. (2006) and available at the National Institute of Standard Technology (NIST)'s website (webbook.nist.gov/chemistry). It contains spectral signature vectors of nine agents $\{\mathbf{s}_i\}_{i=1}^{9}$, eight of them, $\{\mathbf{s}_i\}_{i=2}^{9}$ are composed of 880 bands, and only one of them, $\mathbf{s}_1$, consists of 825 bands.

27.4.2.1 Signature Discrimination

In this subsection, the eight agent signature vectors $\{\mathbf{s}_i\}_{i=2}^{9}$ with the same number of bands were used as signature vectors to be discriminated. The agent $\mathbf{s}_1$ was removed from consideration because it has a different number of bands from those used in other eight signature vectors. As a result, the reference signature vector **r** for OSP-BPC was chosen to be the averaged signature vector over $\mathbf{s}_2$, $\mathbf{s}_3$, $\mathbf{s}_4$, $\mathbf{s}_5$, $\mathbf{s}_6$, $\mathbf{s}_7$, $\mathbf{s}_8$, and $\mathbf{s}_9$. Table 27.9 lists the bands prioritized by OSP-BPC according to (27.9), where the original set of bands, Ω was divided into two subsets of bands, $\Omega_{\mathbf{s}}^{\perp}$ and $\Omega_{\mathbf{s}}$ via (27.10).

Tables 27.10 tabulates the spectral similarity values of SID produced by VNVBS and full bands in discriminating among the eight different agent signatures of $\mathbf{s}_2$–$\mathbf{s}_9$ where VNVBS was implemented via (27.12) and the bands tabulated in Table 27.9.

Since Table 27.10 is symmetric, the SID values and RSDPW values are tabulated in the upper and lower entries of Table 27.10, respectively. As shown in Table 27.10, the signature discrimination performance was improved significantly by VNVBS over that produced by full bands.

To have better visual assessment, Figure 27.5 shows the graphical representation of Table 27.10 where the RSDPW values were graphically plotted as the y-axis against pairs of signature vectors along the x-axis in the order of ($\mathbf{s}_2$, $\mathbf{s}_3$), ($\mathbf{s}_2$, $\mathbf{s}_4$), . . . , ($\mathbf{s}_2$, $\mathbf{s}_9$), ($\mathbf{s}_3$, $\mathbf{s}_4$), . . . , ($\mathbf{s}_3$, $\mathbf{s}_9$), . . . , ($\mathbf{s}_8$, $\mathbf{s}_9$) with $\boldsymbol{\beta}$ specified by the reference signature vector **r**. For example, ($\mathbf{s}_j$,$\mathbf{s}_k$) in Figure 27.5 represents $\text{RSDPW}_{\text{SID}}(\mathbf{s}_j, \mathbf{s}_k;\boldsymbol{\beta})$ of SID comparing $\mathbf{s}_j$ against $\mathbf{s}_k$ with $\boldsymbol{\beta}$ set to the reference signature vector **r** which was the averaged signature vector over $\mathbf{s}_2$ to $\mathbf{s}_9$. As shown in Figure 27.5, SID using VNVBS clearly outperformed SID using full bands.

Table 27.9 $\Omega_{\mathbf{s}}^{\perp}$ and $\Omega_{\mathbf{s}}$ for eight agents $\{\mathbf{s}_i\}_{i=2}^{9}$

Agent	$\Omega_{\mathbf{s}}^{\perp}$	$\Omega_{\mathbf{s}}$
$\mathbf{s}_2$	599–636	1–598, 637–880
$\mathbf{s}_3$	61–69, 74–80, 190–196, 212–222, 309–317, 692–720	1–60, 70–73, 81–189, 197–211, 223–308, 318–691, 721–880
$\mathbf{s}_4$	28–51, 327–343	1–27, 52–326, 344–880
$\mathbf{s}_5$	127–132	1–126, 133–880
$\mathbf{s}_6$	1–9, 12–17, 262–281, 301–309, 318–324, 344–398, 403–407, 736–740, 742–880	10, 11, 18–261, 282–300 310–317, 325–343, 399–402, 408–735, 741
$\mathbf{s}_7$	138–146, 154–159, 383, 389–538, 554–597, 605–613	1–137, 147–153, 160–382 384–388, 539–553, 598–604, 614–880
$\mathbf{s}_8$	282–305	1–281, 306–880
$\mathbf{s}_9$	95–122, 182–189	1–94, 123–181, 190–880

Table 27.10 Discrimination among eight agent signatures using (27.12) and Table 27.9 with SID and RSDPW used as similarity measures on the upper and lower triangles

VNVBS full bands	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9
s_2	0 0	5.490 3.648	6.053 3.514	1.499 2.202	6.021 4.480	0.002 1.069	6.975 3.645	9.157 6.100
s_3	5.391 1.217	0 0	5.371 3.155	3.966 3.949	2.597 3.204	3.506 2.484	2.456 1.676	5.845 4.724
s_4	4.522 1.054	1.192 1.154	0 0	6.223 2.943	4.354 3.934	3.721 2.185	6.062 3.383	7.420 4.523
s_5	2.905 1.140	15.665 1.388	13.138 1.202	0 0	5.491 5.301	1.454 1.134	6.409 4.118	3.373 3.123
s_6	14.843 1.404	2.752 1.153	3.282 1.331	43.121 1.601	0 0	2.338 2.396	0.127 2.495	7.804 7.019
s_7	1.944 3.235	2.772 2.657	2.325 3.068	5.649 3.689	7.633 2.304	0 0	4.664 2.506	4.198 3.824
s_8	5.544 1.469	1.028 1.206	1.226 1.393	16.108 1.675	2.677 1.046	2.851 2.201	0 0	7.423 5.350
s_9	7.568 2.066	1.403 2.515	1.673 2.178	21.989 1.811	1.961 2.901	3.892 6.685	1.365 3.035	0 0

27.4.2.2 Signature Classification/Identification

Following the same treatment carried out in Section 27.4.1.2, we also simulated a mixed signature $\mathbf{t}^{\text{mix}}$ by uniformly mixing eight different agent signature vectors from $\mathbf{s}_2$ to $\mathbf{s}_9$, namely,

$$\begin{aligned}\mathbf{t}^{\text{mix}} &= 0.125 \cdot \mathbf{s}_2 + 0.125 \cdot \mathbf{s}_3 + 0.125 \cdot \mathbf{s}_4 + 0.125 \cdot \mathbf{s}_5 \\ &\quad + 0.125 \cdot \mathbf{s}_6 + 0.125 \cdot \mathbf{s}_7 + 0.125 \cdot \mathbf{s}_8 + 0.125 \cdot \mathbf{s}_9\end{aligned} \tag{27.16}$$

with the reference signature vector $\mathbf{r}$ chosen to be the same as that used in Section 27.4.2.1. As a result, in this particular case, the reference signature vector $\mathbf{r}$ turned out to be the same signature

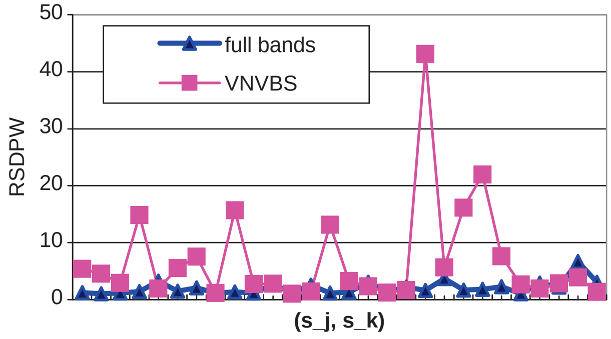


Figure 27.5 Comparison between RSDPW of VNVBS and full bands using SID for signature discrimination among eight agents.

Table 27.11 SID, rescaled SID, and RSDPW values between mixed signature and each component with and without VNVBS using bands obtained by (27.12)

VNVBS full bands	$(\mathbf{t}^{\text{mix}}, \mathbf{s}_2)$	$(\mathbf{t}^{\text{mix}}, \mathbf{s}_3)$	$(\mathbf{t}^{\text{mix}}, \mathbf{s}_4)$	$(\mathbf{t}^{\text{mix}}, \mathbf{s}_5)$	$(\mathbf{t}^{\text{mix}}, \mathbf{s}_6)$	$(\mathbf{t}^{\text{mix}}, \mathbf{s}_7)$	$(\mathbf{t}^{\text{mix}}, \mathbf{s}_8)$	$(\mathbf{t}^{\text{mix}}, \mathbf{s}_9)$
SID	2.17 1.68	0.007 1.38	1.84 1.60	2.73 1.92	0.003 1.20	0.001 0.52	2.37 1.15	3.93 3.48
Rescaled SID	2170 3.23	7 2.65	1840 3.08	2730 3.69	3 2.31	1 1	2370 2.21	3930 6.69
RSDPW	138.33 1.683	27.654 1.383	30.589 1.597	401.87 1.920	9.319 1.199	71.137 1.992	27.949 1.146	16.276 3.479

vector to be classified, $\mathbf{t}^{\text{mix}}$. Table 27.11 tabulates the spectral similarity values produced by SID between $\mathbf{t}^{\text{mix}}$ and each of eight agents $\{\mathbf{s}_i\}_{i=2}^{9}$ using full bands and VNVBS, where the upper and lower values in each entry of Table 27.11 were obtained by VNVBS and all the full 880 bands, respectively, with a smallest value highlighted by shade for classification.

Like the AVIRIS experiments conducted in Section 27.4.1, the signature discrimination performance was significantly improved using VNVBS compared to that produced using full bands. Figure 27.6 also showed the graphical representation of the RSDPW values obtained from Table 27.11 where the RSDPW values were graphically plotted as the y-axis against pairs of signatures $(\mathbf{t}^{\text{mix}},\mathbf{s}_k)$ with $k = 2, 3, \ldots, 9$ along the x-axis with $\boldsymbol{\beta}$ specified by the reference signature vector $\mathbf{r}$.

For example, $(\mathbf{t},\mathbf{s}_j)$ in Figure 27.6 represented the $\text{RSDPW}_{\text{SID}}(\mathbf{t}^{\text{mix}},\mathbf{s}_j;\boldsymbol{\beta})$ values produced by SID comparing $\mathbf{t}^{\text{mix}}$ against $\mathbf{s}_j$ with $\boldsymbol{\beta}$ set to the reference signature vector $\mathbf{r}$ which was the averaged signature vector over $\mathbf{s}_2$ to $\mathbf{s}_9$. Once again, as shown in Figure 27.6, VNVBS demonstrated its superior performance to that produced using full bands in terms of the rescaled values of SID as shown in the second row of Table 27.11. For example, the maximum SID ratio using full bands between eight pairs of $(\mathbf{t}^{\text{mix}}, \mathbf{s}_k)$ with $k = 2, 3, \ldots, 9$ was only 6.69 compared to 3930 produced by VNVBS.

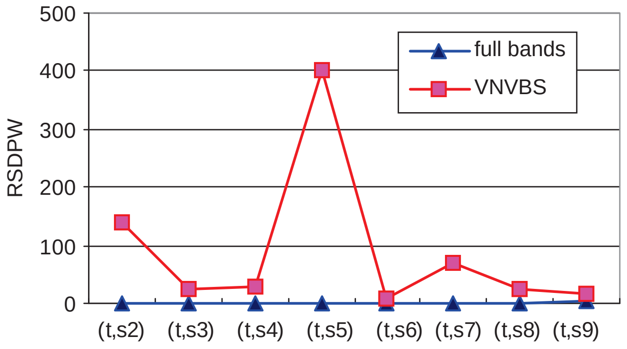


Figure 27.6 Comparison between RSDPW of VNVBS and full bands using SID for mixed signature classification.

Table 27.12 $\Omega_{\mathbf{s}}^{\perp}$ and $\Omega_{\mathbf{s}}$ generated by Approach 1 from $\mathbf{s}_1$ to $\mathbf{s}_9$

S	$\Omega_{\mathbf{s}}^{\perp}$	$\Omega_{\mathbf{s}}$
$\mathbf{s}_1$	596	1–595, 597–825
$\mathbf{s}^*{}_2$	581–611	1–580, 612–825
$\mathbf{s}^*{}_3$	37–44, 49–55, 165–171, 187–197, 284–292, 667–695	1–36, 45–48, 56–164, 172–186, 198–283, 293–666, 696–825
$\mathbf{s}^*{}_4$	2–26, 302–318	1, 27–301, 319–825
$\mathbf{s}^*{}_5$	102–107	1–101, 108–825
$\mathbf{s}^*{}_6$	237–256, 275–284, 293,300, 319–382, 705, 710–825	1–236, 257–274, 285–292, 294–299, 301–318, 383–704, 706–709
$\mathbf{s}^*{}_7$	113–121, 129–134, 358, 364–366, 368–506	1–112, 122–128, 135–357, 359–363, 367, 507–825
$\mathbf{s}^*{}_8$	257–280	1–256, 281–825
$\mathbf{s}^*{}_9$	70–79, 82–86, 89–97, 157–164	1–69, 80–81, 87–88, 98–156, 165–825

27.4.2.3 Signature Discrimination between Two Signatures with Different Numbers of Bands

One of the strengths of VNVBS is its ability to discriminate two signature vectors with different numbers of bands which was not found in the above AVIRIS experiments. To demonstrate this advantage, the agent $\mathbf{s}_1$ with 825 bands was used for this purpose to be compared against the other eight agents, $\mathbf{s}_2$–$\mathbf{s}_9$ with 880 bands. In doing so, two approaches were considered. One was to extract bands from the signature vector with a larger number of bands to match the same number of bands used by the signature vector with a smaller number of bands. As an opposite to the first approach, a second approach was to expand the signature vector with a smaller number of bands to match the same number of bands used by the signature vector with a larger number of bands by zero-padding in the missing bands. These two approaches were considered in the following experiments.

Extracting 825 Bands from the Original 880 Bands for $\mathbf{s}_2$–$\mathbf{s}_9$ with the Same Wavelength Coverage as $\mathbf{s}_1$

In this case, the reference signature $\mathbf{r}$ for VNVBS was chosen by averaging from $\mathbf{s}_1$ to $\mathbf{s}_9^*$, where $\mathbf{s}_k^*$ ($k = 2, 3, \ldots, 9$) denoted the signature with 825 bands extracted from the kth signature with original 880 bands. In this case, the $\mathbf{r}$ had only 825 bands. Table 27.12 lists the bands prioritized by the VNVBS according to (27.9), where the original set of bands was divided into two subsets of bands, $\Omega_{\mathbf{s}}^{\perp}$ and $\Omega_{\mathbf{s}}$ via (27.10).

Table 27.13 tabulates the SID-generated spectral similarity values obtained by VNVBS and full bands, respectively, for comparison where VNVBS also outperformed the use of full bands. Similar to Table 27.10, the SID values and RSDPW values are tabulated in the upper and lower triangles of Table 27.13, respectively.

Expanding Original 825 Bands to 880 Bands Through Zero Padding

In this case, $\hat{\mathbf{s}}_1$ denoted the signature vector $\mathbf{s}_1$ after interpolating the original 825 bands to 880 bands by zero padding. Similarly, Table 27.14 lists the bands prioritized by VNVBS according to (27.9), where the original set of bands was divided into two subsets of bands, $\Omega_{\mathbf{s}}^{\perp}$ and $\Omega_{\mathbf{s}}$ via (27.10).

Table 27.13 Discrimination among nine agent signatures from $\mathbf{s}_1$ to $\mathbf{s}^*_9$ obtained by VNVBS and full bands

VNVBS full bands	$\mathbf{s}_1$	$\mathbf{s}^*_2$	$\mathbf{s}^*_3$	$\mathbf{s}^*_4$	$\mathbf{s}^*_5$	$\mathbf{s}^*_6$	$\mathbf{s}^*_7$	$\mathbf{s}^*_8$	$\mathbf{s}^*_9$
$\mathbf{s}_1$	0 0	0.18 0.63	5.08 4.80	7.77 4.95	6.34 4.24	6.01 5.51	3.27 2.17	8.11 4.82	9.74 8.96
$\mathbf{s}^*_2$	1.01 1.72	0 0	5.29 3.62	6.14 3.51	1.67 2.17	5.94 4.40	1.38 1.04	7.26 3.67	8.94 6.08
$\mathbf{s}^*_3$	4.10 1.79	4.14 1.04	0 0	5.36 3.17	3.96 4.00	259 2.97	3.31 2.48	2.45 1.70	5.84 4.73
$\mathbf{s}^*_4$	3.61 1.62	3.65 1.05	1.13 1.10	0 0	6.15 2.99	4.00 3.56	3.52 2.15	6.04 3.43	7.41 4.48
$\mathbf{s}^*_5$	3.84 1.33	3.79 1.28	15.75 1.34	13.87 1.21	0 0	5.72 5.25	1.19 1.30	6.40 4.20	3.37 3.06
$\mathbf{s}^*_6$	10.93 1.99	11.05 1.16	2.66 1.11	3.02 1.22	42.00 1.49	0 0	2.23 2.36	0.18 2.16	7.80 6.75
$\mathbf{s}^*_7$	1.23 5.30	1.24 3.08	3..33 2.95	2.93 3.26	4.72 3.96	8.88 2.65	0 0	4.99 2.54	3.06 3.74
$\mathbf{s}^*_8$	4.17 2.12	4.22 1.23	1.01 1.18	1.15 1.30	16.04 1.59	2.61 1.06	3.39 2.49	0 0	7.42 5.36
$\mathbf{s}^*_9$	5.73 1.41	5.79 2.43	1.39 2.53	1.58 2.30	22.02 1.89	1.90 2.82	4.65 7.50	3.37 3.01	0 0

Table 27.15 tabulates SID-generated spectral similarity values using VNVBS and full bands respectively for comparison. Like Table 27.13, the SID values and RSDPW values were tabulated in the upper and lower entries of Table 27.15, respectively, where VNVBS also demonstrated its advantage by selecting bands compared to full bands.

Table 27.14 $\Omega_{\mathbf{s}}^{\perp}$ and $\Omega_{\mathbf{s}}$ generated by Approach 2 from $\mathbf{s}_1$ to $\mathbf{s}_9$

$\mathbf{s}$	$\Omega_{\mathbf{s}}^{\perp}$	$\Omega_{\mathbf{s}}$
$\hat{\mathbf{s}}_1$	1–23, 620–621, 853–880	24–619, 622–852
$\mathbf{s}_2$	606–613, 624–636	1–605, 614–623, 637–880
$\mathbf{s}_3$	61–69, 74–80, 190–196, 212–222, 309–317, 692–720	1–60, 70–73, 81–189, 197–211, 223–308, 318–691, 721–880
$\mathbf{s}_4$	28–51, 327–343	1–27, 52–326, 344–880
$\mathbf{s}_5$	127–132	1–126, 133–880
$\mathbf{s}_6$	1–18, 262–281, 301–309, 318–325, 344–399, 403–407, 735–880	19–261, 282–300, 310–317, 326–343, 400–402, 408–734
$\mathbf{s}_7$	138–146, 154–159, 383, 389–531	1–137, 147–153, 160–382, 384–388, 532–880
$\mathbf{s}_8$	282–305	1–281, 306–880
$\mathbf{s}_9$	95–122, 182–189	1–94, 123–181, 190–880

Table 27.15 Discrimination among nine agent signatures from $\hat{\mathbf{s}}_1$ to $\mathbf{s}_9$ obtained by VNVBS and full bands

VNVBS full bands	$\hat{\mathbf{s}}_1$	$\mathbf{s}_2$	$\mathbf{s}_3$	$\mathbf{s}_4$	$\mathbf{s}_5$	$\mathbf{s}_6$	$\mathbf{s}_7$	$\mathbf{s}_8$	$\mathbf{s}_9$
$\hat{\mathbf{s}}_1$	0 0	0.20 0.62	5.85 4.77	7.72 4.86	5.38 4.17	∞ 5.20	3.20 2.13	8.03 4.77	10.5 8.95
$\mathbf{s}_2$	130.96 1.65	0 0	5.29 3.64	6.15 3.50	1.67 2.19	5.86 4.47	1.38 1.05	7.26 3.64	8.94 6.05
$\mathbf{s}_3$	31.57 1.72	4.14 1.04	0 0	5.37 3.14	3.96 3.93	2.59 3.12	3.31 2.45	2.45 1.67	5.84 4.71
$\mathbf{s}_4$	37.21 1.52	3.51 1.07	1.17 1.13	0 0	6.22 2.91	4.30 3.81	3.54 2.14	6.06 3.37	7.41 4.49
$\mathbf{s}_5$	497.44 1.29	3.79 1.27	15.75 1.33	13.36 1.18	0 0	5.41 5.29	1.19 1.31	6.40 4.11	3.37 3.10
$\mathbf{s}_6$	11.63 1.94	11.25 1.17	2.71 1.12	3.19 1.27	42.74 1.50	0 0	2.17 2.38	0.12 2.29	7.79 6.81
$\mathbf{s}_7$	105.25 5.19	124 3.14	3.33 3.00	2.82 3.39	4.73 4.01	9.04 2.66	0 0	4.99 2.49	3.06 3.74
$\mathbf{s}_8$	31.00 2.07	4.22 1.25	1.01 1.20	1.20 1.35	16.04 1.60	2.66 1.06	3.39 2.49	0 0	7.42 5.33
$\mathbf{s}_9$	22.58 1.45	5.79 2.40	1.39 2.52	1.64 2.22	22.02 1.88	1.94 2.83	4.65 7.56	3.37 3.03	0 0

According to Tables 27.13 and 27.15, the two approaches did not have very much impact on SID when full bands were used, but their discriminatory powers calculated for the rescaled SID values and RSDPW values we significantly improved when VNVBS was used for SID.

One final comment is noteworthy. Two approaches proposed for VNVBS in Section 27.4.2.3 are specifically designed for signature vectors but not images to which standard BS techniques such as Mausel et al. (1990), Conese and Maselli (1993), Stearns et al. (1993), Chang et al. (1999), and Huang and He (2005), which are not applicable because the latter cannot process a hyperspectral image whose image pixels vary with different number of bands. OSP-BPS allows users to implement VNVBS for spectral characterization. To the author's best knowledge, there are no existing standard BS techniques that can be used to select bands from a single hyperspectral signature vector as what VNVBS does. Such benefit derived from VNVBS cannot be gained by any image-based band selection.

By concluding this section, it is worth noting that there are significant differences between two data sets, CB data and AVIRIS data. So, VNVBS also performed quite differently for both data sets. The $\Omega_{\mathbf{s}}^{\perp}$ obtained for gas data tended to capture the peaks, while the $\Omega_{\mathbf{s}}^{\perp}$ obtained for hyperspectral AVIRIS data attempted to capture the edges of the spectral signature vectors. This is because the peaks in the gas data and edges in the AVIRIS data are their most distinct spectral features in their local spectral profiles. The VNVBS is particularly designed to characterize such local properties of a signature vector.

27.5 Selection of Reference Signatures

As noted, the selection of the reference signature vector **r** had significant impact on the performance. To address this issue, two general guidelines of how to select the reference signature vector **r** based on our extensive experiments may be helpful.

1. Since the signature vector **r** is used as a reference signature vector between two signature vectors $\mathbf{s}_1$ and $\mathbf{s}_2$, it should have some degree of correlation associated with both signature vectors. Keeping this in mind, when spectral feature characterization is performed such as signature discrimination via a database/spectral library, the best reference signature vector to be selected is the average of all signature vectors in the database/spectral library so that any pair of the two signature vectors drawn from the database/spectral library to be characterized can have some correlation with the averaged signature vector. On the other hand, if a reference signature vector is selected from the database/spectral library, the performance will be determined by how close a signature vector is related to the other signature vector coming from the same database, which is to be compared against. As a consequence, the performance may yield complete different results. Such evidence was demonstrated in Section 27.4.
2. By contrast, if a mixed signature vector is to be classified/identified through a database/spectral library which comprises each mixing component, then the best candidate for the reference signature vector **r** is the mixed signature vector itself. This is because the mixed signature vector itself already provides the desired correlation for the classification/identification between this mixed signature vector and each of its mixing components. Using the average of signature vectors may only smear the spectral characteristics. This was also confirmed by the experiments in Section 27.4.1.2.

27.6 Conclusions

This chapter presents a new approach to BS, called VNVBS for a single hyperspectral signature. Unlike most band selection techniques which are designed for images, VNVBS is designed for characterization of a single hyperspectral signature vector without a need of image sample correlation. To select appropriate bands, an OSP-BPC is also developed, which decomposes the original spectral signature vector into two orthogonal components that can be used for spectral characterization. Experimental results demonstrate that VNVBS is more effective in preserving information for hyperspectral signature characterization than that using full band information in hyperspectral signature characterization and analysis.

28

Kalman Filter-Based Estimation for Hyperspectral Signals

Most popular and widely used approaches in statistical signal estimation are mean squared error (MSE) based approaches among which Kalman filtering (KF) is the most powerful and effective technique that can be implemented in real time under a nonstationary environment. Recently, a Kalman filtering approach to linear spectral unmixing, called Kalman filter-based linear spectral unmixing (KFLU) was developed for mixed pixel classification by Chang and Brumbley (1999a, 1999b). However, its applicability to spectral characterization for spectral estimation, identification, and quantification has not been explored. This chapter presents new applications of KF in spectral estimation, identification, and abundance quantification for which three Kalman filter (KF)-based spectral characterization signal processing (KFSCSP) techniques are developed. These techniques are completely different from KFLU in the sense that the former performs a Kalman filter across a spectral coverage wavelength by wavelength (i.e., band-by-band) as opposed to the latter, which implements a Kalman filter pixel vector by pixel vector throughout an entire image cube. In addition, the proposed Kalman filter-based techniques do not require a linear mixture model as KFLU does. Accordingly, they are not linear spectral unmixing methods but rather spectral signature filters operating as if they are spectral measures. More specifically, the state equation implemented in KFLU is designed to keep track of pixel-to-pixel correlation present in a hyperspectral image cube, whereas the state equation used by KFSCSP techniques is designed to capture the band-to-band correlation within a single signature vector. As a result, KFSCSP techniques can be used for both laboratory data and non-image data analysis compared with KFLU, which is primarily developed for hyperspectral imagery. This is due to the fact that laboratory data do not usually have pixel-to-pixel correlation of which KFLU can take advantage, but they do have band-to-band correlation that can be taken into account by KFSCSP techniques. Also, KFSCSP techniques do not require a linear mixture model as does KFLU, which assumes that image pixels are linearly mixed by a number of image endmembers. Therefore, KFSCSP techniques do not need image endmembers to form a linear mixture model for their implementation. Accordingly, they would rather be considered as spectral signature filters.

28.1 Introduction

Kalman filtering (KF) has been widely used in statistical signal processing for the purpose of parameter estimation (Poor, 1994). It makes use of a measurement equation (input/output

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

equation) and a state equation (process equation) to recursively estimate parameters of the state. A new application of KF in linear spectral unmixing, called Kalman filter-based linear spectral unmixing (KFLU), was recently developed by Chang and Brumbley (1999a, 1999b) for mixed pixel classification. When a Kalman filter is implemented for linear spectral unmixing as a mixed pixel classifier, the state vector **x** in the state equation is specified by the abundance vector $\boldsymbol{\alpha}$ present in an image, and the pixel vector **r** is specified by another equation called measurement equation from which $\boldsymbol{\alpha}$ can be estimated. With the measurement equation, a Kalman filter takes advantage of a linear mixture model commonly used in linear spectral unmixing to describe how a pixel vector **r** is linearly mixed via a measurement equation. By recursively implementing these two equations, KFLU generally produces better estimates of abundance fractions than other linear spectral unmixing methods (Chang and Brumbley, 1999a, 1999b). Several advantages resulting from the use of a Kalman filter are obvious for remote sensing image analysis. One is its ability to deal with nonstationary data, which are generally true in remotely sensed imagery. Another is its recursive structure that makes real-time processing feasible. A third advantage is its utilization of two equations, measurement and state equations, which can be implemented in various forms to accomplish different tasks such as smoothing, filtering, and prediction (Gelb, 1974).

This chapter re-invents the wheel by developing Kalman filter KF-based techniques for spectral signature characterization rather than KFLU that is used for linear spectral unmixing. It explores several new applications of KF in hyperspectral signature analysis for spectral estimation, identification, and quantification by re-deriving the measurement and state equations, which result in various techniques referred to as Kalman filter-based spectral characterization signal processing (KFSCSP) techniques. Several prominent differences between KFLU developed by Chang and Brumbley (1999a, 1999b) and KFSCSP techniques developed in this chapter are noteworthy and described in the following.

In KFLU, the measurement equation is described by a linear mixture model used by linear spectral mixture analysis (LSMA) and the state equation is included to perform abundance vector estimation for the image endmembers, which are used to form a linear mixture model for spectral unmixing. Therefore, KFLU actually operates on a hyperspectral image cube and performs image classification by estimating abundance vectors via its state equation, while each image pixel vector in the image cube serves as an input to the measurement equation. On the contrary, KFSCSP techniques are rather different from KFLU in the sense that KFSCSP technique is a one-dimensional (1D) signal-processing technique that exploits spectral variability of a single signature vector across its spectral coverage to perform spectral analysis without referencing other data sample vectors. As a result, the sample correlation used in the state equation by KFLU is not available in KFSCSP techniques. So, if a KFSCSP technique is applied to an image cube, it only considers each pixel vector as a single and independent signature vector without accounting for sample spectral correlation. It then serves as a spectral signature filter rather than as an image classifier performed by KFLU. Another significant difference is how the measurement equation is interpreted. KFLU requires prior knowledge of image endmembers to define the measurement equation for spectral unmixing. This is no longer true for a KFSCSP technique, because the data to be processed by a KFSCSP technique are a single signature vector, not an image cube in which there is no linear mixing involved. Additionally, a KFSCSP technique can be implemented solely as a signature estimator, an identifier, or an abundance quantifier compared with KFLU, which is implemented as an abundance vector estimator with image endmembers assumed to be known *a priori*. Finally, the state equation used in KFLU is designed for linear spectral unmixing to keep track of the changes in abundance fractions of image endmembers pixel vector by pixel vector. However, the state equation used in a KFSCSP technique is designed to capture spectral variations within a signature vector across its spectral range wavelength by wavelength. Therefore, throughout this chapter, the term "signature vector" is used instead of "pixel vector" to reflect the distinction between different types of data processed by a KFSCSP technique and KFLU.

28.2 Kalman Filter-Based Linear Unmixing

KF has been widely used in statistical signal processing for parameter estimation (Gleb, 1974). It makes use of a measurement equation (input/output equation) and a state equation (process equation) to recursively estimate parameters of states. When a Kalman filter is implemented for spectral unmixing as a mixed pixel classifier, the state vector $\mathbf{x}$ in an equation, called state equation, is specified by the abundance vector $\boldsymbol{\alpha}$ present in an image pixel vector $\mathbf{r}$, which is specified by another equation called measurement equation. With this formulation, a Kalman filter takes advantage of a linear mixing model commonly used in spectral unmixing to describe how a pixel vector $\mathbf{r}$ is linearly mixed via a measurement equation. Meanwhile, KFLU includes an additional equation, state equation, which is absent in spectral unmixing to estimate the abundance fractions of the abundance vector of the currently processed pixel vector in an image based on previously processed pixels. Implementing these two equations recursively, KFLU generally produces better estimates of abundance fractions than spectral unmixing methods.

This and the following sections present several new applications of KF in spectral estimation, identification, and quantification for hyperspectral signature characterization by re-deriving the measurement and state equations, which results in several techniques referred to as KFSCSP techniques. There are important and salient differences between KFLU and KFSCSP techniques that are worth mentioning. In KFLU, the measurement equation is described by a linear mixing model formed by a set of image endmembers and the state equation is included to perform abundance vector estimator of these image endmembers via spectral unmixing. Therefore, KFLU operates on a hyperspectral image as an image cube and performs image classification using the estimated abundance vector produced by its state equation where each image pixel vector is considered as an input to the measurement equation. Quite oppositely, KFSCSP technique performs quite differently from KFLU in the sense that it is a one-dimensional (1D) signal-processing technique that explores spectral variability within a single signature vector across the spectral range band by band where there are no sample vectors involved such as sample covariance matrix used in KFLU. As a result, the sample correlation is not available in KFSCSP technique. So, if it is applied to an image cube, it considers a pixel vector as a single signature vector without accounting for sample correlation and performs as a spectral signature filter rather than an image classifier as KFLU does. Another significant difference is the use of the measurement equation. KFLU requires prior knowledge of image endmembers to form a linear mixture model to implement the measurement equation for spectral unmixing. This is not true for KFSCSP technique because the data to be processed are signature vector and not an image cube, and there is no need of a linear mixture model. Additionally, KFSCSP technique can be implemented as a signature estimator, an identifier, and an abundance quantifier compared with spectral unmixing methods including KFLU, which can only be implemented as an abundance vector estimator for image endmembers assumed to be known *a priori*. Finally, since KFSCSP technique is not developed for image classification, the state equation used in spectral unmixing to keep track of changes in abundance fractions of image endmembers pixel vector by pixel vector is not applicable. Instead, the state equation used in KFSCSP technique is designed to capture spectral variation of a signature vector across its spectral coverage wavelength by wavelength.

Let $\mathbf{r}(x,y)$ be the location of a multispectral/hyperspectral imager pixel vector at the spatial coordinate of (x,y), and L be the total number of spectral channels (bands). Then, $\mathbf{r}(x,y)$ is an $L \times 1$

column vector. We further assume that an **r**(x,y) can be modeled by a linear mixture of p image endmembers, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ which are present in the image with appropriate abundance fractions, $\alpha_1, \alpha_2, \ldots, \alpha_p$ with α_j corresponding to the abundance fraction of the jth endmember $\mathbf{m}_j$ as follows:

$$\mathbf{r}(x, y) = \mathbf{M}\boldsymbol{\alpha}(x, y) + \mathbf{u}(x, y) \tag{28.1}$$

where **u**(x,y) is a measurement or model error introduced at **r**(x,y) and $\mathbf{M} = \left[\mathbf{m}_1\mathbf{m}_2 \cdots \mathbf{m}_p\right]$ is the image endmember matrix formed by $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ and can be viewed as a measurement matrix. It should be noted that the endmember matrix remains invariant and independent of the spatial location of **r**(x,y). However, the abundance fractions $\alpha_1, \alpha_2, \ldots, \alpha_p$ are the parameters that vary with pixel vector by pixel vector and need to be estimated. In order to simplify discussions, we represent the image pixel vector **r**(x,y) by **r**(k), with the parameter k indicating the position of **r**(x,y) that is processed in a top-down and left-right fashion. As a result, its corresponding abundance vector will be denoted by $\boldsymbol{\alpha}(k)$. One of major strengths of a Kalman filter is that it introduces a so-called state equation, which keeps track of changes in states during their transitions and is absent in spectral unmixing methods. If we interpret a state at position k as an abundance vector $\boldsymbol{\alpha}(k)$, then the state equation allows us to capture changes in abundance fractions residing in the image pixel vector **r**(k) at the position k to the image pixel vector $\mathbf{r}(k+1)$ at the position $k+1$. More specifically, we can model the state equation by

$$\boldsymbol{\alpha}(k+1) = \boldsymbol{\Phi}(k+1, k) \cdot \boldsymbol{\alpha}(k) + \mathbf{v}(k) \tag{28.2}$$

where $\boldsymbol{\Phi}(k+1, k)$ is an $L \times L$ transition matrix from the pixel vector at the position k to the pixel vector at the position $k+1$, and **v**(k) can be considered as a state noise or model error at the position k. In order to implement a Kalman filter, we assume that the noise **u**(k) in (28.1) is an additive white noise given by

$$E\left[\mathbf{u}(n)\mathbf{u}^T(k)\right] = \mathbf{R} = \sigma_{\mathbf{u}}^2 \cdot \delta_{nk}\mathbf{I}_{L\times L} \tag{28.3}$$

where δ_{nk} is Kronecker's notation defined by $\delta_{kk} = 1$ for $n = k$ and $\delta_{nk} = 0$ for $n \neq k$. Similarly, we can also assume that the noise **v**(k) in (28.2) is also an additive white noise given by

$$E\left[\mathbf{v}(n) \cdot \mathbf{v}^T(k)\right] = \mathbf{Q} = \sigma_{\mathbf{v}}^2 \cdot \delta_{nk}\mathbf{I}_{p\times p} \tag{28.4}$$

Using (28.1)–(28.4), a Kalman filter can perform three operations: smoothing, filtering, and prediction, of which smoothing and filtering can be essentially derived from prediction by Gelb (1974). Since we are only interested in filtering and prediction that can be used to estimate spectral signatures, only these two operations will be reviewed and briefly discussed in the following. For all the details of KF implementation, we refer to tGelb (1974). We assume that k is the spatial position of the image pixel vector currently being processed and the observed image vectors described by (28.1) are available up to k, $\mathbf{r}(i)|_{i=1}^{k}$. Let $\hat{\boldsymbol{\alpha}}(k+1|k)$ denote the minimum MSE prediction of the abundance vector $\boldsymbol{\alpha}(k)$ at the position $k+1$ and $\hat{\boldsymbol{\alpha}}(k|k)$ denote the minimum MSE estimate of the abundance vector $\boldsymbol{\alpha}(k)$ at the position k based on all the image pixel vectors $\mathbf{r}(i)|_{i=1}^{k}$ that are already visited up to the position k. Similarly, $P(k+1|k)$ and $P(k+1|k)$ represent the one-step MSE prediction error covariance matrix and MSE estimation error covariance matrix incurred at the position k. Then, a Kalman filter that performs linear spectral unmixing can be described in the following procedures:

- Initial conditions

  $\mathbf{R}$ and $\mathbf{Q}$,

  $$\hat{\boldsymbol{\alpha}}(0|-1) = E[\boldsymbol{\alpha}(0)] \text{ is the mean of } \boldsymbol{\alpha}(0)$$

  $$\mathbf{P}(0|-1) = Cov[\boldsymbol{\alpha}(0)] \text{ is the covariance matrix of } \boldsymbol{\alpha}(0)$$

- Kalman gain at k

  $$\mathbf{K}(k) = \left[\boldsymbol{\Phi}(k+1,k)\mathbf{P}(k|k-1)\mathbf{M}^T\right]\left[\mathbf{M}\mathbf{P}(k|k-1)\mathbf{M}^T + \mathbf{R}\right]^{-1}$$

- Abundance update at k

  $$\hat{\boldsymbol{\alpha}}(k|k) = \hat{\boldsymbol{\alpha}}(k|k-1) + \mathbf{K}(k)[\mathbf{r}(k) - \mathbf{M}\hat{\boldsymbol{\alpha}}(k|k-1)]$$

- Error measurement update at k

  $$\mathbf{P}(k|k) = [\mathbf{I} - \mathbf{K}(k)\mathbf{M}]\mathbf{P}(k|k-1)$$

- Abundance prediction at $k+1$

  $$\hat{\boldsymbol{\alpha}}(k+1|k) = \boldsymbol{\Phi}(k+1,k)\hat{\boldsymbol{\alpha}}(k|k)$$

- Error measurement prediction at $k+1$

  $$\mathbf{P}(k+1|k) = \boldsymbol{\Phi}(k+1,k)\mathbf{P}(k|k)\boldsymbol{\Phi}^T(k+1,k) + \mathbf{Q}$$

So, KFLU performs abundance fraction estimation for mixed pixel classification using an image cube with its measurement equation described by the kth image pixel vector $\mathbf{r}(k)$ and state equation characterized by the abundance vector $\boldsymbol{\alpha}(k)$ present in the $\mathbf{r}(k)$. Therefore, the abundance vector $\boldsymbol{\alpha}(k)$ is estimated via a recursive algorithm between the measurement equation and state equation by running through all image pixel vectors, $\mathbf{r}(1)$, $\mathbf{r}(2), \ldots$ in the entire image cube. In what follows, we present three KFSCSP techniques that only deal with signature vectors to perform spectral estimation, identification, and quantification rather than spectral classification performed by KFLU, which needs knowledge of image endmembers to perform classification.

28.3 Kalman Filter-Based Spectral Characterization Signal-Processing Techniques

Three KFSCSP techniques are derived and presented in this section, called Kalman filter-based spectral signature estimator (KFSSE), Kalman filter-based spectral signature identifier (KFSSI), and Kalman filter-based spectral signature quantifier (KFSSQ). In KFSSE, the input and output of its measurement equation are specified by a noise-corrupted signature vector and its true signature vector to be estimated, respectively. KFSSE then uses a state equation to predict spectral values of the true signature vector across its spectral coverage via a signal model such as the Gaussian–Markov model. So, the role of KFSSE is to capture spectral signature changes between adjacent spectral bands compared with KFLU, which is developed to capture changes in abundance fractions between two adjacent pixel vectors. Most importantly, as noted previously, KFSSE does not need a linear mixture model as required by KFLU. Therefore, there is no need for KFSSE to find image endmembers to form a linear mixture model. On the other hand, KFSSI is developed to identify a signature vector via a matching signature vector chosen from a known database or spectral library. In doing so, KFSSI is derived from KFSSE by replacing the true signature vector used in KFSSE with an auxiliary signature vector that enables it to capture the matching signature vector in identifying the unknown signature vector. According to functionality, both KFSSE and KFSSI are developed as signature vector estimators, but not abundance vector estimators as the way that KFLU is originally designed. This is because no linear mixture model is used to unmix abundance fractions of image endmembers. Unlike KFSSE or KFSSI, KFSSQ can be considered as a follow-up signature filter. It models its state equation as a zero-holder interpolator by taking a KFSSE-estimated signature vector, that is, spectral estimate in its measurement equation as a system gain vector to achieve spectral quantification of the estimated signature vector. As a result, the quantification of the spectral signature value at the current lth band is used as a prediction of the spectral value at the next adjacent band, the $(l+1)$st band. Its ability in spectral quantification makes KFSSQ particularly useful in applications of chemical/biological (CB) defense where the lethal level of a CB agent is determined by its concentration (Wang *et al.*, 2004), and the collected samples are not necessarily correlated as image pixel vectors in an image cube. Therefore, the quantification of a CB agent is crucial in damage control assessment. Finally, in order to demonstrate the utility of the three KFSCSP techniques in spectral estimation, identification, and abundance quantification, experiments including computer simulations and real data are conducted for performance analysis.

28.3.1 Kalman Filter-based Spectral Signature Estimator

We assume that $\mathbf{t} = (t_1, t_2, \ldots, t_L)^T$ is a true signature vector to be estimated and $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ is an observable signature vector from which the ture signature vector $\mathbf{t}$ can be estimated. Since the measurement equation and the state equation described by (28.1) and (28.2) are developed for image classification, they do not directly meet our current requirements and must be modified as follows:

$$\text{Measurement equation}: \; r_l = c_l t_l + u_l \tag{28.5}$$

$$\text{State equation}: \; t_{l+1} = \phi(l+1, l) t_l + v_l \tag{28.6}$$

where the index l denotes the lth band, $\mathbf{c} = (c_1, c_2, \ldots, c_L)^T$ is a system gain vector, $\phi(l+1, l)$ is a transition parameter from the lth band to the $l+1$st band, and $\mathbf{u} = (u_1, u_2, \ldots, u_L)^T$ and $\mathbf{v} = (v_1, v_2, \ldots, v_L)^T$ are white noise vectors, all of which must be determined *a priori*. According to Chang and Brumbley (1999a, 1999b), in order to implement (28.5) and (28.6) recursively, the initial condition to start the recursive algorithm between measurement and state equations is set to $\hat{t}_1 = 0$ by assuming that the estimate of true spectral signature at the first band is 0. The KF is then implemented recursively until it reaches the last band (see recursive formulas (6–10) in Chang and Brumbley (1999a)). Comparing (28.5) and (28.6) to (28.1) and (28.2), there are several salient differences between KFLU and KFSSE. First of all, KFLU uses (28.1) and (28.2) to model the same image vector. That is, (28.1) represents the status of the image pixel $\mathbf{r}$, which is linearly mixed by a set of p known image endmembers, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ and (28.2) specifies its corresponding abundance vector $\boldsymbol{\alpha} = \left(\alpha_1, \alpha_2, \ldots, \alpha_p\right)^T$ that is unknown but needs to be estimated via (28.1). Therefore, KFLU requires the prior knowledge of image endmembers to be used in the linear mixture model (28.1). As a rather different approach, KFSSE is developed to estimate a target signature vector $\mathbf{t}$ rather than an abundance vector $\boldsymbol{\alpha}$. It consideres a true signature vector $\mathbf{t} = (t_1, t_2, \ldots, t_L)^T$ as a state vector in (28.6) and estimates the state vector band by band via an observable vector $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ specified by the measurement equation (28.5). As a result, the variables in both equations (28.5) and (28.6) are scalars and represent spectral signature values of the lth band of the target signature vector $\mathbf{t}$ within the observed signature vector $\mathbf{r}$. In this case, the measurement equation (28.5) is simply a true signature vector corrupted by noise $\mathbf{u} = (u_1, u_2, \ldots, u_L)^T$ not a linear mixture model specified by (28.1). Another major difference is that both (28.5) and (28.6) are linear functions of the same lth band of the true spectral signature $\mathbf{t}$, t_l. This is different from (28.1) to (28.2), which are linear functions of two distinct vectors, image vector $\mathbf{r}(k)$ and abundance vector $\boldsymbol{\alpha}(k)$ with different dimensions L and p, respectively. Therefore, compared with KFLU, in which input and output are an image pixel vector $\mathbf{r}(k)$ and an estimate of the abundance vector $\boldsymbol{\alpha}(k)$ and $\hat{\boldsymbol{\alpha}}^{\text{KFLU}}(k)$, respectively, the input of KFSSE is simply the lth band spectral value of the observable spectral signature vector $\mathbf{r}$, r_l, and its output is an estimate of t_l, which is the lth band spectral value of true signature vector $\mathbf{t}$, denoted by $\hat{t}_l^{\text{KFSSE}}$. Furthermore, the state in (28.6) is designed to predict the spectral value of the $(l+1)$st band of the true signature vector $\mathbf{t}$ by updating the currently estimated spectral value of its lth band accoridng to the spectral variation between two adjacent bands. This is quite different from the state in (28.2), which is included to keep track of changes in abundance vectors between two adjacent image pixel vectors. Besides, (28.6) models the relationship of the spectral signature values between one wavelength and the next adjacent wavelength as a Gaussian–Markov process specified by the state transition parameter $\phi(l+1, l)$ and the additive Gaussian noise, $\mathbf{v} = (v_1, v_2, \ldots, v_L)^T$. If the variance σ_{v}^2 or standard deviation of $\mathbf{v}$ is set too low, then the state equation may not be effective enough to capture real variations in spectral values of one material. Accordingly, throughout the chapter, the standard deviation of the state noise is always assumed to be high.

28.3.2 Kalman Filter-Based Spectral Signature Identifier

In the previous section, KFSSE is developed to estimate a true signature vector $\mathbf{t}$ through an observed signature vector $\mathbf{r}$. By remodeling its measurement and state equations, KFSSE can be re-interpretated as a spectral signature identifier, which is referred to as KFSSI. In order to perform spectral signature idenitification, it always assumes that there is a database or spectral signature library available for this purpose, denoted by $\Delta = \{\mathbf{s}_k\}_{k=1}^{K}$, which is made up of K spectral signature vectors, $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k$.

We suppose that the observable spectral signature vector is $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$, which needs to be identified via a data base or spectral library, $\Delta = \{\mathbf{s}_k\}_{k=1}^{K}$. The idea is to use the measurement equation to describe the observable spectral signature vector $\mathbf{r}$ as a noise-corrupted matching spectral signature vector $\mathbf{s}_k$ selected from Δ. An auxiliary spectral signature vector $\mathbf{t}$ is then introduced in the state equation to model the state that describes the ability of identifying a given matching signature vector $\mathbf{s}$ in the observable signature vector $\mathbf{r}$. In other words, the identification of the observable spectral signature vector $\mathbf{r}$ is performed by finding a particular spectral signature vector $\mathbf{s}_k$ from the Δ that best matches $\mathbf{r}$. In this case, the measurement equation (28.5) and the state equation (28.6) can be re-modeled as

$$\text{Measurement equation}: \; r_l = c_l s_{kl} + u_l \tag{28.7}$$

and

$$\text{State equation}: \; t_{l+1} = t_l + v_l \tag{28.8}$$

respectively, where the scalar system gain c_l is generally considered as 1. Unfortunately, (28.7) and (28.8) are uncorrelated in the sense that there is no state parameter t_l in the measurement equation. In order to resolve this dilemma, we re-express (28.7) as

$$\text{Measurement equation}: \; r_{l+1} = (s_{kl}/t_l)t_l + u_l \tag{28.9}$$

where the matching spectral signature vector $\mathbf{s}_k = (s_{k1}, s_{k2}, \ldots, s_{kL})^T$ is included in the measurement equation (28.9) and therefore is the signature vector to be used to match the observable signature vector $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$. The auxiliary spectral signature vector **t** in (28.9) then serves as a bridge between the observable signature vector **r** and the matching signature vector $\mathbf{s}_k$ to dictate the ability of a given signature vector $\mathbf{s}_k$ to match the observable signature vector **r**. As a result of introducing the target signature vector **t** in (28.9) the system gain parameter c_l in (28.7) is re-expressed in (28.9) and becomes a spectral-varying parameter specified by s_{kl}/t_l. Interestingly, the use of such a spectral-varying system gain parameter by $c_l = s_{kl}/t_l$ takes care of the effects resulting from a matching signature vector $\mathbf{s}_k$ in (28.9). This is a key difference between KFSSE and KFSSI. Therefore, the KF using the pair of (28.9) and (28.8) to perform spectral signature identification is called Kalman filter-based spectral signature identifier (KFSSI). In this case, the input is provided by the observable spectral signature vector $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ to be identified by a particular signature vector in $\Delta = \{\mathbf{s}_k\}_{k=1}^{K}$ via KFSSI. And the output of KFSSI is the estimate of the auxiliary signature vector **t**, $\hat{\mathbf{t}}^{\text{KFSSI}}(\mathbf{s}) = \left(\hat{t}_1^{\text{KFSSI}}(\mathbf{s}), \hat{t}_2^{\text{KFSSI}}(\mathbf{s}), \ldots, \hat{t}_l^{\text{KFSSI}}(\mathbf{s})\right)^T$. To implement (28.8) and (28.9) recursively, the initial condition used to start the recursive algorithm between measruement and state equations is set to $\hat{t}_1^{\text{KFSSI}}(\mathbf{s}) = 0$ by assuming that the estimate of the auxiliary signature vector **t** at the first band is 0. Then KF is implemented recursively until it reaches the last band. Since KF is an MSE-based estimation technique, the least squares error (LSE) between $\hat{t}_l^{\text{KFSSI}}(\mathbf{s})$ and the observable spectral signature $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ is used instead of MSE as a quantitative measure defined by

$$\text{LSE}(\mathbf{r}, \mathbf{s}_k) = ||\mathbf{r} - \hat{t}^{\text{KFSSI}}(\mathbf{s}_k)|| = \sqrt{\sum\nolimits_{l=1}^{L} \left(r_l - \hat{t}_l^{\text{KFSSI}}(\mathbf{s}_k)\right)^2} \tag{28.10}$$

Using (28.10), we define the observable spectral signature vector $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ as a particular spectral signature vector $\mathbf{s}_k$ in Δ that yields the least value of LSE(**r**, **s**).

It is worth noting that in addition to the auxiliary spectral signature vector **t** introduced in (28.9), another major distinction between KFSSE and KFSSI is that the latter identifies a signature vector via $\Delta = \{\mathbf{s}_k\}_{k=1}^{K}$, whereas the former estimates the true signature vector without the need of Δ. As a result, their measurement equations are implemented differently where the signature vector used in the measurement equation (28.5) for KFSSE is the true signature vector **t**, whereas the signature vector used in the measurement equation (28.7) or (28.9) for KFSSI is a matching signature vector $\mathbf{s}_k$ chosen from the database or spectral library, $\Delta = \{\mathbf{s}_k\}_{k=1}^{K}$. A third major distinction is that the standard deviation of the state noise **v**, $\sigma_{\mathbf{v}}$ specified in (28.8), generally has a significant effect on LSE compared to $\sigma_{\mathbf{v}}$ used by KFSSE, which does not have such effect on the estimation performance of KFSSE. This is because KFSSI performs identification via a matching signature vector in Δ. Consequently, a small variation caused by $\sigma_{\mathbf{v}}$ may result in an incorrect identification. This is particularly true when the spectral signature vectors in the Δ are similar. Additionally, a fourth major distinction is that the two equations (28.8) and (28.9) implemented in KFSSI must process two different signature vectors, observable signature vector **r** and matching signature vector $\mathbf{s}_k$, compared with KFSSE, which only has the observable signature vector **r** processed in (28.5) and (28.6). Accordingly, KFSSI is sensitive to the measurement noise $\sigma_{\mathbf{u}}$ and the state noise $\sigma_{\mathbf{v}}$, both of which are correlated. However, this is not the case for KFSSE. So, in KFSSI, both measurement noise and state noise must be appropriately addressed as will be demonstrated by the experiments in the following sections.

The relationship between KFSSE and KFSSI, which can be illustrated by the relationship between constrained energy minimization (CEM) and orthogonal subspace projection (OSP) in Chapters 2 and 12, is worth commenting on. CEM assumes that there is a desired target signature vector **d** to be detected. This **d** actually corresponds to the target signature vector **t** assumed in KFSSE to be estimated. It then uses **d** to generate detected abundance frcations of **d** present in all the pixels in an image cube to perform target detection, whereas KFSSE estimates $\hat{t}_l^{\text{KFSSE}}$ to approxmiate the lth band r_l in the observed signature **r**. In other words, CEM performs target detection in an image cube, whereas KFSSE estimates target abundance in a single signature vector. On the other hand, OSP assumes that there are p image endmembers that are used to unmix pixel vectors in an image cube by estimating abundance fractions of each of p image enedmembers present in each of the image pixel vectors. This is exactly what KFSSI does for a single signature vector where it uses a database formed by K signatures $\Delta = \{\mathbf{s}_k\}_{k=1}^{K}$ that correspond to p image endmembers used by OSP for unmixing and then identifies a signature vector from the database by finding a best matching signature vector. This functionality is equivalent to that of OSP, which uses its p estimated abundance fractions for each of image pxiel vectors to perform linear spectral unmixing. The key difference is that CEM and OSP operate on image cubes, whereas both KFSSE and KFSSI operate on single signature vector only.

28.3.3 Kalman Filter-Based Spectral Signature Quantifier

So far, KFSSE and KFSSI developed in the two previous subsections make attempts to perform signature vector estimation and identification from an observable signature vector **r**. This subsection presents another new application of KFSSE in spectral quantification to quantify a signature vector **r**, which is referred to as KFSSQ.

One of great challenges in hyperspectral data exploitation is spectral quantification. This is particularly important for CB defense where the concentrations of targets are of major interest rather than their shapes and sizes generally encountered in image processing. The concentration of an agent is a key element in the assessment of threat (Kwan et al., 2006). Over the past years, many algorithms have been developed for quantification (Goetz and Boardman, 1989; Shimabukuro and Smith, 1991; Settle and Drake, 1993; Smith et al., 1994; Tompkins et al., 1997; Heinz and Chang, 2001; Kwan et al., 2006). However, most of them are image analysis-based linear spectral unmixing methods, which estimate abundance fractions of image endmembers assumed to be present in a linear mixture model such as fully constrained least squares (FCLS) method (Heinz and Chang, 2001). The technique developed in this subsection is based on signature vector and allows quantification of a particular material substance present in a single signature vector of interest band by band without appealing for a linear mixing model. Once again, a direct application of KFSSE in spectral quantification is not applicable, since the state equation described by (28.6) is the spectral value t_l but not its abundance fraction α_l, which is of interest in quantification. In order to estimate the abundance fraction α_l in the state equation, the state t_l in (28.6) is replaced with its abundance fraction α_l and the system gain c_l in (28.6) is replaced with the lth band of the target signature vector **t**, t_l. The state transition parameter $\phi(l+1, l)$ is set to 1 such that the state equation is actually a zero-holder interpolator. The resulting measurement equation and state equation become

$$\text{Measurement equation: } r_l = \alpha_l t_l + u_l \tag{28.11}$$

$$\text{State equation: } \alpha_{l+1} = \alpha_l \tag{28.12}$$

By virtue of the pair of (28.11) and (28.12), a KFSSQ can be designed and implemented to quantify the target signature $\mathbf{t} = (t_1, t_2, \ldots, t_L)^T$ present in the data sample vector **r** in terms of its abundance fraction α_l. In this case, the inputs of KFSSQ are observable signature vector $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ and target signature vector $\mathbf{t} = (t_1, t_2, \ldots, t_L)^T$. Its output is the estimate of α_l associated with target signature vector **t**, denoted by $\hat{\alpha}_l^{\text{KFSSQ}}(\mathbf{t})$.

There are two scenarios that can be implemented for KFSSQ. One is that the target signature vector **t** in (28.11) is assumed to be known as *a priori*. However, in many applications, we may not have such prior knowledge. Under this circumstance, another scenario is that **t** can be estimated from the observable signature vector $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$via KFSSE where the t_l in (28.11) is then replaced by its KFSSE estimation, $\hat{t}_l^{\text{KFSSE}}$. The resulting measurement equation becomes

$$\text{Measurement equation}: \; r_l = \alpha_l \hat{t}_l^{\text{KFSSE}} + u_l \tag{28.13}$$

The pair of (28.12) and (28.13) specifies the implementation of KFSSQ for unknown target signature vector **t** where the input is KFSSE-estimated $\hat{t}_l^{\text{KFSSE}}$specified by (28.13) and the output is the estimate of α_l in (28.12) corresponding to $\hat{t}_l^{\text{KFSSE}}$, denoted by $\hat{\alpha}_l^{\text{KFSSQ}}(\hat{\mathbf{t}}^{\text{KFSSE}})$.

KFSSQ described by (28.12) and (28.13) is quite different from KFLU, which assumes all target signature vectors, that is, image endmembers are known *a priori* and then quantifies all the known signatures present in the data vector **r** via a linear mixing model by FCLS. However, the second scenario of KFSSQ specified by (28.12) and (28.13) can be implemented without prior knowledge of the target signature vector **t**, a task that cannot be accomplished by KFLU.

In the following, the algorithmic steps of each of the three KFSCSP techniques, KFSSE, KFSSI, and KFSSQ, are described. We assume that $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ and $\mathbf{t} = (t_1, t_2, \ldots, t_L)^T$ are the observable signature vector and the target signature vector to be estimated, respectively.

KFSSE

1. Initial conditions:
   - Preset the values of $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{v}}$.
   - Set $c_l = \phi_{l+1|l} = 1$for all $l = 1, 2, \ldots L$.
   - let $\hat{t}_{1|}0 = 0$ and $P_{1|0} = Var[t_1] = 1$.
2. Start r_l with $l = 1$:
   - Calculate Kalman gain

     $$K_l = [\phi_{l+1|l} \cdot P_{l|l-1} \cdot c_l]\left[c_l \cdot P_{l|l-1} \cdot c_l + \sigma_{\mathbf{u}}\right]^{-1}.$$

   - Update the state at $l = 1$ by

     $$\hat{t}_{l|l} = \hat{t}_{l|l-1} + K_l\left[r_l - c_l \cdot \hat{t}_{l|l-1}\right].$$

   - Update variance of the state at $l = 1$ by

     $$P_{l|l} = [1 - K_l \cdot c_l] \cdot P_{l|l-1}.$$

   - Predict the state at $l + 1$ by

     $$\hat{t}_{l+1|l} = \phi_{l+1|l} \cdot \hat{t}_{l|l}.$$

   - Predict variance of the state at $l + 1$ by

     $$P_{l+1|l} = \phi_{l+1|l} \cdot P_{l|l} \cdot \phi_{l+1|l} + \sigma_{\mathbf{v}}.$$

3. Increase l by 1 and proceed r_l with $l = 2$, repeat the five recursive steps outlined in step 2 until $l = L$.
4. Output KFSSE-estimated $\hat{\mathbf{t}}^{\text{KFSSE}} = \left(\hat{t}_1^{\text{KFSSE}}, \hat{t}_2^{\text{KFSSE}}, \ldots, \hat{t}_L^{\text{KFSSE}}\right)^T$ with $\hat{t}_l^{\text{KFSSE}} = \hat{t}_l$ for all $1 \leq l \leq L$ obtained from steps 2 and 3.

KFSSI

1. Initial conditions:
   - Assume that $\Delta = \{\mathbf{s}_k\}_{k=1}^K$ is a given spectral library or database and $\mathbf{s}_k$ is a matching signatures selocted from Δ.
   - Pre-set the values of $\sigma_\mathbf{u}$ and $\sigma_\mathbf{v}$.
   - Set $\hat{t}_{1|0}(\mathbf{s}_k) = 0$ and $P_{1|0} = Var[t_1] = 1$.
   - Set $c_l = s_{kl}/\hat{t}_l$ for all $l = 1, 2, \ldots, L$.
   - Set $\phi_{l+1|l} = 1$ for all $l = 1, 2, \ldots, L$.
2. Start r_l with $l = 1$:
   - Calculate Kalman gain

     $$K_l = \left[\phi_{l+1|l} \cdot P_{l|l-1} \cdot c_l\right]\left[c_l \cdot P_{l|l-1} \cdot c_l + \sigma_\mathbf{u}\right]^{-1}.$$

   - Update the state at $l = 1$ by

     $$\hat{t}_{l|l} = \hat{t}_{l|l-1} + K_l\left[r_l - c_l \cdot \hat{t}_{l|l-1}\right].$$

   - Update variance of the state at $l = 1$ by

     $$P_{l|l} = [1 - K_l \cdot c_l] \cdot P_{l|l-1}.$$

   - Predict the state at $l + 1$by

     $$\hat{t}_{l+1|l} = \phi_{l+1|l} \cdot \hat{t}_{l|l}.$$

   - Predict variance of the state at $l + 1$ by

     $$P_{l+1|l} = \phi_{l+1|l} \cdot P_{l|l} \cdot \phi_{l+1|l} + \sigma_\mathbf{v}.$$

3. Increase l by 1 and proceed r_l with $l = 2$. Update $c_l = s_{kl}/\hat{t}_l$ and repeat the five recursive steps outlined in step 2 until $l = L$.
4. Output KFSSI-estimated $\hat{\mathbf{t}}^{\text{KFSSI}}(\mathbf{s}_k) = \left(\hat{t}_1^{\text{KFSSI}}(\mathbf{s}_k), \hat{t}_2^{\text{KFSSI}}(\mathbf{s}_k), \ldots, \hat{t}_L^{\text{KFSSI}}(\mathbf{s}_k)\right)^T$ with $\hat{t}_l^{\text{KFSSE}}(\mathbf{s}_k) = \hat{t}_l$ for all $1 \leq l \leq L$ obtained from steps 2 and 3.
5. Find the $\mathbf{s}^*$ which yields the minimum LSE between $\hat{\mathbf{t}}^{\text{KFSSI}}(\mathbf{s}_k)$ and $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ over $\mathbf{s}_k$ in Δ via (28.10), that is, $\mathbf{s}^* = \arg\left\{\min_{\mathbf{s}_k \in \Delta} \text{LSE}(\mathbf{r}, \mathbf{s}_k)\right\}$.
6. Identify $\mathbf{r} = (r_1, r_2, \ldots, r_L)^T$ as the signature $\mathbf{s}^*$.

KFSSQ

1. Initial conditions:
   - Preset the values of $\sigma_\mathbf{u}$.
   - Set $\sigma_\mathbf{v} = 0$.
   - Assume that $\boldsymbol{\alpha}(\mathbf{t})$ is the abundance vector corresponding to the target signature t. Set $c_l = t_l$ for $l = 1, 2, \ldots, L$.
   - Let $\hat{\boldsymbol{\alpha}}_{1|0}(\mathbf{t}) = 0$ and$P_{1|0} = Var[\alpha_1] = 1$.
   - Set $\phi_{l+1|l} = 1$ for all $l = 1, 2, \ldots, L$.

2. Start r_l with $l = 1$:
   - Calculate Kalman gain

     $$K_l = [\phi_{l+1|l} \cdot P_{l|l-1} \cdot c_l]\left[c_l \cdot P_{l|l-1} \cdot c_l + \sigma_{\mathbf{u}}\right]^{-1}.$$

   - Update the state at $l = 1$ by

     $$\hat{\alpha}_{l|l}(\mathbf{t}) = \hat{\alpha}_{l|l-1}(\mathbf{t}) + K_l\left[r_l - c_l \cdot \hat{\alpha}_{l|l-1}(\mathbf{t})\right].$$

   - Update variance of the state at $l = 1$ by

     $$P_{l|l} = [1 - K_l \cdot c_l] \cdot P_{l|l-1}.$$

   - Predict the state at $l + 1$ by

     $$\hat{\alpha}_{l+1|l} = \phi_{l+1|l} \cdot \hat{\alpha}_{l|l}.$$

   - Predict variance of the state at $l + 1$ by

     $$P_{l+1|l} = \phi_{l+1|l} \cdot P_{l|l} \cdot \phi_{l+1|l} + \sigma_{\mathbf{v}}.$$

3. Increase l by 1 and proceed r_l with $l = 2$, update $c_l = t_l$ and repeat the five recursive steps outlined in step 2 until $l = L$.
4. Output KFSSQ-estimated abundance vector $\hat{\boldsymbol{\alpha}}^{\mathrm{KFSSQ}}(\mathbf{t}) = \left(\hat{\alpha}_1^{\mathrm{KFSSQ}}(\mathbf{t}), \hat{\alpha}_2^{\mathrm{KFSSQ}}(\mathbf{t}), \ldots, \hat{\alpha}_L^{\mathrm{KFSSQ}}(\mathbf{t}\right)^T$ with $\hat{\alpha}_l^{\mathrm{KFSSQ}}(\mathbf{t}) = \hat{\alpha}_l$ for all $1 \leq l \leq L$ obtained from steps 2 and 3.

28.4 Computer Simulations Using AVIRIS Data

In order to demonstrate the utility of KFSCSP techniques in spectral estimation, identification, and quantification, computer simulations and real data experiments were conducted for performance evaluation and analysis. For computer simulations, five Airborne Visible InfraRed Imaging Spectrometer (AVIRIS) reflectance data in Figure 1.8 are reproduced in Figure 28.1 for reference. There are five signature vectors, blackbrush, creosote leaves, dry grass, redsoil, and sagebrush, to be used for experiments where these spectra have 158 bands after water bands are removed.

According to Figure 28.1, the signatures of blackbrush, creosote leaves, and sagebrush are close to each other in terms of spectral shape. In particular, the spectral signatures of creosote leaves and sagebrush are very similar. A detailed quantitative analysis of these three signatures can be found in Chang (2003a).

Since KFSCSP techniques are signature vector-based techniques not to be used for image pixel vectors, KFSSE, KFSSI, and KFSSQ are not designed for classification. Therefore, their performance is evaluated by signature vector-based spectral measures such as spectral angle mapper (SAM), spectral information divergence (SID) rather than image classifiers.

28.4.1 KFSSE

To implement KFSSE, the system gain c_l in (28.5) was set to be 1 for all $1 \leq l \leq L$, the standard deviation of the state noise $\mathbf{v}$, $\sigma_{\mathbf{v}}$ was empirically set to 10^3 and the standard deviation of the measurement noise $\mathbf{u}$, $\sigma_{\mathbf{u}}$ was chosen to achieve signal-to-noise ratio (SNR) = 30 dB, where the SNR was defined as the ratio of half a signature reflectance mean to the standard deviation of noise (Harsanyi and Chang, 1994). It should be noted that throughout our extensive experiments, the results demonstrated that KFSSE was robust to the selection of $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{v}}$. More specifically, once the value of $\sigma_{\mathbf{v}}$ was greater than 10^3, the $\sigma_{\mathbf{u}}$ had limited impact on the performance of KFSSE.

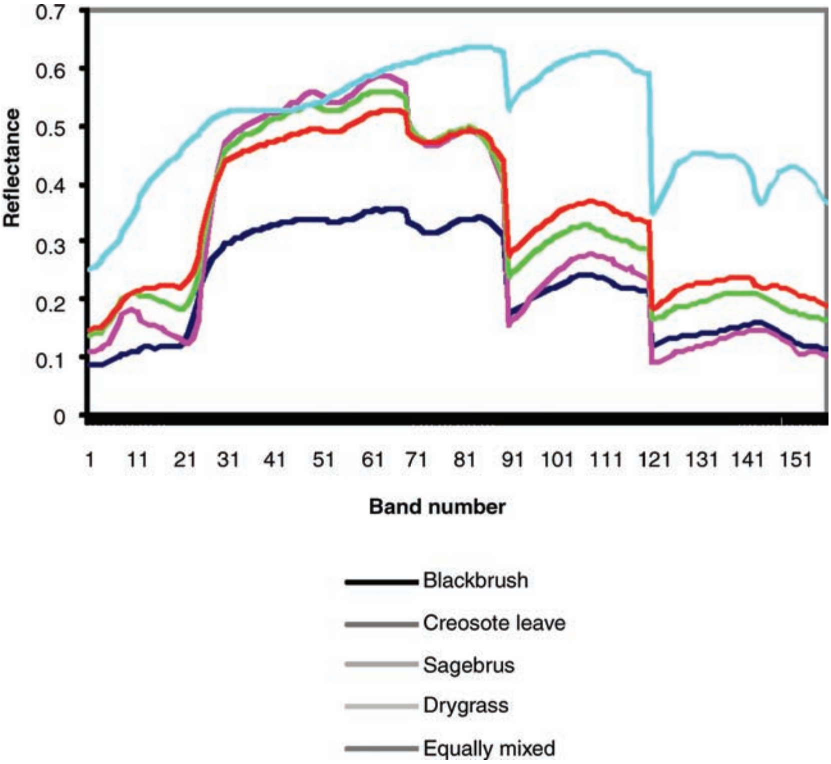


Figure 28.1 Reflectances of creosote leaves, blackbrush, sagebrush, drygrass, and redsoil.

Therefore, for KFSSE implemented in this chapter, the values of $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{v}}$ were fixed at 30 dB and 10^3, respectively.

Figure 28.2(a)–(e) shows the five KFSSE-estimated reflectance spectra of the signature vectors given in Figure 28.1. Since the values of the corresponding estimation errors obtained by KFSSE were very small, Figure 28.3(a)–(e) shows the ratios of estimation errors to their corresponding reflectance spectra in Figure 28.1 by the logarithm function to have better visual assessment.

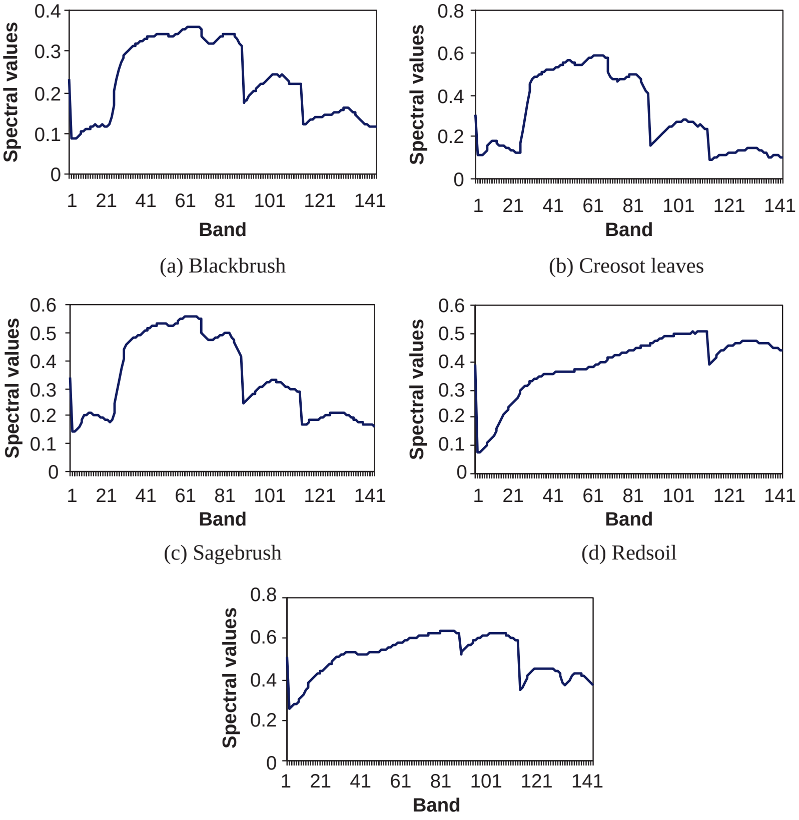
According to Figure 28.3, KFSSE was very effective in estimating spectral signatures with estimation errors nearly close to zero. Obviously, large estimation errors always occurred at wavelengths where their spectral values had changed drastically. To the contrary, if there was a smooth transition between two wavelengths, the estimation errors were relatively small.

28.4.2 KFSSI

Two scenarios were implemented by KFSSI: one for subpixel target identification and the other for mixed target identification with the target signature vector **t** to be identified and assumed to be either known or unknown.

28.4.2.1 Subpixel Target Identification by KFSSI

First of all, we simulated a subpixel target implanted in a signature vector. Without loss of generality, we assumed that the subpixel target signature was the creosote leaves (C) and the background signature was the redsoil (R). Three target pixel vectors, $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$ were generated in accordance



(a) Blackbrush

(b) Creosot leaves

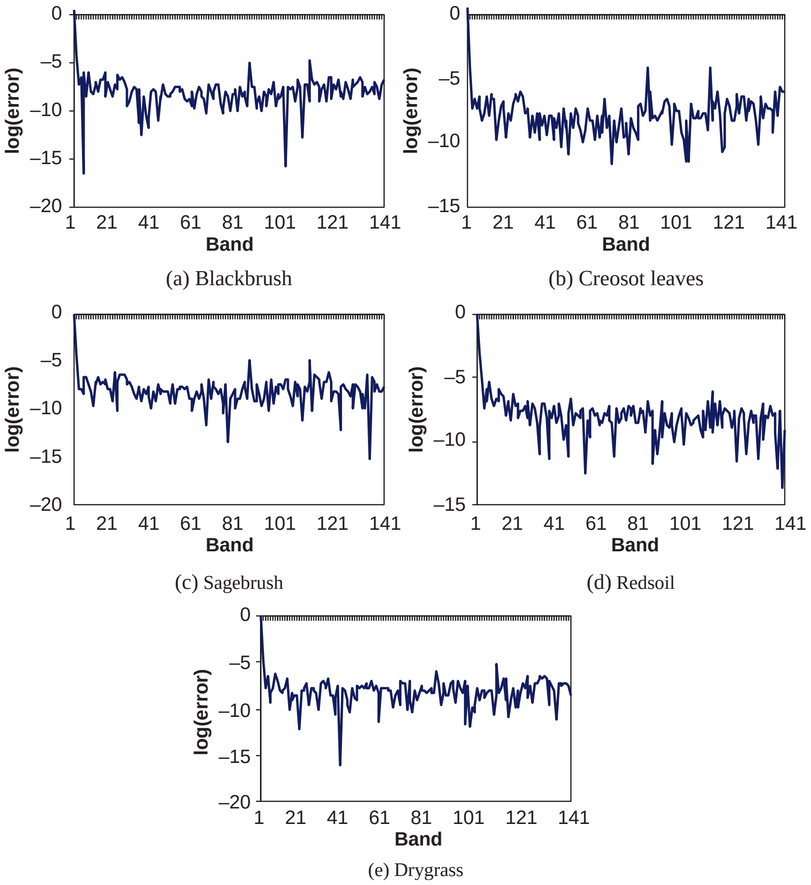
(c) Sagebrush

(d) Redsoil

(e) Drygrass

Figure 28.2 KFSSE-estimated spectra of the five reflectance signatures in Figure 28.1.

with the abundance fractions of the subpixel target signature, creosote leaves set to 75%, 50%, and 25% mixed with the redsoil as the background signature to make up 100% abundance, respectively, that is, $\mathbf{t}_1 = \frac{3}{4}C + \frac{1}{4}R$, $\mathbf{t}_2 = \frac{1}{2}C + \frac{1}{2}R$, and $\mathbf{t}_3 = \frac{1}{4}C + \frac{3}{4}R$. KFSSI was then implemented to identify C from the three target signature vectors, $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$ via a database $\Delta = \{$creosote leaves and redsoil$\}$. The identification was carried out by first setting the values for both $\sigma_\mathbf{u}$ and $\sigma_\mathbf{v}$, then randomly choosing a matching signature vector from the database Δ to match the unknown subpixel target panel according to (28.8) and (28.9), and finally identifying the unknown subpixel target panel as the one that yielded the minimum LSE. The resulting LSE corresponding to different sizes of subpixel target panels were tabulated in Table 28.1 with $\sigma_\mathrm{v} = \sqrt{1000}$ and various values of the standard deviation of noise $\mathbf{u}$, $\sigma_\mathbf{u}$. However, it should be noted that the selection of $\sigma_\mathbf{u}$ was empirically chosen and dependent upon the target signature vector to be used for experiments. The simulated data conducted in this section were designed to demonstrate and illustrate the utility and effectiveness of KFSSI in signature identification under the impact of the parameter $\sigma_\mathbf{u}$. In doing so, we had experimentally adjusted the value of $\sigma_\mathbf{u}$ in accordance with our simulation data to dictate the impact of subpixel target size on the performance. Figure 28.4(a) and (b) plot the LSEs of creosote leaves and redsoil, respectively, versus the values of $\sigma_\mathbf{u}$ for three sizes of subpixel targets, $\frac{3}{4}$, $\frac{1}{2}$, and $\frac{1}{4}$, where the values of $\sigma_\mathbf{u}$ varying from 10 to 1000 with step size set to 10.



(a) Blackbrush

(b) Creosot leaves

(c) Sagebrush

(d) Redsoil

(e) Drygrass

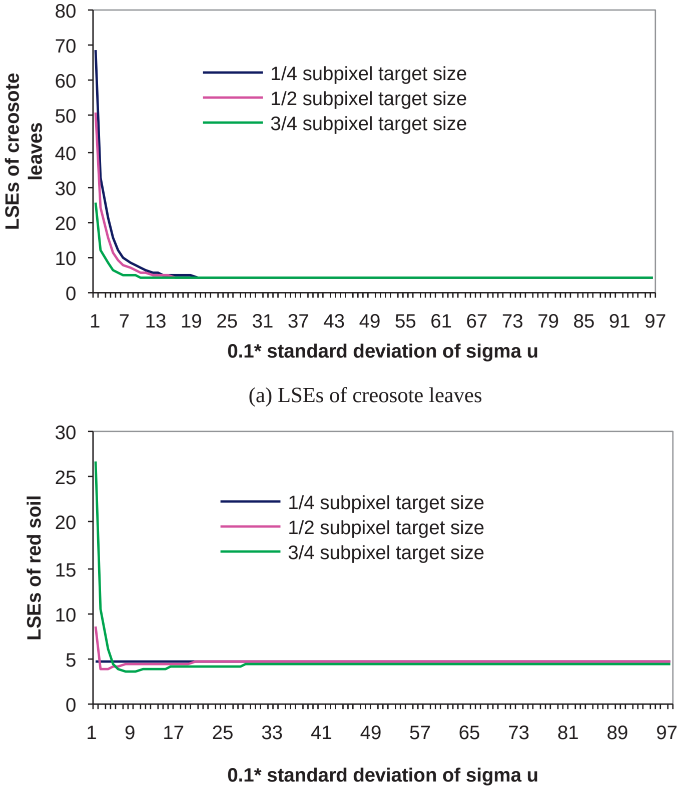
Figure 28.3 Ratios of the estimation errors to the five reflectance spectra in Figure 28.1 in the logarithm functions.

For example, in order to correctly identify the subpixel panel with ¼ size of a pixel, according to Table 28.1, the $\sigma_{\mathbf{u}}$ must be greater than 159 compared to the subpixel panel with ¾ size of a pixel which only requires $\sigma_{\mathbf{u}}$ smaller than 12. Additionally, LSEs resulting from KFSSI was in proportion to the values of $\sigma_{\mathbf{u}}$.

As pointed out, the selection of $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{v}}$ had a significant impact on the performance of KFSSI and must be chosen appropriately in order to achieve acceptable LSEs. According to our exepriments, once $\sigma_{\mathbf{v}}$ was fixed, the minimum value of $\sigma_{\mathbf{u}}$ could be set approximately in the same order of magnitude in order to correctly identify the subpixel target as creosote leaves. Table 28.2 is

Table 28.1 LSE corresponding to different sizes of subpixel target panels (creosote leaves) according to (28.10) via $\Delta = \{$creosote leaves and redsoil$\}$

Subpixel target panels (creosote leaves)	¼ pixel ($\sigma_{\mathbf{u}} \geq 159$)	½ pixel ($\sigma_{\mathbf{u}} \geq 155$)	¾ pixel ($\sigma_{\mathbf{u}} \leq 12$)
Creosote leaves	4.4214	4.3616	4.4069
Redsoil	4.7903	4.5715	4.4273



(a) LSEs of creosote leaves

(b) LSEs of Reds oil

Figure 28.4 Relationship between $\sigma_{\mathbf{u}}$ and LSE according to different sizes of the subpixel target.

included here to illustrate the relationship between $\sigma_{\mathbf{v}}$ and $\sigma_{\mathbf{u}}$ with three different values of $\sigma_{\mathbf{v}}$ and seven subpixel targets of size specified by (1/8, 2/8, 3/8, 4/8, 5/8, 6/8, 7/8).

In order to provide further insights into KFSSI, similar experiments were also conducted by replacing creosote leaves with blackbrush, sagebrush, and drygrass, with their results tabulated in Tables 28.3–28.5, respectively.

According to Tables 28.2–28.5, two factors affected the selection of $\sigma_{\mathbf{u}}$. One is target size and the other is the standard deviation of the state noise $\sigma_{\mathbf{v}}$. When similar experiments were conducted by replacing the creosote leaves with blackbrush, sagebrush, and drygrass, the changes in $\sigma_{\mathbf{u}}$ were

Table 28.2 Relationship between $\sigma_{\mathbf{v}}$ and $\sigma_{\mathbf{u}}$ with creosote leaves as target signature embedded into redsoil

$\sigma_{\mathbf{v}}$	$\sqrt{10}$	$\sqrt{100}$	$\sqrt{1000}$
1/8 subpixel size	$\sigma_{\mathbf{u}} \geq 18$	$\sigma_{\mathbf{u}} \geq 53$	$\sigma_{\mathbf{u}} \geq 158$
2/8 subpixel size	$\sigma_{\mathbf{u}} \geq 19$	$\sigma_{\mathbf{u}} \geq 53$	$\sigma_{\mathbf{u}} \geq 159$
3/8 subpixel size	$\sigma_{\mathbf{u}} \geq 22$	$\sigma_{\mathbf{u}} \geq 55$	$\sigma_{\mathbf{u}} \geq 157$
4/8 subpixel size	$\sigma_{\mathbf{u}} \geq 30$	$\sigma_{\mathbf{u}} \geq 63$	$\sigma_{\mathbf{u}} \geq 155$
5/8 subpixel size	$\sigma_{\mathbf{u}} \geq 123$	$\sigma_{\mathbf{u}} \geq 108$	$\sigma_{\mathbf{u}} \geq 177$
6/8 subpixel size	$\sigma_{\mathbf{u}} \leq 5$	$\sigma_{\mathbf{u}} \leq 11$	$\sigma_{\mathbf{u}} \leq 12$
7/8 subpixel size	$\sigma_{\mathbf{u}} \leq 9$	$\sigma_{\mathbf{u}} \leq 22$	$\sigma_{\mathbf{u}} \leq 52$

Table 28.3 Relationship between $\sigma_{\mathbf{v}}$ and $\sigma_{\mathbf{u}}$ with sagebrush as target signature embedded into redsoil

$\sigma_{\mathbf{v}}$	$\sqrt{10}$	$\sqrt{100}$	$\sqrt{1000}$
1/8 subpixel size	$\sigma_{\mathbf{u}} \geq 12$	$\sigma_{\mathbf{u}} \geq 33$	$\sigma_{\mathbf{u}} \geq 92$
2/8 subpixel size	$\sigma_{\mathbf{u}} \geq 16$	$\sigma_{\mathbf{u}} \geq 34$	$\sigma_{\mathbf{u}} \geq 94$
3/8 subpixel size	$\sigma_{\mathbf{u}} \geq 45$	$\sigma_{\mathbf{u}} \geq 52$	$\sigma_{\mathbf{u}} \geq 96$
4/8 subpixel size	$\sigma_{\mathbf{u}} \leq 4$	$\sigma_{\mathbf{u}} \leq 8$	$\sigma_{\mathbf{u}} \leq 7$
5/8 subpixel size	$\sigma_{\mathbf{u}} \leq 8$	$\sigma_{\mathbf{u}} \leq 21$	$\sigma_{\mathbf{u}} \leq 47$
6/8 subpixel size	$\sigma_{\mathbf{u}} \leq 10$	$\sigma_{\mathbf{u}} \leq 28$	$\sigma_{\mathbf{u}} \leq 70$
7/8 subpixel size	$\sigma_{\mathbf{u}} \leq 12$	$\sigma_{\mathbf{u}} \leq 34$	$\sigma_{\mathbf{u}} \leq 88$

Table 28.4 Relationship between $\sigma_{\mathbf{v}}$ and $\sigma_{\mathbf{u}}$ with blackbrush as target signature embedded into redsoil

$\sigma_{\mathbf{v}}$	$\sqrt{10}$	$\sqrt{100}$	$\sqrt{1000}$
1/8 subpixel size	$\sigma_{\mathbf{u}} \geq 32$	$\sigma_{\mathbf{u}} \geq 97$	$\sigma_{\mathbf{u}} \geq 298$
2/8 subpixel size	$\sigma_{\mathbf{u}} \geq 30$	$\sigma_{\mathbf{u}} \geq 93$	$\sigma_{\mathbf{u}} \geq 284$
3/8 subpixel size	$\sigma_{\mathbf{u}} \geq 29$	$\sigma_{\mathbf{u}} \geq 87$	$\sigma_{\mathbf{u}} \geq 265$
4/8 subpixel size	$\sigma_{\mathbf{u}} \geq 26$	$\sigma_{\mathbf{u}} \geq 80$	$\sigma_{\mathbf{u}} \geq 239$
5/8 subpixel size	$\sigma_{\mathbf{u}} \geq 23$	$\sigma_{\mathbf{u}} \geq 69$	$\sigma_{\mathbf{u}} \geq 203$
6/8 subpixel size	$\sigma_{\mathbf{u}} \geq 18$	$\sigma_{\mathbf{u}} \geq 52$	$\sigma_{\mathbf{u}} \geq 150$
7/8 subpixel size	$\sigma_{\mathbf{u}} \geq 10$	$\sigma_{\mathbf{u}} \geq 26$	$\sigma_{\mathbf{u}} \geq 66$

noticeable as shown in Table 28.2–28.5. Three interesting findings from Tables 28.2–28.5 are intriguing.

Finding 1: When target size was fixed, the value of $\sigma_{\mathbf{u}}$ was in proportional to the value of $\sigma_{\mathbf{v}}$.

Finding 2: The range of $\sigma_{\mathbf{u}}$ was closely related to the subpixel target signature to be identified. For example, according to the study by Chang (2000, 2003a), the creosote leaves was the most difficult signature to be discriminated, since it was very similar to both sagebrush and blackbrush. Table 28.2 demonstrates this fact by providing this evidence that the range of $\sigma_{\mathbf{u}}$ must be bounded below from 18 to 123 as the subpixel target size from 1/8 to 5/8 with $\sigma_{\mathbf{v}}$ fixed at $\sqrt{10}$. Interestingly, once the size was greater than 5/8, the range of $\sigma_{\mathbf{u}}$ was suddenly reversed from bounded below to bounded above. The situation was slightly improved when the subtarget signature vector was sagebrush in Table 28.3, where a sudden change in the range of $\sigma_{\mathbf{u}}$ occurred at size ½. This phonemonon was further evidenced by the blackbrush in Table 28.4, where the range of $\sigma_{\mathbf{u}}$ was all bounded

Table 28.5 Relationship between $\sigma_{\mathbf{v}}$ and $\sigma_{\mathbf{u}}$ with drygrass as target signature embedded into redsoil

$\sigma_{\mathbf{v}}$	$\sqrt{10}$	$\sqrt{100}$	$\sqrt{1000}$
1/8 subpixel size	$\sigma_{\mathbf{u}} \leq 4$	$\sigma_{\mathbf{u}} \leq 8$	$\sigma_{\mathbf{u}} \leq 13$
2/8 subpixel size	$\sigma_{\mathbf{u}} \leq 10$	$\sigma_{\mathbf{u}} \leq 29$	$\sigma_{\mathbf{u}} \leq 74$
3/8 subpixel size	$\sigma_{\mathbf{u}} \leq 16$	$\sigma_{\mathbf{u}} \leq 48$	$\sigma_{\mathbf{u}} \leq 138$
4/8 subpixel size	$\sigma_{\mathbf{u}} \leq 22$	$\sigma_{\mathbf{u}} \leq 66$	$\sigma_{\mathbf{u}} \leq 195$
5/8 subpixel size	$\sigma_{\mathbf{u}} \leq 27$	$\sigma_{\mathbf{u}} \leq 82$	$\sigma_{\mathbf{u}} \leq 248$
6/8 subpixel size	$\sigma_{\mathbf{u}} \leq 31$	$\sigma_{\mathbf{u}} \leq 96$	$\sigma_{\mathbf{u}} \leq 294$
7/8 subpixel size	$\sigma_{\mathbf{u}} \leq 35$	$\sigma_{\mathbf{u}} \leq 109$	$\sigma_{\mathbf{u}} \leq 337$

below from 10 to 32 that were inversely proprtional to target size with $\sigma_{\mathbf{v}}$ also fixed at $\sqrt{10}$. Finally, since the drygrass was very dissimilar to all the three signature vectors, creosote leaves, sagebrush, and blackbrush, Table 28.5 reflects the fact that the range of $\sigma_{\mathbf{u}}$ was all bounded above from 4 to 35 that were proprtional to target size when with the $\sigma_{\mathbf{v}}$ was fixed at $\sqrt{10}$, a case that was completely opposite to blackbrush in Table 28.4.

Finding 3: The selection of initial values for $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{v}}$ was very much dependent on the data to be processed, such as spectral similarity among signature vectors. This must be done on an empirical basis. For example, if signature vectors to be considered were spectrally distinct, the results would be very robust to the selection. On the other hand, if the signature vectors were spectrally similar, the results would be sensitive to how the initial values were selected. In the above experiments, they were determined empirically based on *a priori* knowledge about the material substance signatures and subpixel size. Nevertheless, from our extensive experiments a general guideline to determine the initial values of $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{v}}$ may be useful. Without loss of generality, we can assume a mixing signature vector specified by $\mathbf{r} = \mathbf{s} + \mathbf{b}$, where $\mathbf{s}$ and $\mathbf{b}$ can be interpreted as the embedded signature and background, respectively. In this case, the size of subpixel target also has a significant effect on the subpixel target by the background signature. The initial guess for the value of $\sigma_{\mathbf{u}}$ to properly identify the signature vector $\mathbf{s}$ embedded into the signature vector $\mathbf{r}$ is closely related to $\sigma_{\mathbf{v}}$ in a form of $\sigma_{\mathbf{u}} = 10 \cdot \sigma_{\mathbf{v}} \cdot \left[\frac{\sigma_{(\mathbf{r}-\mathbf{t})}}{\max(\sigma_{\mathbf{t}},\sigma_{\mathbf{b}})}\right]$, where the target signature vector $\mathbf{t}$ is any auxiliary signature vector related to $\mathbf{s}$ as specified by (28.8) and (28.9). In regard to the value of $\sigma_{\mathbf{v}}$, it can be empirically set to $\sqrt{10}$, $\sqrt{100}$, $\sqrt{1000}$, etc. However, we would like to point out that this guideline only serves as a reference and should not take as a criterion for all the cases.

The above findings offer a new look of KFSSI in to how to use signature characterization to perform subpixel target identification. The standard deviation of the measurement noise, $\sigma_{\mathbf{u}}$, used in KFSSI is a very important parameter for identification and varies with the subpixel target signature to be identified, which makes sense. These interesting findings cannot be observed by any spectral measure as demonstrated in Tables 28.6–28.9, which show the results of the same experiments using SAM and SID, where the SID values are given in parentheses. As shown in these tables, it was impossible for SAM and SID to detect the subpixel target panel if its size was smaller than ½ size of a panel. To the contrary, KFSSI could detect subpixel targets correctly even its size was smaller than ½ size of a pixel as long as $\sigma_{\mathbf{u}}$ was chosen to be the values tabulated in Tables 28.2–28.5.

The above experiments demonstrate an important advantage of KFSSI which is that KFSSI could perform well in the subpixel identification, even if the size of a subpixel target was less than ½ of ground sampling distance, that is, pixel resolution, which could not be achieved by any other spectral measure.

Table 28.6 Identification of subpixel panels with creosote leaves as a target signature vector embedded into redsoil by SAM and SID

Subpixel size	Creosote leaves	Redsoil
1/8	0.5104 (0.4891)	0.0610 (0.0056)
1/4	0.4456 (0.3853)	0.1258 (0.0235)
3/8	0.3771 (0.2888)	0.1942 (0.0559)
1/2	0.3055 (0.2012)	0.2658 (0.1057)
5/8	0.2312 (0.1245)	0.3401 (0.1769)
3/4	0.1549 (0.0616)	0.4164 (0.2756)
7/8	0.0775 (0.0175)	0.4938 (0.4109)

Table 28.7 Identification of subpixel panels with sagebrush as a target signature vector embedded into redsoil by SAM and SID

Subpixel size	Sagebrush	Redsoil
1/8	0.4010 (0.2685)	0.0504 (0.0039)
1/4	0.3484 (0.2056)	0.1030 (0.0162)
3/8	0.2938 (0.1494)	0.1576 (0.0378)
1/2	0.2374 (0.1005)	0.2140 (0.0697)
5/8	0.1794 (0.0597)	0.2720 (0.1133)
3/4	0.1202 (0.0282)	0.3312 (0.1706)
7/8	0.0603 (0.0075)	0.3911 (0.2442)

Table 28.8 Identification of subpixel panels with blackbrush as a target signature vector embedded into redsoil by SAM and SID

Subpixel size	Blackbrush	Redsoil
1/8	0.5398 (0.2292)	0.0319 (0.0015)
1/4	0.5044 (0.1892)	0.0677 (0.0069)
3/8	0.4647 (0.1489)	0.1079 (0.0174)
1/2	0.4199 (0.1091)	0.1533 (0.0352)
5/8	0.3693 (0.0711)	0.2048 (0.0631)
3/4	0.3121 (0.0372)	0.2633 (0.1056)
7/8	0.2478 (0.0111)	0.3300 (0.1699)

Table 28.9 Identification of subpixel panels with drygrass as target signature vector embedded into redsoil by SAM and SID

Subpixel size	Drygrass	Redsoil
1/8	0.1842 (0.0645)	0.0336 (0.0025)
1/4	0.1526 (0.0437)	0.0652 (0.0089)
3/8	0.1230 (0.0281)	0.0948 (0.0185)
1/2	0.0952 (0.0168)	0.1226 (0.0304)
5/8	0.0691 (0.0088)	0.1487 (0.0440)
3/4	0.0446 (0.0037)	0.1732 (0.0591)
7/8	0.0216 (0.0009)	0.1962 (0.0753)

28.4.2.2 Mixed Target Identification by KFSSI

In order to make our experiments more interesting and appealing, we further simulated a mixed pixel vector $\mathbf{t}^{\text{mix}}$ by equally mixing ¼ blackbrush ($\mathbf{s}^{\text{B}}$), ¼ creosote leaves ($\mathbf{s}^{\text{C}}$), ¼ dry grass ($\mathbf{s}^{\text{D}}$), and ¼ sagebrush ($\mathbf{s}^{\text{S}}$) as follows:

$$\mathbf{t}^{\text{mix}} = 0.25\mathbf{s}^{\text{B}} + 0.25\mathbf{s}^{\text{C}} + 0.25\mathbf{s}^{\text{D}} + 0.25\mathbf{s}^{\text{S}} \tag{28.14}$$

whose spectral signature vector is also shown in Figure 28.1. In this case, no signature vector was preferred to another. KFSSI was used to identify unknown target signature vector $\mathbf{t}$ present in the mixed pixel vector $\mathbf{t}^{\text{mix}}$ using the database $\Delta = \{$drygrass, blackbrush, creosote leaves, sagebrush$\}$.

Table 28.10 LSE between the mixed pixel $\mathbf{t}^{\text{mix}}$ and different matching signature vectors according to (28.10)

	Blackbrush	Creosote leaves	Sagebrush	Drygrass
$\mathbf{t}^{\text{mix}}$	17.4077	4.8313	4.5572	4.6092

According to the composition of $\mathbf{t}^{\text{mix}}$, all the four signature vectors had equal opportunity to be identified as the target signature vector. The resulting LSEs corresponding to four signature vectors used to match the mixed pixel vector $\mathbf{t}^{\text{mix}}$ were shown in Table 28.10.

Interestingly, since both blackbrush and creosote leaves were close to sagebrush by spectral similarity measures, SAM and SID, KFSSI believed that blackbrush and creosote leaves were part of sagebrush with small variations. As a consequence, it identified $\mathbf{t}^{\text{mix}}$ as sagebrush. Table 28.10 demonstrates that KFSSI could successfully identify the mixed pixel $\mathbf{t}^{\text{mix}}$ as sagebrush as long as $\sigma_{\mathbf{u}} \leq 50$ and drygrass otherwise. As noted, the values of $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{v}}$ might be chosen comparably at the same order of magnitude. In this case, $\sigma_{\mathbf{v}}$ was chosen to be 10, which was of the same order as $\sigma_{\mathbf{u}} \leq 50$. If the $\sigma_{\mathbf{v}}$ in (28.8) was set to 10^3, which is similar to the value used in Table 28.1, the LSEs resulting from KFSSI for subpixel target identification were large.

28.4.3 KFSSQ

This section presents experiments to further demonstrate the use of KFSSQ in quantification of subpixel targets and target signature vectors present in mixed pixels. Compared with KFSSI, which is used to identify an unknown target signature vector $\mathbf{t}$, KFSSQ can also be implemented for the target signature vector $\mathbf{t}$, which is either unknown or known *a priori*. It should also be noted that unlike KFSSI, there is no state noise vector $\mathbf{v}$ in the state equation (28.12) implemented by KFSSQ.

28.4.3.1 Subpixel Target Quantification by KFSSQ

Using the same subpixel targets $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$ simulated in Section 28.4.2.1, we implemented KFSSQ to quantify these three creosote leaves-simulated subpixel target panels $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$ of three respective sizes, ¾, ½, and ¼ with two scenarios described in the following examples.

EXAMPLE 28.1

(Target signature vector t is known)

This example assumed that the target signature $\mathbf{t}$ was known to be creosote leaves. In this case, the pair of (28.11) and (28.12) was implemented to quantify three subpixel targets of creosote leaves, $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$, and the results are tabulated in Table 28.5. Since the subpixel target panels only occupied part of a pixel, its size could be interpreted as portion of abundance fraction in terms of percentage. In this case, the three subpixel target panels $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$ with size of ¾, ½, and ¼ could be considered as targets with abundance fractions 75%, 50%, and 25%, respectively, with KFSSQ-estimated abundance fractions given in Table 28.11, where the quantification results were very accurate with appropriate chosen values of $\sigma_{\mathbf{u}}$.

Table 28.11 Abundance fractions estimated by KFSSQ

KFSSQ	Quantification	$\sigma_{\mathbf{u}}$ in KFSSQ
$\mathbf{t}_1 = 25\%$ creosote	0.2495	7.5677
$\mathbf{t}_2 = 50\%$ creosote	0.5014	4.3506
$\mathbf{t}_3 = 75\%$ creosote	0.7597	2.4521

Table 28.12 Abundance fractions estimated by KFSSQ

KFSSQ	Quantification	$\sigma_{\mathbf{u}}$ in KFSSE	$\sigma_{\mathbf{u}}$ in KFSSQ
$\mathbf{t}_2 = 50\%$ creosote	0.5056	30 dB	4.5247
$\mathbf{t}_3 = 75\%$ creosote	0.7440	30 dB	2.6156

EXAMPLE 28.2

(Target signature vector t is unknown)

Unlike Example 28.1, the prior knowledge of target signature vector **t** was not given in this example. In this case, KFSSQ must use (28.5) and (28.6) to first identify the target signature vector **t** that specified the three subpixel targets $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$ where KFSSE was used for this purpose. The estimated $\hat{\mathbf{t}}^{\mathrm{KFSSE}}$ was then used to replace **t** in (28.11) to produce the abundance vector $\hat{\boldsymbol{\alpha}}^{\mathrm{KFSSQ}}$ specified by (28.12) in Table 28.6. As a result, two different values of $\sigma_{\mathbf{u}}$ were used for measurement noise in KFSSQ: one in (28.5) for KFSSE and the other in (28.13) for KFSSQ. Since both were not correlated, they could be determined independently as tabulated in Table 28.12.

Due to the fact that KFSSE could not correctly estimate the target signature vector **t**, provided the abundance of subpixel target vector **t** was below 50%, in which case it was reasonable, the results for the size of subpixel target being ¼ were not included in Table 28.12.

28.4.3.2 Mixed Target Quantification by KFSSQ

In this subsection, we used the same mixed pixel vector, $\mathbf{t}^{\mathrm{mix}}$ simulated in Section 28.4.2.2 for further experiments. KFSSQ was implemented to quantify the four signatures, blackbrush, creosote leaves, drygrass, and sagebrush, each of which shared a 25% abundance fraction in the pixel vector $\mathbf{t}^{\mathrm{mix}}$. Like Section 28.4.2.3.1, two scenarios were considered.

EXAMPLE 28.3

(All the four target signature vectors are known)

This example assumed that all the four signatures were known. The pair of (28.11) and (28.12) was implemented for KFSSQ. Table 28.13 tabulates the quantification results of the four signatures with their appropriately chosen values of $\sigma_{\mathbf{u}}$. However, it should be noted that the sensitivity of $\sigma_{\mathbf{u}}$ was generally determined by various applications. For example, when KFSSI and KFSSQ were implemented, the results would be sensitive to the value chosen for $\sigma_{\mathbf{u}}$ as shown in Tables 28.10 and 28.12. On the other hand, if KFSSE was implemented as an estimator, the results would be relatively robust to the $\sigma_{\mathbf{u}}$.

As we can see from this table, KFSSQ-estimated abundance fractions of all the four signatures were very accurate and close to true values. In order to make further comparison, the results produced by the FCLS method were also included and tabulated in Table 28.14, where their results are comparable to those obtained in Table 28.13.

Table 28.13 Abundance fractions of the four signatures embedded into the mixed pixel $\mathbf{t}^{\mathrm{mix}}$ estimated by KFSSQ

KFSSQ	Quantification results	$\sigma_{\mathbf{u}}$
25% Sagebrush	0.2504	7.9506
25% Blackbrush	0.2491	6.9136
25% Creosote leaves	0.2499	7.7778
25% D grass	0.2508	8.6420

Table 28.14 Abundance fractions of the four target signature vectors embedded into $\mathbf{t}^{\text{mix}}$ estimated by FCLS

FCLS	Quantification results
25% Sagebrush	0.25135
25% Blackbrush	0.24935
25% Creosote leaves	0.24922
25% Drygrass	0.25007

EXAMPLE 28.4

(Target signature vector is unknown)

The purpose of including this experiment was to demonstrate that KFSSQ could be implemented even if the target signature vector was not known. The experiment conducted in Section 28.4.2.2 assumed that the exact knowledge of target signature vector, sagebrush, is provided *a priori*. In this example, this prior knowledge was not given. KFSSQ first used KFSSI to identify the unknown mixed target signature vector $\mathbf{t}^{\text{mix}}$ of (28.14) as sagebrush and then used KFSSE to estimate the sagebrush signature vector from $\mathbf{t}^{\text{mix}}$ as the desired signature that was used in the follow-up abundance quantifier, KFSSQ, to estimate its abundance fraction, 0.28. Compared with the value of 0.2504 in Table 28.13 and 0.25135 in Table 28.14, it was slightly off the true abundance 0.25, but was still very good. This made sense since the results in Tables 28.13 and 28.14 are obtained by assuming exact knowledge of the sagebrush as opposed to KFSSE-estimated sagebrush used in this example.

As noted further, when KFSSQ was implemented in the above estimation, two cases were considered. If the true signature vector $\mathbf{t}$ was used, the value of $\sigma_{\mathbf{u}}$ was determined by (28.11). On the other hand, if the target signature vector used in KFSSQ was estimated first by KFSSI followed by KFSSE, the value of $\sigma_{\mathbf{u}}$ in (28.11) was then determined by the value of $\sigma_{\mathbf{u}}$ used by the two estimators, that is, both (28.5) for KFSSE and (28.7) for KFSSI.

EXAMPLE 28.5

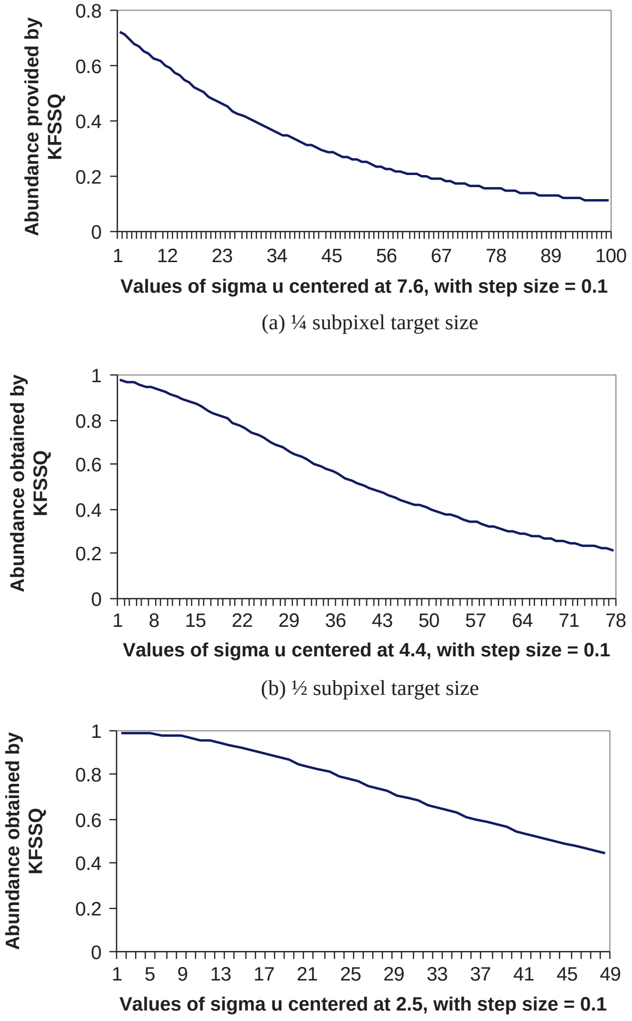
(Sensitivity of KFSSQ to $\sigma_{\mathbf{u}}$)

This experiment was designed to investigate the sensitivity of KFSSQ to the values of $\sigma_{\mathbf{u}}$ in quantification of subpixel targets. Experiments similar to those in Example 28.1 were conducted by using the values of $\sigma_{\mathbf{u}}$ given in Table 28.11 for three different target sizes (1/4, 1/2, and 3/4), where three proper ranges of $\sigma_{\mathbf{u}}$ for three different target sizes (1/4, 1/2, and 3/4) were fluctuated around 7.5, 4.3, and 2.4, respectively, for correct quantification. Figure 28.5(a)–(c) plots the results of KFSSQ for three different target sizes (1/4, 1/2, and 3/4) versus $\sigma_{\mathbf{u}}$ with step size 0.1.

As shown by the plotted curves in Figure 28.5(a)–(c), the abundance fractions quantified by KFSSQ were inversely proportional to the values of $\sigma_{\mathbf{u}}$. The curves in Figure 28.5 indicate that KFSSQ was indeed very sensitive to the value of $\sigma_{\mathbf{u}}$, where an appropriate values of $\sigma_{\mathbf{u}}$ must be carefully selected in order to perform correct quantification for subpixel targets.

Finally, three concluding remarks are worth noting.

1. One is the selection of $\sigma_{\mathbf{u}}$. Since there is generally no prior knowldege available for analysis, it is impossible to determine an appropriate value for $\sigma_{\mathbf{u}}$ *a priori*. A general approch is to use a trial-and-error approach to obtain *a posteriori* knowledge that can help to determine an adequate range of the $\sigma_{\mathbf{u}}$. From there, the values for $\sigma_{\mathbf{u}}$ can be properly selected by further experiments. Nevertheless, a common guideline is that the more similar to signatures in the database the target signature is, the smaller the $\sigma_{\mathbf{u}}$ is required.



(a) ¼ subpixel target size

(b) ½ subpixel target size

(c) ¾ subpixel target size

Figure 28.5 KFSSQ-estimated abundance fractions versus values of $\sigma_\mathbf{u}$ for different sizes of the subpixel target with step size 0.1.

2. It should be noted that KFSCSP techniques are signature vector-based not image pixel-based techniques. Its performance is determined by the spectral profiles of hyperspectral signature vectors not by data sample size or spectral correlation. Most importantly, KFSCSP technique is not designed or developed as a classifier. Therefore, no prior knowldege of training samples or classification information such as the number of classes to be classified is required.
3. More experiments were also conducted for other data sets such as Cuprite in Figure 1.9. Since the conclusions are nearly the same as what were presented in this section, the results are not included here.

28.5 Computer Simulations Using NIST-Gas Data

In order to demonstrate the utility of KFSSE in spectral estimation, identification, and quantification, the spectral signature vectors in the data set Δ to be used for experiments were those in Figure 1.10 available at the National Institute of Standard Technology (NIST)'s website http://www.nist.gov/srd/nist35.htm. It contains nine agent signatures $\{\mathbf{s}_i\}_{i=1}^{9}$, eight of them, $\{\mathbf{s}_i\}_{i=2}^{9}$ are composed of 880 bands, and only one of them, $\mathbf{s}_1$, consists of 825 bands.

As mentioned previously, since KFSCSP technique developed in this chapter is signature a vector-based and not an image-based technique, KFSSE, KFSSI, and KFSSQ are not designed for classification. Therefore, their performance will be evaluated by signature vector-based spectral measures such as SAM and SID rather than image classifiers.

28.5.1 KFSSE

To implement KFSSE, the system gain c_l in (28.5) was set to be 1 for all $1 \leq l \leq L$, the standard deviation of the state noise $\mathbf{v}$, $\sigma_{\mathbf{v}}$, was empirically set to 10^3 and the standard deviation of the measurement noise $\mathbf{u}$, $\sigma_{\mathbf{u}}$, was chosen to make $\mathrm{SNR} = 30\,\mathrm{dB}$, where the SNR was defined before. It should be noted that throughout our extensive experiments, the results demonstrated that KFSSE was robust to the selection of $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{v}}$. More specifically, once the value of $\sigma_{\mathbf{v}}$ was greater than 10^3, the $\sigma_{\mathbf{u}}$ had limited impact on the performance of KFSSE. Therefore, for KFSSE implemented in this chapter, the $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{v}}$ were fixed at 30 dB and 10^3, respectively.

Figure 28.6(a)–(e) shows KFSSE-estimated spectra of the nine reflectance spectra in Figure 1.10. Since the values of the corresponding estimation errors obtained by KFSSE were very small, Figure 28.7(a)–(e) shows the plots of taking logarithm function of ratios of estimation errors to their corresponding reflectance spectra in Figure 1.10 to have better visual assessment.

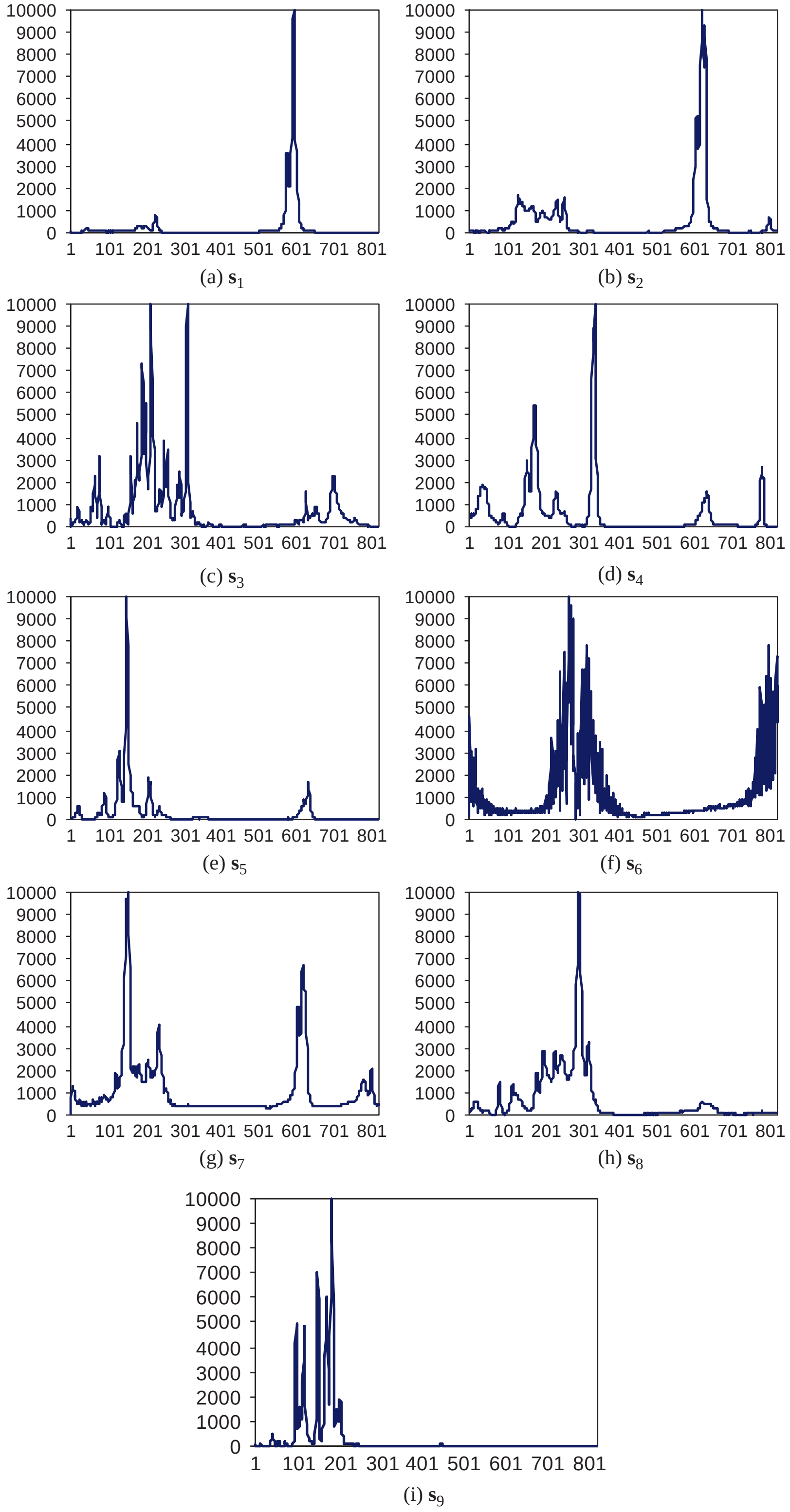
According to Figure 28.7, KFSSE performed very effectively in estimating spectral signatures with estimation errors nearly close to zero. Obviously, the large estimation errors always occurred at wavelengths where their spectral values had drastic changes. On the contrary, if there was a smooth transition between two wavelengths, the estimation errors were relatively small.

28.5.2 KFSSI

Two scenarios were implemented by KFSSI for subpixel target identification and mixed target identification with the target signature vector $\mathbf{t}$ to be identified assumed to be either known or unknown.

28.5.2.1 Subpixel Target Identification by KFSSI

First of all, we used Figure 4.1 to simulated a subpixel target embedded into a pixel vector. Assume that the subpixel target was the $\mathbf{s}_6$, and the background was $\mathbf{s}_7$. Figure 28.8(a) shows how a subpixel target $\mathbf{t}_1$ of ¾ pixel size was simulated. In order to simulate the subpixel target $\mathbf{t}_1$, we first simulated 1 pixel vector specified by $\mathbf{s}_6$ which was considered as the background signature and 3 pixel vectors specified by $\mathbf{s}_7$ to form a 4-pixel square panel as shown at the bottom layer in Figure 28.8(a). The 4-pixel square panel was then shrunk to its ¼ size by averaging all 4 pixel vectors to a 4-pixel square panel with the same spatial resolution 1.56 m shown at the top layer of Figure 28.8 where each pixel vector in the shrunk 4-pixel square panel was only ¼ size of its corresponding pixel vector at the bottom layer of Figure 28.8(a) as a result of $1/4(\mathbf{s}_6 + \mathbf{s}_7 + \mathbf{s}_7 + \mathbf{s}_7)$. This shrinking process is described in Figure 28.8. The shrunk 4-pixel vector at the top layer in Figure 28.8(a) was the desired pixel vector p_1 that contained a subpixel target $\mathbf{t}_1$ of ¾ pixel size specified by



(a) $\mathbf{s}_1$

(b) $\mathbf{s}_2$

(c) $\mathbf{s}_3$

(d) $\mathbf{s}_4$

(e) $\mathbf{s}_5$

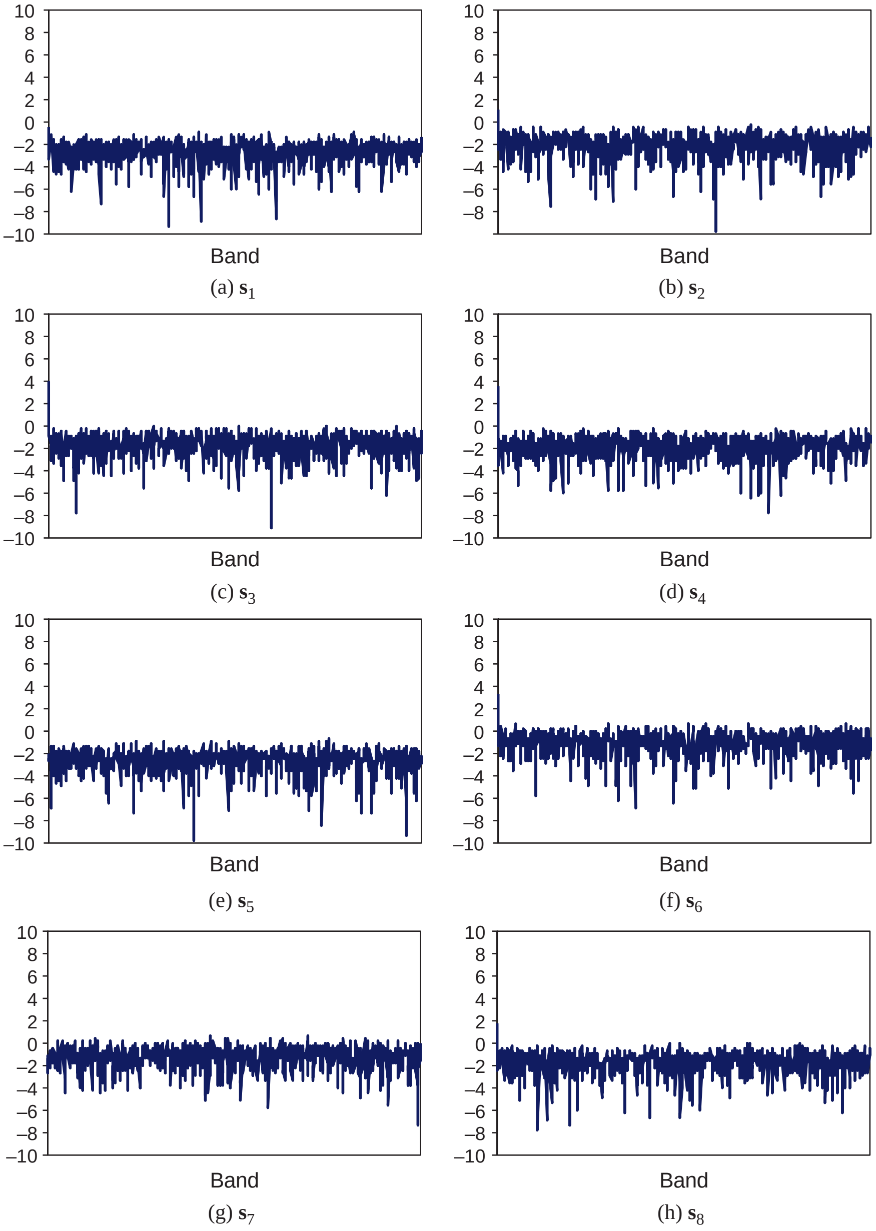
(f) $\mathbf{s}_6$

(g) $\mathbf{s}_7$

(h) $\mathbf{s}_8$

(i) $\mathbf{s}_9$

Figure 28.6 KFSSE-estimated spectra of the nine reflectance spectra in Figure 1.10.



(a) $\mathbf{s}_1$

(b) $\mathbf{s}_2$

(c) $\mathbf{s}_3$

(d) $\mathbf{s}_4$

(e) $\mathbf{s}_5$

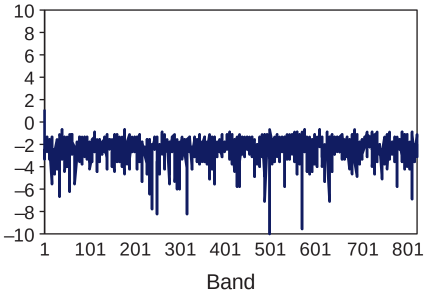
(f) $\mathbf{s}_6$

(g) $\mathbf{s}_7$

(h) $\mathbf{s}_8$

Figure 28.7 Ratios of the estimation errors to the five reflectance spectra in Figure 1.10 by taking the logarithm function.

Calcite. Similarly, two pixel vectors p_2 and p_3 shown in Figure 28.8(b) and (c) were also simualted in the same fashion as results of $1/4(\mathbf{s}_6 + \mathbf{s}_6 + \mathbf{s}_7 + \mathbf{s}_7)$ and $1/4(\mathbf{s}_6 + \mathbf{s}_6 + \mathbf{s}_6 + \mathbf{s}_7)$, respectively, where p_2 contained a subpixel target $\mathbf{t}_2$ of ½ pixel size specified by $\mathbf{s}_7$ and p_3 contained $\mathbf{t}_1$ of ¼ pixel size specified by $\mathbf{s}_6$.



(i) $\mathbf{s}_9$

Figure 28.7 (*Continued*)

KFSSI was implemented to identify an unknown subpixel target panel generated by the three panels p_1, p_2, and p_3 in Figure 28.8(a)–(c) via a database $\Delta = \{\mathbf{s}_6, \mathbf{s}_7\}$. The identification was carried out by first setting the values for both $\boldsymbol{\sigma}_\mathbf{u}$ and $\boldsymbol{\sigma}_\mathbf{v}$, then randomly choosing a matching signature vector from the database Δ to match the unknown subpixel target panel according to (28.8) and (28.9), and finally identifying the unknown subpixel target panel as the one which yielded the minimum LSE. The resulting LSE corresponding to different sizes of subpixel target panels are tabulated in Table 28.1 with various values of the standard deviation of noise $\mathbf{u}$, $\boldsymbol{\sigma}_\mathbf{u}$ and also with $\sigma_\mathbf{v} = 1000$. However, it should be noted that the selection of $\boldsymbol{\sigma}_\mathbf{u}$ empirically depended upon the target signature vector to be used for experiments. The simulated data conducted in this section were only used to demonstrate and illustrate the utility and effectiveness of KFSSI in signature identification under the impact of the parameter $\boldsymbol{\sigma}_\mathbf{u}$. In doing so, we had experimentally adjusted the value of $\boldsymbol{\sigma}_\mathbf{u}$ in accordance with our simulation data to dictate the impact of subpixel target size on the performance shown in Table 28.15. Figure 28.9(a) and (b) plot LSEs of $\mathbf{s}_6$ and $\mathbf{s}_7$, respectively, versus the values of $\boldsymbol{\sigma}_\mathbf{u}$ for three sizes of subpixel targets, ¾, ½, and ¼, where the values of $\boldsymbol{\sigma}_\mathbf{u}$ varied from 10 to 1000 with step size set to be 10.

Table 28.15 LSE corresponding to different sizes of subpixel target panels ($\mathbf{s}_6$) according to (28.10) via $\Delta = \{\mathbf{s}_6, \mathbf{s}_7\}$

Subpixel target panels ($\mathbf{s}_6$)	¼ pixel ($\sigma_\mathbf{u} \geq 450$)	½ pixel ($\sigma_\mathbf{u} \geq 30$)	¾ pixel ($\sigma_\mathbf{u} \leq 550$)
$\mathbf{s}_6$	328.68	452.01	535.10
$\mathbf{s}_7$	436.51	482.94	612.40

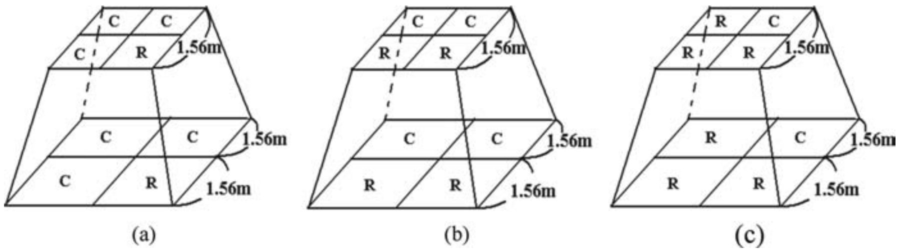
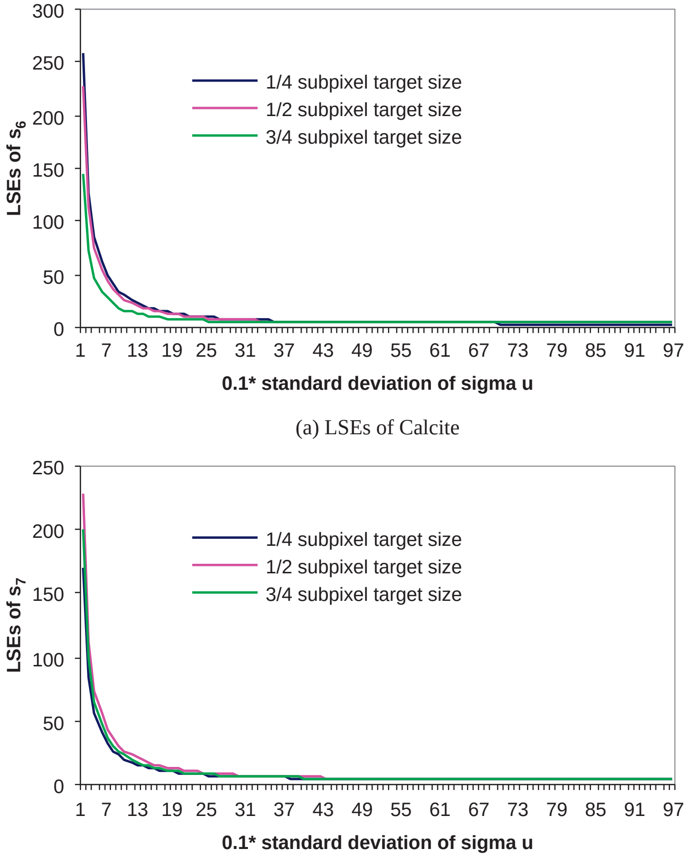


Figure 28.8 Simulations of subpixel target panels: (a) subpixel target panel with ¼ $\mathbf{s}_6$ + ¾ $\mathbf{s}_7$; (b) subpixel target panels with ½ $\mathbf{s}_6$ + ½ $\mathbf{s}_7$; (c) subpixel target panel with ¾ $\mathbf{s}_6$ + ¼ $\mathbf{s}_7$.



(a) LSEs of Calcite

(b) LSEs of Muscovite

Figure 28.9 Relationship between $\sigma_{\mathbf{u}}$ and the corresponding LSE according to different sizes of the subpixel target.

For example, in order to correctly identify the subpixel panel with ¼ size of a pixel, according to Table 28.1, the $\sigma_{\mathbf{u}}$ must be greater than 450 compared to the subpixel panel with ¾ size of a pixel which only requires $\sigma_{\mathbf{u}}$ smaller than 550. As pointed out at the end of Section 28.4, the selections of $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{v}}$ had significant impact on the performance of KFSSI and must be chosen appropriately in order to achieve acceptable LSEs. According to our exepriments, once $\sigma_{\mathbf{v}}$ was fixed, the minimum value of $\sigma_{\mathbf{u}}$ could be set approximately at the same order of magnitude in order to correctly identify the subpixel target as Calcite. Table 28.16 is included here to illustrate the relationship between $\sigma_{\mathbf{v}}$ and $\sigma_{\mathbf{u}}$ with three different values of $\sigma_{\mathbf{v}}$, and seven subpixel targets of size specified by (1/8, 2/8, 3/8, 4/8, 5/8, 6/8, and 7/8).

Table 28.17 tabulates the same experiments that were performed by SAM and SID, with the SID values included in parentheses where there was impossible for SAM and SID to detect the subpixel target panel if its size is smaller than ½ size of a panel. On the contrary, KFSSI could detect subpixel targets correctly even its size was smaller than ½ size of a pixel as long as $\sigma_{\mathbf{u}}$ was chosen to be the values tabulated in Table 28.2.

The above experiments demonstrated an important advantage of KFSSI. More specifically, KFSSI could perform well in the subpixel identification, even if the size of a subpixel target was

Table 28.16 The relationship between $\sigma_{\mathbf{v}}$ and $\sigma_{\mathbf{u}}$ with $\mathbf{s}_6$ as target embedded into $\mathbf{s}_7$

$\sigma_{\mathbf{v}}$	$\sqrt{10}$	$\sqrt{100}$	$\sqrt{1000}$
1/8 subpixel size	$\sigma_{\mathbf{u}} \geq 150$	$\sigma_{\mathbf{u}} \geq 150$	$\sigma_{\mathbf{u}} \geq 500$
2/8 subpixel size	$\sigma_{\mathbf{u}} \geq 150$	$\sigma_{\mathbf{u}} \geq 140$	$\sigma_{\mathbf{u}} \geq 450$
3/8 subpixel size	$\sigma_{\mathbf{u}} \geq 150$	$\sigma_{\mathbf{u}} \geq 120$	$\sigma_{\mathbf{u}} \geq 400$
4/8 subpixel size	$\sigma_{\mathbf{u}} \geq 5$	$\sigma_{\mathbf{u}} \geq 25$	$\sigma_{\mathbf{u}} \geq 30$
5/8 subpixel size	$\sigma_{\mathbf{u}} \leq 90$	$\sigma_{\mathbf{u}} \leq 250$	$\sigma_{\mathbf{u}} \leq 800$
6/8 subpixel size	$\sigma_{\mathbf{u}} \leq 50$	$\sigma_{\mathbf{u}} \leq 150$	$\sigma_{\mathbf{u}} \leq 500$
7/8 subpixel size	all values	all values	$\sigma_{\mathbf{u}} \leq 500$

Table 28.17 Estimated size of subpixel panels with $\mathbf{s}_6$ as target embedded into $\mathbf{s}_7$ by SAM and SID

Subpixel size	$\mathbf{s}_6$	$\mathbf{s}_7$
1/8	1.1180 (1.6294)	0.1570 (0.0702)
1/4	0.9410 (1.1529)	0.3340 (0.2206)
3/8	0.7525 (0.8004)	0.5226 (0.4211)
1/2	0.5651 (0.5282)	0.7099 (0.6636)
5/8	0.3911 (0.3172)	0.8840 (0.9492)
3/4	0.2380 (0.1579)	1.0370 (1.2890)
7/8	0.1081 (0.0483)	1.1669 (1.7161)

less than ½ of ground sampling distance, that is, pixel resolution, which could not be achieved by any other spectral measure.

28.5.2.2 Mixed Target Identification by KFSSI

In order to make our experiments more interesting and appealing, we further simulated a mixed pixel vector $\mathbf{t}^{\text{mix}}$ by mixing ¼ $\mathbf{s}_3$, ¼ $\mathbf{s}_4$, ¼ $\mathbf{s}_6$, and ¼ $\mathbf{s}_7$ together. In other words, $\mathbf{t}^{\text{mix}}$ was simulated by mixing equal amount of abundance among the four signature vectors as follows:

$$\mathbf{t}^{\text{mix}} = 0.25\mathbf{s}_3 + 0.25\mathbf{s}_4 + 0.25\mathbf{s}_6 + 0.25\mathbf{s}_7 \tag{28.15}$$

shown in Figure 28.1. In this case, no signature vector was preferred to another. KFSSI was used to identify unknown target signature vector $\mathbf{t}$ present in the mixed pixel vector $\mathbf{t}^{\text{mix}}$ using the database $\Delta = \{\mathbf{s}_3, \mathbf{s}_4, \mathbf{s}_6, \mathbf{s}_7\}$. According to the composition of $\mathbf{t}^{\text{mix}}$, all the four signature vectors had equal opportunity to be identified as the target signature vector. The resulting LSEs corresponding to different matching components to the mixed pixel vector are shown in Table 28.18.

Table 28.18 LSE between mixed pixel and different matching signature vectors according to (28.10)

	$\mathbf{s}_3$	$\mathbf{s}_4$	$\mathbf{s}_6$	$\mathbf{s}_7$
$\sigma_{\mathbf{u}} \leq 400$	1136.2	1686.6	628.64	444.48
$400 \leq \sigma_{\mathbf{u}} \leq 2000$	328.63	320.69	296.54	346.91
$\sigma_{\mathbf{u}} \geq 2000$	334.15	312.87	337.90	351.16

28.5.3 KFSSQ

This section presents experiments to further demonstrate the use of KFSSQ in quantification of subpixel targets and target signature vectors present in mixed pixels. However, unlike KFSSI, which was used to identify an unknown target signature vector **t**, KFSSQ could also be implemented for the target signature vector **t**, which was either unknown or known as *a priori*. It should also be noted that unlike KFSSI, there was no state noise vector **v** in the state equation (28.12) implemented by KFSSQ.

28.5.3.1 Subpixel Target Identification by KFSSQ

Using the same subpixel targets $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$ simulated in Section 28.4.2.1, we implemented KFSSQ to quantify these three Calcite-simulated subpixel target panels $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$ specified by ¾, ½, and ¼ size of a pixel in Figure 28.8 with two scenarios described in the following examples.

EXAMPLE 28.6

(Target signature vector t is known)

This example assumed that the target signature vector **t** was known to be $\mathbf{s}_6$. In this case, the pair of (28.11) and (28.12) was implemented to quantify three subpixel targets $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$ of $\mathbf{s}_6$. It should be noted that since the subpixel target panels only occupied part of a pixel; its size could be interpreted as portion of abundance fraction in terms of percentage. In this case, the three subpixel target panels $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$ with size of ¾, ½, and ¼ could be considered as targets with abundance fractions 75%, 50%, and 25%, respectively, as indicated in Table 28.19.

As shown in Table 28.19, KFSSQ-estimated quantification results were very accurate with appropriate chosen values of $\sigma_{\mathbf{u}}$ with no occurrence of $\sigma_{\mathbf{v}}$.

EXAMPLE 28.7

(Target signature t is unknown)

Unlike Example 28.1, the prior knowledge of target signature vector **t** was not given in this example. In this case, KFSSQ must first identify via (28.5) and (28.6) the target signature vector **t** that specified the three subpixel targets $\mathbf{t}_1$, $\mathbf{t}_2$, and $\mathbf{t}_3$ where KFSSE was used for this purpose and the estimated target denoted by $\hat{\mathbf{t}}^{\text{KFSSE}}$. The estimated $\hat{\mathbf{t}}^{\text{KFSSE}}$ was then used to replace **t** in (28.11) to produce the abundance vector $\hat{\boldsymbol{\alpha}}^{\text{KFSSQ}}$ specified by (28.12) in Table 28.6. As a result, there were two different values of $\sigma_{\mathbf{u}}$ used for measurement noise in KFSSQ, one in (28.5) used for KFSSE and the other in (28.13) used for KFSSQ. Since both were not correlated, they could be determined independently as tabulated in Table 28.20.

Due to the fact that KFSSE could not correctly estimate the target signature vector **t** provided that the abundance of subpixel target **t** was below 50%, which was reasonable, the results for the size of subpixel target being ¼ are not included in Table 28.20.

28.5.3.2 Mixed Target Quantification by KFSSQ

In this subsection, we used the same mixed pixel vector, $\mathbf{t}^{\text{mix}}$ simulated in Section 28.4.2.2 for further experiments. KFSSQ is implemented to quantify the four signature vectors, $\mathbf{s}_3$, $\mathbf{s}_4$, $\mathbf{s}_6$, and

Table 28.19 KFSSQ-estimated abundance fractions

KFSSQ	Quantification	$\sigma_{\mathbf{u}}$ in KFSSQ
$\mathbf{t}_1 = 25\%\ \mathbf{s}_6$	0.2527	55000
$\mathbf{t}_2 = 50\%\ \mathbf{s}_6$	0.5000	31000
$\mathbf{t}_3 = 75\%\ \mathbf{s}_6$	0.7493	18000

Table 28.20 KFSSQ-estimated abundance fractions

KFSSQ	Quantification	$\sigma_{\mathbf{u}}$ in KFSSE	$\sigma_{\mathbf{u}}$ in KFSSQ
$\mathbf{t}_2 = 50\% \ \mathbf{s}_6$	0.4997	1000	46500
$\mathbf{t}_3 = 75\% \ \mathbf{s}_6$	0.7558	1000	30000

$\mathbf{s}_7$, each of which shared a 25% abundance fraction in the pixel vector $\mathbf{t}^{\text{mix}}$. Like Section 28.4.3.1 of subpixel target quantification, two scenarios were also considered.

EXAMPLE 28.8

(All the four target signatures are known)

This example assumed that all the four signature vectors were known. The pair of (28.11) and (28.12) was implemented for KFSSQ. Table 28.21 tabulates the quantification results of the four signature vectors with their appropriately chosen values of $\sigma_{\mathbf{u}}$. However, it should be noted that the sensitivity of $\sigma_{\mathbf{u}}$ was generally determined by various applications. For example, when KFSSI and KFSSQ were implemented, the results would be sensitive to the value chosen for $\sigma_{\mathbf{u}}$ as shown in Tables 28.18 and 28.20. On the other hand, if KFSSE was implemented as an estimator, the results would be rather robust to the $\sigma_{\mathbf{u}}$.

As we can see from Table 28.21, KFSSQ-estimated abundance fractions of all the four signature vectors were very accurate and close to true values.

In order to make comparison, the results produced by FCLS were also included and tabulated in Table 28.22, where their results comparable to those obtained in Table 28.21.

EXAMPLE 28.9

(Sensitivity of KFSSQ to $\sigma_{\mathbf{u}}$)

This experiment was designed to investigate sensitivity of KFSSQ to the values of $\sigma_{\mathbf{u}}$ in quantification of subpixel targets. Performing similar experiments conducted for Example 28.1, the results tabulated in

Table 28.21 Results of implementing KFSSQ for mixed pixel with target signature vectors known

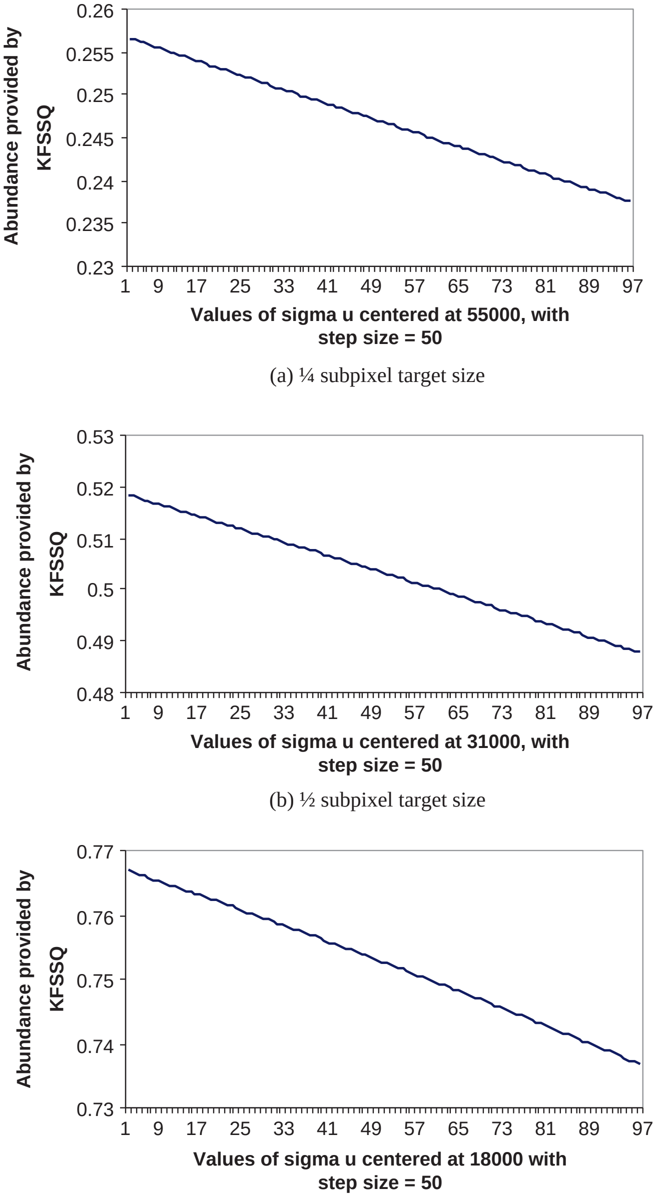
KFSSQ	Quantification results	$\sigma_{\mathbf{u}}$
25% $\mathbf{s}_3$	0.2563	47000
25% $\mathbf{s}_4$	0.2519	45000
25% $\mathbf{s}_6$	0.2518	51000
25% $\mathbf{s}_7$	0.2524	51000

Table 28.22 The abundance fractions of the four target signature vectors in $\mathbf{t}^{\text{mix}}$ estimated by FCLS

FCLS	Quantification results
25% $\mathbf{s}_3$	0.2500
25% $\mathbf{s}_4$	0.2500
25% $\mathbf{s}_6$	0.2500
25% $\mathbf{s}_7$	0.2500

Table 28.22 were used to select three proper ranges of $\sigma_{\mathbf{u}}$ for three different target sizes (1/4, 1/2, and 3/4) for correct quantification. Figure 28.10(a)–(c) plots their results of KFSSQ for three different target sizes (1/4, 1/2, and 3/4) versus $\sigma_{\mathbf{u}}$ with step size 0.1.

The curves plotted in Figure 28.10(a)–(c) show that the abundance fractions quantified by KFSSQ were inversely proportional to the values of $\sigma_{\mathbf{u}}$, where the curves were flatted out throughout the range of $\sigma_{\mathbf{u}}$. This



(a) ¼ subpixel target size

(b) ½ subpixel target size

(c) ¾ subpixel target size

Figure 28.10 Results of KFSSQ versus values of $\sigma_{\mathbf{u}}$ for different sizes of the subpixel target.

indicates that KFSSQ was indeed very sensitive to the value of $\sigma_{\mathbf{u}}$ where an appropriate value of $\sigma_{\mathbf{u}}$ must be carefully selected in order to perform correct quantification for subpixel targets.

Finally, a concluding remark on the selection of $\sigma_{\mathbf{u}}$ is worthwhile. Since there is generally no prior knowledge available for analysis, it is impossible to determine an appropriate value for $\sigma_{\mathbf{u}}$ *a priori*. A general approach is to use a trial-and-error approach to obtain *a posteriori* knowledge that can help to determine an adequate range of the $\sigma_{\mathbf{u}}$. From there, the values for $\sigma_{\mathbf{u}}$ can be properly selected by further experiments. Nevertheless, a common guideline is that the more similar the target signature is to signatures in the database, the smaller is the $\sigma_{\mathbf{u}}$ required.

28.6 Real Data Experiments

In this section, the Hyperspectral Digital Imagery Collection Experiment (HYDICE) image scene shown in Figure 1.15(a) was used for experiments, which has a size of 64×64 pixel vectors with 15 panels in the scene and the ground truth map provided in Figure 1.15(b).

28.6.1 KFSSE

To implement KFSSE on the HYDICE real data, the variances of the two additive Gaussian noises $\mathbf{v}$ and $\mathbf{u}$ in the state and measurement equations were set to $\sigma_{\mathbf{v}} = 10^3$ according to experiments conducted by Chang and Brumbley (1999a, 1999b), and $\sigma_{\mathbf{u}}$ was chosen as to achieve SNR = 30 dB. Figure 28.11(a)–(e) shows estimates of the five panel signatures $\mathbf{p}_i$ for $i = 1, 2, \ldots, 5$, while Figure 28.12(a)–(e) shows the ratios of estimation errors to their corresponding reflectance spectra in Figure 1.16 in terms of the logarithm function so as to achieve better visualization for assessement.

As shown in Figure 28.12, KFSSE performed relatively well in estimating the five panel signatures.

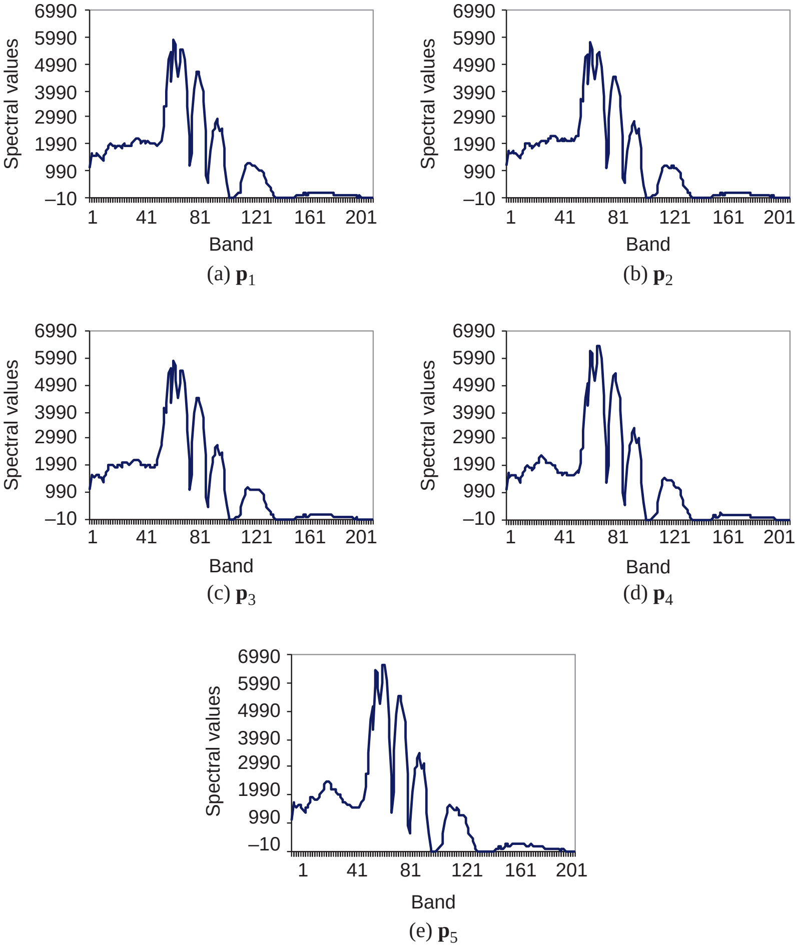
28.6.2 KFSSI

In this subsection, the 19 R panel pixels in Figure 1.15(b) were used for experiments to evaulate the performance of KFSSI in panel pixel identification. The variance of the measurement noise must be adjusted in accordance with the problems to be considered. In implementing KFSSI, we assumed that the five panel signatures $\{\mathbf{p}_i\}_{i=1}^{5}$ were given as a database Δ. KFSSI then used the database Δ to identify each of the 19 R panel pixels. Once again, the identification was implemented by first setting the values for both $\sigma_{\mathbf{u}}$ and $\sigma_{\mathbf{v}}$, then randomly choosing a matching signature from the database Δ to match the unknown subpixel target panel according to (28.8) and (28.9), and finally identifying the unknown subpixel target panel as the one that yielded the minimum LSE. The experimental results are tabulated in Table 28.23, where an identification is highlighted by shade and an incorrectly identified panel pixel is labeled by a cross. The range of $\sigma_{\mathbf{u}}$ chosen for each panel pixel is also listed in the last column with $\sigma_{\mathbf{v}}$ fixed at 10.

According to Table 28.23, 16 out of 19 R panel pixels were correctly identified except subpixel pixel p_{23}, which was wrongly identified as $\mathbf{p}_1$.

In order to make further comparison, Table 28.24 also tabulates the results obtained by SAM and SID, where the values in the upper and lower values in each entry were obtained by SAM and SID, respectively, with the SID values in parentheses. Like Table 28.23, the identification is highlighted by shade and the incorrectly identified panel pixels are labeled by a cross.

As shown in Table 28.24, both SAM and SID made the same six identification errors, p_{13}, p_{33}, p_{411}, p_{412}, p_{43}, and p_{53}, four of which were subpixel panels, p_{13}, p_{33}, p_{43}, and p_{53}. In other words, both the SAM and SID identified all the five subpixel panels $\{p_{i3}\}_{i=1}^{5}$ as $\mathbf{p}_2$ and only p_{23} was



(a) $\mathbf{p}_1$

(b) $\mathbf{p}_2$

(c) $\mathbf{p}_3$

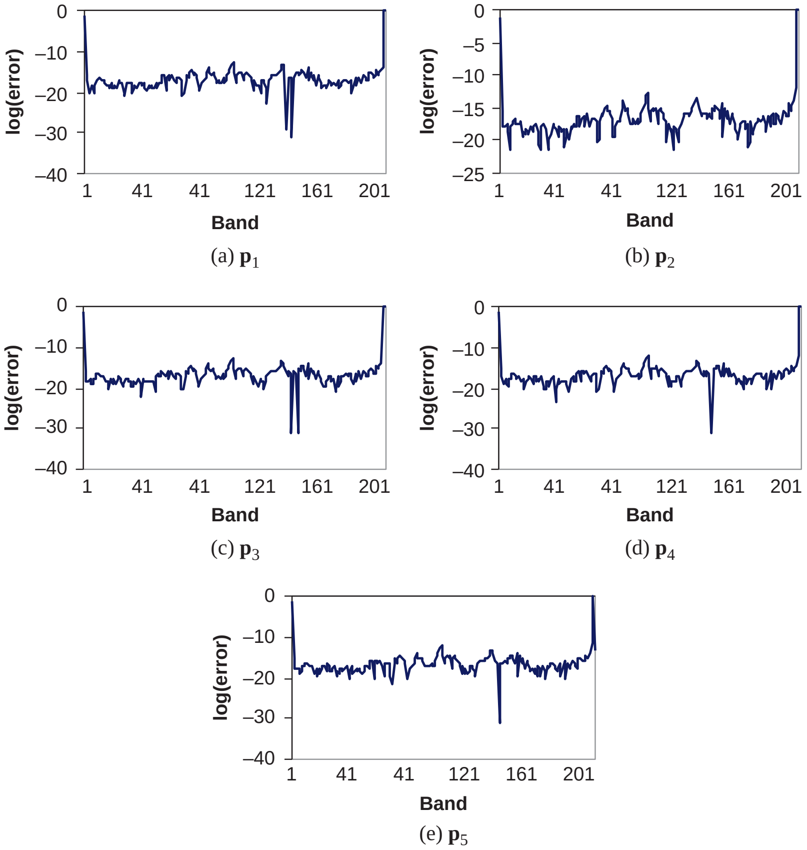
(d) $\mathbf{p}_4$

(e) $\mathbf{p}_5$

Figure 28.11 KFSSE-estimated spectra of the five reflectance spectra in Figure 1.16.

identified as $\mathbf{p}_2$ correctly. Comparing Table 28.23 with Table 28.24, KFSSI performed significantly better than spectral measures.

Since the 19 R panel pixels were directly extracted from the real HYDICE image scene, there existed sample correlation among the 19 R panel pixels, of which KFSSI and spectral measures did not take advantage of. In order to further investigate this issue, these 19 R panel pixels were first extracted from the original image cube and then concatenated pixel-by-pixel starting from panel pixels p_{11}, p_{12}, p_{13}, p_{211}, p_{221}, p_{22}, p_{23}, p_{311}, p_{212}, p_{32}, p_{33}, p_{23} to p_{411}, p_{412}, p_{42}, p_3, p_{511}, p_{521}, p_{52}, and p_{53} to retain sample correlation among panel pixels so that (28.2) implemented by KFLU by Chang and Brumbley (1999a, 1999b) could be used to capture such sample correlation as it processed these 19 lined-up R panel pixels as a vector. For a fair comparison, the same level of prior knowledge about five panel signatures $\{\mathbf{p}_i\}_{i=1}^{5}$ used by KFSSI was also assumed for KFLU. Table 28.25 tabulates the identification results produced by KFLU, which also made four identification errors, p_{13}, p_{32}, p_{33}, and p_{53}, three of which were subpixel panels, p_{13}, p_{33}, and p_{53}, where an identification is highlighted by shade and an incorrectly identifiied panel pixel is labeled by a cross.



(a) $\mathbf{p}_1$

(b) $\mathbf{p}_2$

(c) $\mathbf{p}_3$

(d) $\mathbf{p}_4$

(e) $\mathbf{p}_5$

Figure 28.12 Ratios of estimation errors to the five reflectance spectra in Figure 1.16.

Comparing Table 28.25 to Tables 28.23 and 28.24, KFLU performed better than spectral measures due to its use of pixel-to-pixel sample correlation, but worse than KFSSI due to its failure of capturing band-to-band spectral correlation as KFSSI did.

Finally, Table 28.26 summarizes incorrect panel pixel identification results produced by KFSSI in Table 28.23, spectral measures SAM/SID in Table 28.24 and KFLU in Table 28.25.

According to Table 28.26, the best results were produced by KFSSI, with, the worst being SAM/SID. Since two spectral measures, SAM/SID, did not take advantage of pixel-to-pixel or band-to-band sample correlation, their results in Table 28.24 were the worst as expected with six identification errors in Table 28.26. KFLU performed better than SAM/SID in that it reduced six errors made by the SAM/SID to five errors in Table 28.26 by making use of its state equation to capture pixel-to-pixel sample correaltion among the 19 panel pixels. However, it was KFSSI that really showed its strength in identification by reducing five errors to only three errors in Table 28.26. Such imporvement was a result of KFSSI's using a state equation to keep track of band-to-band correlation, which turned out to be more crucial than the pixel-to-pixel correaltion used by KFLU. Furthermore, SAM/SID and KFLU made nearly the same identification errors, that is, the four errors on the four subpixel panels, p_{13}, p_{33}, p_{43}, and p_{53} made by KFLU and also made by SAM/SID in Table 28.26. Quite contrarily, the five out of six errors made by SAM and SID and three out of five errors made by KFLU are correctly identified by KFSSI. Instead, two out of three

Table 28.23 LSEs corresponding to different 19 panel pixels using different matching panel signatures by KFSSI according to (28.10)

	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$	$\sigma_{\mathbf{u}}$
p_{11}	328.1	328.8	328.8	328.1	328.3	[45, 450]
p_{12}	169.5	428.0	462.2	561.3	846.5	< 1
p_{13}	212.6	259.3	253.4	228.1	298.9	< 0.2
p_{211}	1216.8	539.9	897.8	870.7	871.6	< 0.5
p_{221}	971.3	523.3	864.7	661.0	677.5	[0.01, 0.1]
p_{22}	280.8	240.3	290.0	2279.7	2758.4	[0.01, 0.2]
~~p_{23}~~	223.3	244.0	232.3	1575.4	2072.0	Any value
p_{311}	561.2	286.6	205.9	307.3	428.0	< 0.1
p_{312}	1319.9	928.8	328.8	2077.1	2400.3	< 0.5
~~p_{32}~~	1154.7	320.7	594.3	1309.0	1303.4	Any value
~~p_{33}~~	1943.0	173.5	291.5	5909.0	6433.1	Any value
p_{411}	345.8	345.7	346.4	344.0	345.2	[30, 60]
p_{412}	351.9	353.4	364.0	335.0	335.7	[5, 25]
p_{42}	338.8	335.4	339.3	328.5	330.1	[15, 45]
p_{43}	240.6	269.8	265.6	228.2	245.5	[2, 8]
p_{511}	2293.8	1865.3	2177.6	468.8	299.5	< 10
p_{521}	656.0	560.2	597.1	383.5	293.8	< 45
p_{52}	436.2	354.1	395.6	168.3	93.1	< 25
p_{53}	274.6	228.6	228.4	268.5	253.5	[5, 80]

Table 28.24 Similarity values obtained by comparing the five subpixel panels $\{p_{i3}\}_{i=1}^{5}$ against the five panel signatures, $\{\mathbf{p}_i\}_{i=1}^{5}$ using SAM and SID

	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
p_{11}	0.0533 (0.0061)	0.0883 (0.0143)	0.0980 (0.0178)	0.1005 (0.0121)	0.1045 (0.0135)
p_{12}	0.0084 (0.0000)	0.0413 (0.0022)	0.0656 (0.0052)	0.1157 (0.0172)	0.1255 (0.0232)
~~p_{13}~~	0.0579 (0.0070)	0.0453 (0.0040)	0.0737 (0.0068)	0.1512 (0.0338)	0.1636 (0.0437)
p_{211}	0.0436 (0.0026)	0.0151 (0.0000)	0.0401 (0.0025)	0.1459 (0.0252)	0.1536 (0.0310)
p_{221}	0.0451 (0.0027)	0.0163 (0.0000)	0.0387 (0.0025)	0.1490 (0.0260)	0.1565 (0.0318)
p_{22}	0.0387 (0.0025)	0.0169 (0.0000)	0.0473 (0.0027)	0.1378 (0.0234)	0.1465 (0.0300)
p_{23}	0.0662 (0.0068)	0.0336 (0.0019)	0.0647 (0.0050)	0.1639 (0.0352)	0.1749 (0.0444)
p_{311}	0.0808 (0.0082)	0.0817 (0.0083)	0.0538 (0.0040)	0.1556 (0.0265)	0.1565 (0.0299)
p_{312}	0.0861 (0.0100)	0.0880 (0.0099)	0.0579 (0.0043)	0.1593 (0.0301)	0.1601 (0.0342)
p_{32}	0.0845 (0.0084)	0.0479 (0.0026)	0.0467 (0.0024)	0.1819 (0.0396)	0.1907 (0.0477)
~~p_{33}~~	0.1065 (0.0152)	0.0681 (0.0064)	0.0767 (0.0071)	0.2022 (0.0526)	0.2125 (0.0630)
~~p_{411}~~	0.1535 (0.0316)	0.1897 (0.0473)	0.2031 (0.0536)	0.0467 (0.0035)	0.0411 (0.0027)
~~p_{412}~~	0.1764 (0.0444)	0.2119 (0.0618)	0.2242 (0.0683)	0.0705 (0.0085)	0.0625 (0.0068)
p_{42}	0.1170 (0.0175)	0.1503 (0.0286)	0.1671 (0.0350)	0.0101 (0.0000)	0.0252 (0.0014)
~~p_{43}~~	0.0775 (0.0069)	0.0644 (0.0039)	0.0959 (0.0090)	0.1424 (0.0273)	0.1563 (0.0363)
p_{511}	0.1380 (0.0278)	0.1713 (0.0414)	0.1844 (0.0475)	0.0368 (0.0035)	0.0177 (0.0000)
p_{521}	0.1942 (0.0609)	0.2291 (0.0814)	0.2390 (0.0876)	0.0922 (0.0184)	0.0759 (0.0111)
p_{52}	0.1521 (0.0343)	0.1862 (0.0497)	0.1973 (0.0551)	0.0510 (0.0058)	0.0327 (0.0019)
~~p_{53}~~	0.0747 (0.0067)	0.0565 (0.0031)	0.0885 (0.0079)	0.1466 (0.0287)	0.1593 (0.0372)

Table 28.25 KFLU-unmixed results for 19 R lined-up pixels

	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
p_{11}	1.8861	−0.9656	0.1757	−0.8647	0.8395
p_{12}	0.8643	0.1195	0.0080	0.0678	−0.0486
~~p_{13}~~	0.2492	0.8463	−0.1677	0.7971	−0.7910
p_{211}	0.0858	0.8540	0.0997	−0.2254	0.2138
p_{212}	0.1913	0.8060	0.1103	−0.3100	0.2527
p_{22}	−0.0210	0.8857	0.0405	0.0501	0.0214
p_{23}	−0.2559	1.4540	−0.2504	0.4854	0.4880
p_{311}	0.7414	−1.2288	1.5070	−0.8208	0.7968
p_{312}	0.9121	−1.6183	1.7738	−0.7464	0.7100
~~p_{32}~~	−0.6778	1.0766	0.5740	0.6857	−0.6361
~~p_{33}~~	−0.9756	1.7707	0.1450	0.8814	−0.8706
p_{411}	0.6155	−0.9269	0.1519	0.7564	0.4500
p_{412}	0.4704	−1.0378	0.2250	0.7165	0.6234
p_{42}	−0.0679	0.0421	−0.0056	1.0091	0.0441
~~p_{43}~~	−1.0178	1.9227	−0.3714	1.5175	−1.1170
p_{511}	0.0556	−0.1320	−0.0032	−0.1444	1.2131
p_{512}	0.6630	−1.2352	0.1986	−0.7602	2.1914
p_{52}	0.2635	−0.6100	0.1945	−0.2421	1.4157
~~p_{53}~~	−0.9822	1.9771	−0.3898	1.1475	−0.8207

errors made in Table 28.26 by KFSSI are among those that were correctly identified by SAM/SID and KFLU in Tables 28.23 and 28.24. These interesting findings in Table 28.26 demonstrated the significant difference of KFSSI from SAM/SID and KFLU in the sense of their functionalities.

28.6.3 KFSSQ

As noted, the subpixel panels, p_{13}, p_{23}, p_{33}, p_{43}, and p_{53}, had the same size of 1 m × 1 m smaller than the 1.56 m × 1.56 m spatial resolution. This indicated that the size of these five subpixel panels is $1/(1.56\,\text{m})^2 = 0.41091\,\text{m}^2$, which was smaller than 50% of pixel size. Therefore, KFSSE could not accurately estimate the subpixel panels without prior knowledge. In this case, we assumed that the five panel signatures $\{\mathbf{p}_i\}_{i=1}^{5}$ in Figure 1.16 were provided as *a priori*. Table 28.27 tabulates the quantification results of each of the 19 R panel pixels with their respective chosen standard deviations of the measurement noise $\sigma_{\mathbf{u}}$, where the $\mathbf{p}_i$ was used as the prior knowledge for subpixel panel p_{i3} for each $i = 1, 2, 3, 4,$ and 5.

As shown in Table 28.27, KFSSQ performed very well in quantifying the five subpixel panels, provided that the values of $\sigma_{\mathbf{u}}$ were chosen approprately as given in the table. A remark similar to the one made on the selection of $\sigma_{\mathbf{u}}$ at the end of Section 28.4 was also applied to real image experiments.

Table 28.26 Panel pixels wrongly identified by the three scenarios

	Panel pixels wrongly identified
SAM/SID	$p_{13}, p_{33}, p_{411}, p_{412}, p_{43}, p_{53}$
KFLU	$p_{13}, p_{32}, p_{33}, p_{43}, p_{53}$
KFSSI	p_{23}, p_{32}, p_{33}

Table 28.27 Abundance fraction estimated by KFSSQ and $\sigma_{\mathbf{u}}$ for subpixel target panels using known target signatures

Subpixel panel	Size of subpixel target	$\sigma_{\mathbf{u}}$
p_{11}	0.9993	9300
p_{12}	0.4073	36000
p_{13}	0.4109	33050
p_{211}	0.5236	29500
p_{221}	0.3435	43000
p_{22}	0.6931	19300
p_{23}	0.4106	33600
p_{311}	0.8626	13000
p_{312}	0.9965	7500
p_{32}	0.5399	28000
p_{33}	0.4109	33600
p_{411}	0.9992	8900
p_{412}	0.9975	7000
p_{42}	0.7279	20500
p_{43}	0.4115	33300
p_{511}	0.7308	20000
p_{521}	0.9985	11500
p_{52}	0.7742	19500
p_{53}	0.4107	33300

Table 28.28 Comparison among KFLU, KFSSE, KFSSI, and KFSSQ

	KFLU	KFSSE	KFSSI	KFSSQ
Functionality	Abundance estimator $\hat{\boldsymbol{\alpha}}_{(n)}^{\text{KFLU}}$	Signature estimator $\hat{t}_l^{\text{KFSSE}}$	Signature identifier $\hat{t}_l^{\text{KFSSI}}$	Abundance quantifier $\hat{\alpha}_l^{\text{KFSSQ}}$
Linear mixing model	Yes	No	No	No
Sample correlation	Yes	No	No	No
Data samples	Image pixel vector $\mathbf{r}(n)$	lth band scalar r_l	lth band scalar s_l	lth band scalar r_l
Initial condition	$\hat{\boldsymbol{\alpha}}(1) = \mathbf{0}$	$\hat{t}_1^{\text{KFSSE}} = 0$	$\hat{t}_1^{\text{KFSSI}} = 0$	$\hat{\alpha}_1^{\text{KFSSQ}} = 0$
Measurement input	Image pixel $\mathbf{r}(n)$	lth band scalar r_l	r_l and s_l	r_l and s_l
Measurement output	$\boldsymbol{\alpha}(n)$	$\hat{t}_l^{\text{KFSSE}}$	$\hat{t}_l^{\text{KFSSI}}$	$\hat{\alpha}_l^{\text{KFSSQ}}$
State input	$\hat{\boldsymbol{\alpha}}_n^{\text{KFLU}}(n)$	$\hat{t}_l^{\text{KFSSE}}$	$\hat{t}_l^{\text{KFSSI}}$	$\hat{\alpha}_l^{\text{KFSSQ}}$
State output	$\hat{\boldsymbol{\alpha}}_{n+1}^{\text{KFLU}}(n)$	$\hat{t}_{l+1}^{\text{KFSSE}}$	$\hat{t}_{l+1}^{\text{KFSSI}}$	$\hat{\alpha}_{l+1}^{\text{KFSSQ}}$

28.7 Conclusions

Spectral characterization provides important and crucial features in target discrimination, detection, classification, identification, and quantification. This chapter presents new applications of Kalman filtering in spectral characterization and further develops three KFSCSP techniques, a spectral abundance estimator, called KFSSE, a spectral quantifier, called KFSSQ, and a spectral identification, called KFSSI. The proposed KFSSE, KFSSI, and KFSSQ are quite different from the Kalman filter-based linear spectral unmixing developed by Chang and Brumbley (1999a, 1999b) in the sense that while the former is designed for one-dimensional spectral data the latter is developed for unmixing image data. A comparison among KFLU, KFSSE, KFSSI, and KFSSQ is summarized in Table 28.28.

It is worth noting that depending upon applications, the measurement and state equations used to describe KFSSE, KFSSI, and KFSSQ are all different. For example, both KFSSE and KFSSI have the state noise **v** included in their state equations, while KFSSQ does not have one in its state equation. Moreover, KFSSE uses the same target signature vector **t** in both measurement equation and state equation compared with KFSSI, which uses a matching signature vector **s** in its measurement equation and the target signature vector **t** in its state equation. Due to the use of two different signature vectors, **t** and **s**, KFSSI is more sensitive to both measurement noise **u** and state noise **v**, both of which must be chosen appropriately and are generally selected at the magnitude of the same order. On the other hand, KFSSE uses the target signature vector **t** in these two equations. So, it is more robust to the noise in both equations. Since KFSSQ performs spectral quantification using the zero-holder interpolator with noise involved this leads to the fact that measurement noise **u** has a significant impact on the performance of KFSSQ as demonstrated in our experiments.

29

Wavelet Representation for Hyperspectral Signals

Wavelet analysis has been used successfully in many areas in signal and image processing. Its applications to remote sensing have also been evidenced by many publications. This chapter presents a new application of wavelets in hyperspectral signature representation for spectral characterization. In particular, a new algorithm, called wavelet-based signature characterization algorithm (WSCA), is developed for hyperspectral signature discrimination, classification, and identification. The key idea of WSCA is to decompose a hyperspectral signature vector into two signature components, referred to as detail and approximation signatures, respectively, via the discrete wavelet transform (DWT) specified by Mallet's algorithm where two types of filters, high-pass and low-pass filters, are constructed to generate these two components. Two specific functions, called "wavelet function," and "scaling function," are used for DWT to span two orthogonal vector spaces. Since the wavelet function is effective in capturing the details of a signature vector that corresponds to the high-frequency domain information of the original signature vector, it can be used to generate signature details. On the other hand, the scaling function represents the low-frequency domain information inherited in the original signature vector to retain majority of the signature vector; thus, it can be used to produce the signature approximation. By means of these two details and approximation signatures, WSCA can perform self-tuning and self-correction to characterize a hyperspectral signature vector for signature discrimination, classification, and identification.

29.1 Introduction

Wavelet analysis has been well studied and used successfully in many areas in signal and image processing (Daubechies, 1992; Akansu and Haddad, 1992; Vetterli and Kovacevic, 1995; Strang and Nguyen, 1996; Mallat, 1999). This chapter explores a new application of wavelet analysis in hyperspectral signature characterization by taking advantage of its multiple-scale (multiscale) representation. Two specific functions resulting from wavelet analysis, called "wavelet function" and "scaling function," are used to decompose a hyperspectral signature vector in multiscale representation for characterization. In other words, an original hyperspectral signature vector can be decomposed into detail signature component and approximation signature component via wavelet function and scaling function whose shifted and scaled versions span two orthogonal vector spaces. The wavelet function is effective in capturing details of a signature that

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

correspond to the high-frequency domain information of the original signature, while the scaling function preserves the low-frequency domain information inherited in the original signature. Generally, such wavelet transform can be discrete or continuous depending upon the scalars used to translate and also dilate both the wavelet and scaling functions. In this chapter, only discrete orthonormal bases of wavelets are of interest and a specific algorithm, called Mallat's algorithm considered in Mallat (1989), is used to construct a high-pass filter and a low-pass filter. The pair of such high-pass and low-pass filters is then used to perform DWT for hyperspectral signature characterization such as self-adjustment, similarity, discrimination, classification, and so on.

This chapter marries two seemingly different approaches together, wavelets and Kalman filters developed in Chapter 28, by taking advantage of the concept of an innovations process used in the Kalman filtering to develop WSCA that can be used for various applications. With the introduction of such an innovations process, a hyperspectral signature vector can be decomposed into two components, detail signature and approximation signature, to perform self-adjustment in an iterative manner. When a reference signature is available, WSCA can be used to perform signature self-tuning to the reference signature, referred to as WSCA signature self-tuning (WSCA-SST). If a hyperspectral signature vector is contaminated or even mistaken as some other similar signature, WSCA can be used for signature self-correction, referred to as WSCA for signature self-correction (WSCA-SSC). Such a WSCA-based approach can also be extended to the applications of hyperspectral signature self-discrimination, self-classification, and self-identification provided that a reference signature vector is given.

29.2 Wavelet Analysis

Wavelet analysis is a widely used technique in signal processing and communications, where its applications range from one-dimensional (1D) signal processing, such as speech, sonar, and audio processing, to multidimensional signal processing, such as two-dimensional (2D) image processing and three-dimensional (3D) video processing. One of the major features of wavelet analysis is the use of the so-called scaling function to generate a set of wavelets that decompose signals in multiple pair-wise disjoint orthogonal representations, referred to as signal resolutions. When the signals to be considered are one dimensional, the multiple signal resolutions are referred in this chapter to as multiple signal scales. With this interpretation, the resulting multiple pair-wise disjoint orthogonal representations are then called multiscale signal representation. On the other hand, if the signals to be considered are 2D or 3D images, the multiple signal resolutions are referred to as image resolutions and the resulting multiple pair-wise disjoint orthogonal representations are then called multiple image resolutions. Since the main focus of this chapter is 1D signal processing, the term "multiscale" will be used throughout this chapter.

29.2.1 Multiscale Approximation

The idea of multiscale approximation of wavelet analysis is briefly reviewed in this section. First, let Z and R denote the sets of integers and real numbers, respectively, and $L^2(R)$ denote the vector space of measurable, square-integrable one-dimensional functions. More specifically, assume that f (x) is a signal function and A_{2^j} is an operator that approximates the $f(x)$ at the scale of 2^j. Let V_{2^j} be the multiscale approximation vector space, which can be interpreted as the set of all possible approximations of functions in $L^2(R)$ at the scale of 2^j. Then $A_{2^j}f(x)$ is the function that is most similar to $f(x)$ among all the possible approximation functions in V_{2^j} at the scale of 2^j. For more details about A_{2^j}, we refer to Mallat (1989).

29.2.2 Scaling Function

Let $(V_{2^j})_{j \in Z}$ represent a sequence of multiscale approximations of functions in $L^2(R)$. Then there exists a unique function $\phi(x) \in L^2(R)$, called a scaling function, that satisfies the following property:

If we define $\phi_{2^j}(x) = 2^j \phi(2^j x)$ for $j \in Z$ (the dilation of $\phi(x)$ by 2^j), then $\left(\sqrt{2^j}\phi_{2^j}\left(x - 2^{-j}n\right)\right)_{n \in Z}$ is an orthonormal basis of $(V_{2^j})_{j \in Z}$.

Using the scaling function defined above, any randomly chosen function $f(x) \in L^2(R)$ can be approximated by

$$A_{2^j} f(x) = 2^{-j} \sum_{n=-\infty}^{\infty} \left\langle f(u), \phi_{2^j}(u - 2^{-j}n) \right\rangle \cdot \phi_{2^j}(x - 2^{-j}n) \tag{29.1}$$

where

$$\left\langle f(u), \varphi_{2^j}(u - 2^{-j}n) \right\rangle = \int_{-\infty}^{+\infty} f(u) \varphi_{2^j}(u - 2^{-j}n) du \tag{29.2}$$

Since (29.1) is a continuous approximation, its discrete approximation can be derived from (29.2) as the following inner product:

$$A_{2^j}^d f(x) = \left(\left\langle f(u), \phi_{2^j}(u - 2^{-j}n) \right\rangle \right)_{n \in Z} \tag{29.3}$$

where $A_{2^j}^d f(x)$ is called a discrete approximation of $f(x)$ at the scale of 2^j. So far, we have introduced the continuous and discrete approximations of a function only at a given multiscale level of 2^j. In practice, only a signal at a finite scale is considered for measurement. Without loss of generality, it often assumes that the finite scale for measurement is normalized to 1. Therefore, let $A_1^d f(x)$ be the discrete approximation at the scale 1. In order to obtain the multiscale transform in an iterative manner, the following equation allows us to compute all the discrete approximations $A_{2^j}^d f(x)$ for $j < 0$ from $A_1^d f(x)$:

$$\begin{aligned} &\left\langle f(u), \phi_{2^j}(u - 2^j n) \right\rangle \\ &= \sum_{k=-\infty}^{\infty} \langle \phi_{2^{-1}}(u), \phi(u - (k - 2n)) \rangle \left\langle f(u), \phi_{2^{j+1}}(u - 2^{-j-1}k) \right\rangle \end{aligned} \tag{29.4}$$

Furthermore, a low-pass filter can be defined to simplify (29.4). Let H be a discrete filter whose impulse response is given by

$$\forall n \in \mathbf{Z}, \quad h(n) = \langle \phi_{2^{-1}}(u), \phi(u - n) \rangle \tag{29.5}$$

and also let $\tilde{H}$ be a mirror filter with the impulse response given by

$$\tilde{h}(n) = h(-n) \tag{29.6}$$

such that (29.4) can be simplified as follows:

$$\left\langle f(u), \phi_{2^j}(u - 2^j n) \right\rangle = \sum_{k=-\infty}^{\infty} \tilde{h}(2n - k) \left\langle f(u), \phi_{2^{j+1}}(u - 2^{-j-1}k) \right\rangle \tag{29.7}$$

29.2.3 Wavelet Function

In Section 29.2.1, a signal can be discretely approximated by $A^d_{2^j}f(x)$ at the signal scale of 2^j via (29.3). The resulting approximation error is the signal loss resulting from a signal representation of $f(x)$ at the scale of 2^j, denoted by $f_{2^j}(x)$. Such detail signal $f_{2^j}(x)$ can be characterized by the difference between the approximations of a signal $f(x)$ at two consecutive scales 2^{j+1} and 2^j, where $f_{2^j}(x)$ is considered as the approximation of $f(x)$ at the scale of 2^j and can be obtained by orthogonal projections of the original signal $f(x)$ onto the orthogonal complement of V_{2^j}, denoted by O_{2^j}, which is the orthogonal complement vector space of V_{2^j} in the space $V_{2^{j+1}}$. Similarly, the approximation of $f(x)$ at the scale of 2^{j+1}, $f_{2^{j+1}}(x)$, can also be obtained by mapping the original signal $f(x)$ into $O_{2^{j+1}}$.

Let $(V_{2^j})_{j\in Z}$ be a sequence of multiscale vector spaces and $\phi(x)$ be the scaling function. Also, let Ψ(x) denote a function whose Fourier transform is given by $\boldsymbol{\psi}(w) = \boldsymbol{G}(w/2)\boldsymbol{\varphi}(w/2)$

$$\boldsymbol{G}(w) = e^{-jw}\boldsymbol{H}(w+\pi) \tag{29.8}$$

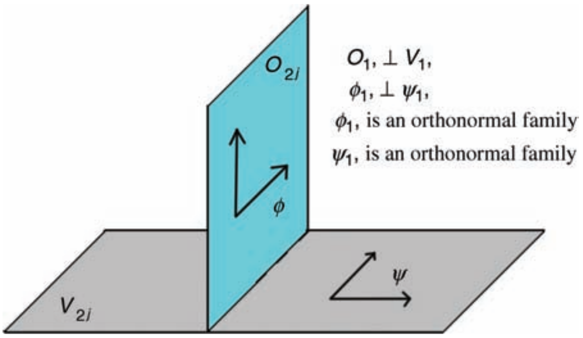

O_{2j}
$O_1, \perp V_1,$
$\phi_1, \perp \psi_1,$
ϕ_1, is an orthonormal family
ψ_1, is an orthonormal family
ϕ
ψ
V_{2j}

Figure 29.1 Geometric interpretation between V_{2^j} and O_{2^j}.

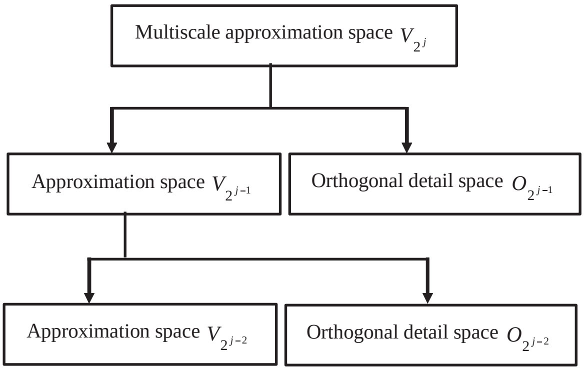

Multiscale approximation space V_{2^j}
Approximation space $V_{2^{j-1}}$
Orthogonal detail space $O_{2^{j-1}}$
Approximation space $V_{2^{j-2}}$
Orthogonal detail space $O_{2^{j-2}}$

Figure 29.2 Geometric interpretation of approximation and orthogonal detail spaces at different scales.

where $\boldsymbol{H}(w)$ and $\boldsymbol{\varphi}(w)$ are the Fourier transforms of the discrete low-pass filter H defined by (29.5) and scaling function $\phi(x)$, respectively. Therefore, the function Ψ(x) defined by (29.8) satisfies the following property:

$$\left(\sqrt{2^j}\psi_{2^j}(x - 2^{-j}n)\right)_{n \in Z} \text{ is an orthonormal basis of } O_{2^j}$$

By the wavelet function, it is easy to find the orthogonal projection of $f(x)$ and its corresponding discrete version in the vector space O_{2^j} specified by

$$P_{O^{2^j}} f(x) = 2^{-j} \sum_{k=-\infty}^{\infty} \left\langle f(u), \psi_{2^j}(u - 2^{-j}n) \right\rangle \psi_{2^j}(x - 2^{-j}n) \tag{29.9}$$

which yields the detail signal representation of $f(x)$ at the scale of 2^j. It is characterized by the set of inner products

$$D^d_{2^j} f(x) = \left(\left\langle f(u), \psi_{2^j}(u - 2^{-j}n) \right\rangle \right)_{n \in Z} \tag{29.10}$$

$D^d_{2^j} f(x)$ is called the discrete detail signal of $f(x)$ at the scale of 2^j, which contains the information difference between $A^d_{2^j} f(x)$ and $A^d_{2^{j+1}} f(x)$. Similarly, the multiscale wavelet transform can also be computed iteratively and simplified by defining a high-pass and mirror filter. Let G be a discrete filter with the impulse response given by

$$g(n) = \langle \psi_{2^{-1}}(u), \phi(u - n) \rangle \tag{29.11}$$

and $\tilde{G}$ be the corresponding mirror filter with the impulse response given by

$$\tilde{g}(n) = g(-n) \tag{29.12}$$

Following the same deviation in (29.4), we obtain

$$\left\langle f(u), \psi_{2^j}(u - 2^j n) \right\rangle = \sum_{k=-\infty}^{\infty} \tilde{g}(2n - k) \cdot \left\langle f(u), \phi_{2^{j+1}}(u - 2^{-j-1}k) \right\rangle \tag{29.13}$$

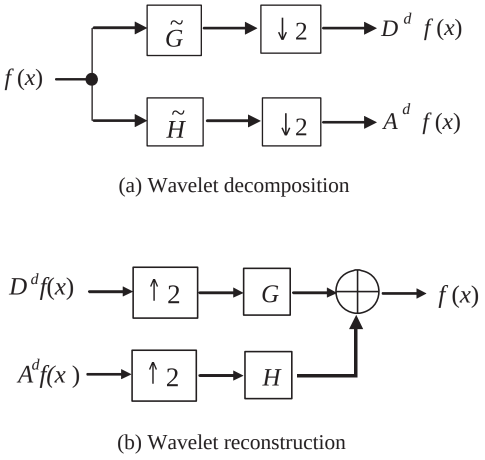
Figure 29.1 delineates the geometric interpretation of the relationships between V_{2^l} and O_{2^j} with a detailed block diagram shown in Figure 29.2.

29.3 Wavelet-Based Signature Characterization Algorithm

In this section, WSCA is developed for hyperspectral signal processing, which can perform signature self-tuning (SST) and signature self-correction (SSC), referred to as WSCA-SST and WSCA-SSC, respectively.

29.3.1 Wavelet-Based Signature Characterization Algorithm for Signature Self-Tuning

The SST ability of WSCA arises from the fact that a signature can be wavelet-decomposed into two orthogonal signature components, called details signature and approximation signature, an idea borrowed from the innovations process used in the Kalman filtering. Mallet's algorithm (1989) demonstrated that the wavelet decomposition and reconstruction of a signal $f(x)$ could be related to each other by the two flow diagrams depicted in Figure 29.3.



(a) Wavelet decomposition

(b) Wavelet reconstruction

Figure 29.3 Flow diagrams for both wavelet decomposition and reconstruction.

The signal function $f(x)$ represents the original hyperspectral signature, which will be later used as a reference signature. The two signature components of $f(x)$, $D^d f(x)$ and $A^d f(x)$, are referred to as its details and approximation signatures, respectively. The G and H in Figure 29.3 denote the high-pass and low-pass filters derived from the wavelet function and scaling function, respectively, as described in Section 29.2, where $\tilde{G}$ and $\tilde{H}$ are the corresponding mirror filters of G and H. Assume that the detail signature of the original signature $f(x)$ is $D^d f(x)$. Let its noise-corrupted signature be denoted by $D^{\text{corrupt}} f(x)$. The $D^d f(x)$ can then be self-tuned from $D^{\text{corrupt}} f(x)$ through a feedback and an iterative convergent process as described below.

Let the original signature $f(x)$ be decomposed into two components:

$$f(x) = f(x) - \hat{f}(x) + \hat{f}(x) \tag{29.14}$$

where $\hat{f}(x)$ is a predicted signature produced by the flow diagram and the corrupted signature $D^{\text{corrupt}} f(x)$ serves as an input signature, and the approximation signature $A^d f(x)$ remains clean and is exactly the same as depicted in Figure 29.3. After the resulting predicted signature, denoted by $\hat{f}(x)$, is generated as the output of the diagram depicted in Figure 29.4, an error signature, denoted by $\varepsilon(x)$, can be defined by $\varepsilon(x) = f(x) - \hat{f}(x)$. According to Kalman filtering, the error signature $\varepsilon(x)$ is also referred to as innovations signature that contains new information that cannot be predicted from the signature $f(x)$. Such an innovations signature can be used to update $\hat{f}(x)$ in the same way as is carried out by a Kalman filter. Using wavelet analysis, the error signature $\varepsilon(x)$ can be further decomposed by downsampling into two components $D^{\text{innovations}} f(x)$ and $A^{\text{innovations}} f(x)$ as shown in Figure 29.5.

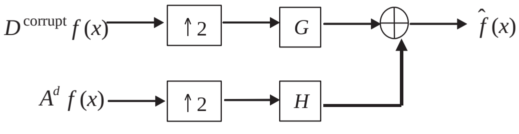


Figure 29.4 Flow diagram for generating a prediction signature of the original signature $f(x)$.

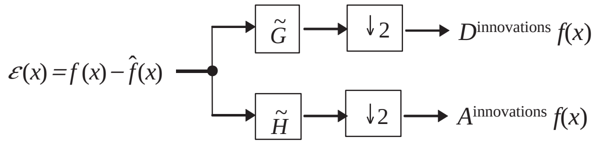

$\varepsilon(x) = f(x) - \hat{f}(x)$
$\tilde{G}$
$\downarrow 2$
$D^{\text{innovations}} f(x)$
$\tilde{H}$
$\downarrow 2$
$A^{\text{innovations}} f(x)$

Figure 29.5 Wavelet decomposition of error signature ε(x) by down sampling.

Using the innovations details signature $D^{\text{innovations}}f(x)$ in Figure 29.5, we can further define a new detail signature $D^{\text{new}}f\,(\text{x})$ by

$$D^{\text{new}}f(x) = D^{\text{innovations}}f(x) + D^{\text{corrupted}}f(x) \tag{29.15}$$

The $D^{\text{new}}f(x)$ obtained by (29.15) is then used to replace the corrupted detail signature $D^{\text{corrupt}}f(x)$ to produce a new reconstructed signature $\hat{f}^{\text{new}}(x)$ by up sampling $D^{\text{new}}f(x)$ and $A^d f(x)$ as shown in Figure 29.6.

Accordingly, the new detail signature $D^{\text{new}}f(x)$ is more accurate than the corrupted signature $D^{\text{corrupted}}f(x)$ so that the signature reconstructed from the $D^{\text{new}}f(x)$ is more similar than that reconstructed from the $D^{\text{corrupt}}f(x)$ in terms of least squares error (LSE). The same procedure is continued until the difference between two consecutive LSEs is within a prescribed threshold ε. The above procedure is called WSCA-SST and its detailed implementation is summarized as follows.

Wavelet-based signature characterization algorithm for self-tuning (WSCA-SST)

1. Let ε be a stopping threshold and the initial value of LSE denoted by LSE_{old} be zero.
2. Reconstruct the signature $\hat{f}(x)$ through the $D^{\text{corrupt}}f(x)$.
3. Calculate the LSE_{new} between $\hat{f}(x)$ and $f(x)$.
4. If $(\text{LSE}_{\text{new}} - \text{LSE}_{\text{old}}) < \varepsilon$, go to step 5. Otherwise, continue the following procedure:
    a. Apply wavelet decomposition with the innovation signature $\varepsilon(x) = f(x) - \hat{f}(x)$ as the input to find the innovations details signature $D^{\text{innovations}}f(x)$.
    b. Sum the resulting $D^{\text{innovations}}f(x)$ and $D^{\text{corrupt}}f(x)$ to form a new detail $D^{\text{new}}f(x)$ according to (29.15).
    c. Replace the old $D^{\text{corrupt}}f(x)$ with new generated detail $D^{\text{new}}f(x)$ and go to step 2.
5. Output $\hat{f}(x)$ as the prediction of $f(x)$, and also output $D^{\text{new}}f(x)$ as the self-tuned version of $D^{\text{corrupt}}f(x)$.

For an illustrative purpose, Figure 29.7 depicts a flowchart that implements the five steps described in the above WSCA-SST.

The above WSCA-SST is presented by taking the corrupted DWT details signature as an example to show how it works. On the other hand, it can also be easily extended to the case of the corrupted DWT approximation signature by simply replacing all the details signatures in WSCA-SST with the approximation signatures. The only difference lies in the fact that the required

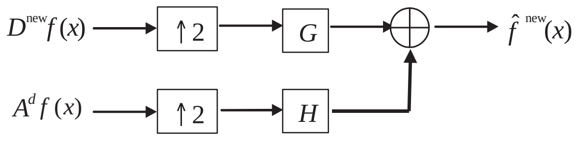


Figure 29.6 Reconstruction of $f(x)$ using $D^{\text{new}}f(x)$ and $A^d f(x)$.

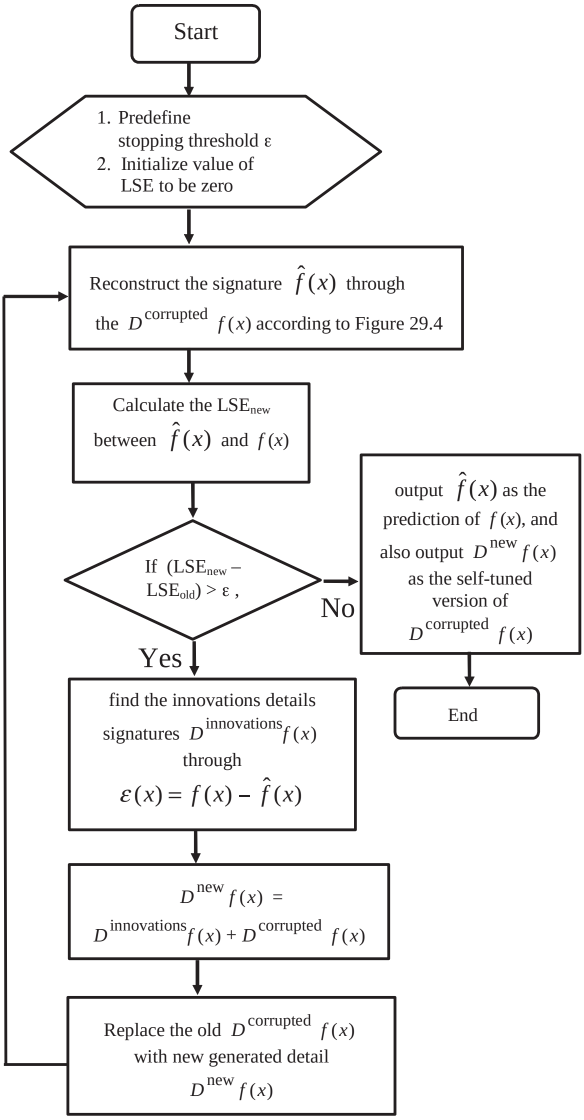

Start
1. Predefine stopping threshold ε
2. Initialize value of LSE to be zero
Reconstruct the signature $\hat{f}(x)$ through the $D^{\text{corrupted}} f(x)$ according to Figure 29.4
Calculate the LSE$_{\text{new}}$ between $\hat{f}(x)$ and $f(x)$
If (LSE$_{\text{new}}$ – LSE$_{\text{old}}$) > ε ,
No
Yes
output $\hat{f}(x)$ as the prediction of $f(x)$, and also output $D^{\text{new}} f(x)$ as the self-tuned version of $D^{\text{corrupted}} f(x)$
End
find the innovations details signatures $D^{\text{innovations}} f(x)$ through $\varepsilon(x) = f(x) - \hat{f}(x)$
$D^{\text{new}} f(x) = D^{\text{innovations}} f(x) + D^{\text{corrupted}} f(x)$
Replace the old $D^{\text{corrupted}} f(x)$ with new generated detail $D^{\text{new}} f(x)$

Figure 29.7 Flowchart for WSCA-SST.

number of iterations for corrupted approximation signatures is greater than that required for corrupted detail signatures because the approximation signature contains much more information about the original hyperspectral signature vector than does the detail signature.

29.3.2 Wavelet-Based Signature Characterization Algorithm for Signature Self-Correction

In the case of WSCA-SST, either details signature or approximation signature is assumed to be corrupted after DWT. What would it be if the original hyperspectral signature are corrupted by random noise, in which case both the approximation and details signatures are corrupted by noise? Or even in the worst case where the original hyperspectral signature is mistaken as another similar signature with both the approximation and details signatures corrupted by some other signature? Following the same innovations approach described above, we can generate two innovations signatures: one is the innovations approximation signature denoted by $A^{\text{innovations}}f(x)$ and the other is the innovations details signature denoted by $D^{\text{innovations}}f(x)$. Using these two innovations signatures, three methods for updating the approximation signature $A^{\text{corrupted}}f(x)$ and detail signature $D^{\text{corrupted}}f(x)$ can be derived as follows:

1. Only update the old approximation signature

$$A^{\text{new}}f(x) = A^{\text{innovations}}f(x) + A^{\text{corrupted}}f(x) \tag{29.16}$$

2. Only update the old details signature according to (29.15).
3. Update both the old details signature and old approximation signature according to (29.15) and (29.16).

These three update methods produce three different LSEs. An optimal strategy is to choose an update method that yields the minimum LSE, and then go back to the next feedback and iteration step. This procedure is called WSCA-SSC and summarized as follows.

Wavelet-based signature characterization algorithm for signature self-correction (WSCA-SSC)

1. Let ε be a stopping threshold and the initial value of LSE denoted by LSE_{old} be zero.
2. Reconstruct the signature $\hat{f}(x)$ through the $D^{\text{corrupted}}f(x)$ and $A^{\text{corrupted}}f(x)$.
3. Calculate the LSE_{new} between $\hat{f}(x)$ and f (x).
4. If $(\text{LSE}_{\text{new}} - \text{LSE}_{\text{old}}) < \varepsilon$, go to step 5. Otherwise, continue the following procedure:
   a. Apply wavelet decomposition with the innovation signature $\varepsilon(x) = f(x) - \hat{f}(x)$ as the input to find the innovations details signature $D^{\text{innovations}}f(x)$ and the innovation approximation signature $A^{\text{innovations}}f(x)$.
   b. Regenerate three new LSEs according to (29.15), (29.16), or both, respectively, to decide the updating rule, that is, only updating the old details signature or approximation signature, or both. The strategy is to choose a method that yields the minimum new LSE.
   c. Update the details signature, approximation signature, or even both according to the updating rule generated by step (b).
   d. Replace the old $D^{\text{corrupted}}f(x)$ with new generated detail $D^{\text{new}}f(x)$, or replace the old $A^{\text{corrupted}}f(x)$ with new generated detail $A^{\text{new}}f(x)$, or both according to the updating rule generated by step (b). Then go back to step 2.
5. Output $\hat{f}(x)$ as the prediction of $f(x)$, and also output $D^{\text{new}}f(x)$ as the self-tuned version of $D^{\text{corrupted}}f(x)$ and $A^{\text{new}}f(x)$ as the self-tuned version of $A^{\text{corrupted}}f(x)$.

29.3.3 Signature Self-Discrimination, Classification, and Identification

In this section, WSCA-SSC will be extended to hyperspectral signature discrimination, classification, and identification. The idea can be described as follows. When WSCA-SSC finally converges,

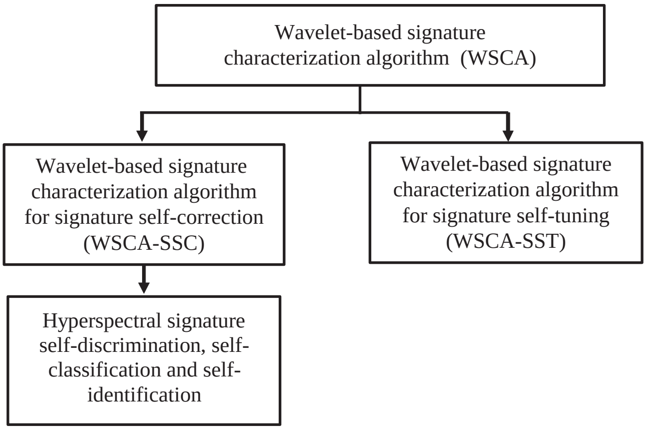

Wavelet-based signature characterization algorithm (WSCA)
Wavelet-based signature characterization algorithm for signature self-correction (WSCA-SSC)
Wavelet-based signature characterization algorithm for signature self-tuning (WSCA-SST)
Hyperspectral signature self-discrimination, self-classification and self-identification

Figure 29.8 Relationships derived from WSCA, which are WSCA-SST, WSCA-SSC, and signature self-discrimination, classification, and identification.

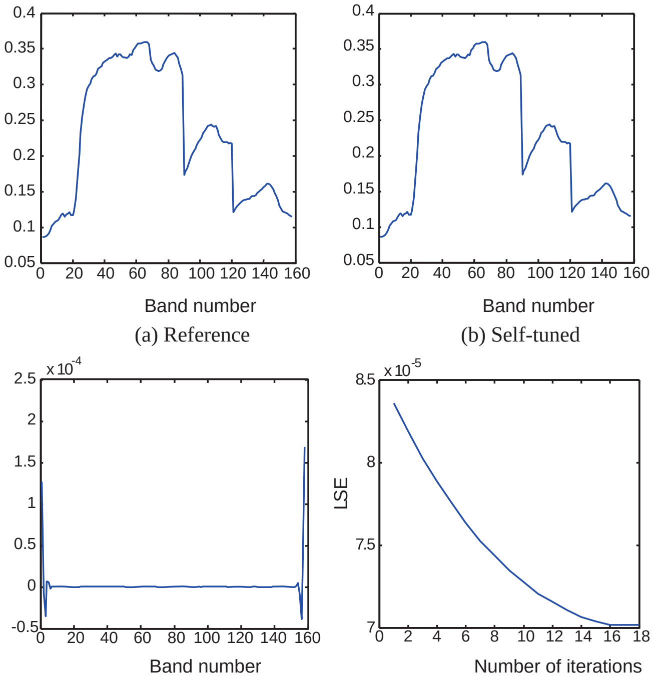
the resulting LSE can be served as a quantitative indication as well as a measure for signature identification, classification, and discrimination. For signature discrimination, the input parameters are two different signature vectors to be discriminated, rather than corrupted and reference signature vectors. The smaller the LSE, the more similar the two input signature vectors. It should be noted that since the resulting LSE is not strictly symmetric, exchanging the roles of the two input parameters might result in different LSEs. Therefore, in order for WSCA-SSC to be extended to perform as a discriminator between two different signature vectors, we define the discrimination power denoted by $\mathrm{DP}^{\text{WSCA-SSC}}$ as

$$\mathrm{DP}^{\text{WSCA-SSC}}(\mathbf{s},\mathbf{t}) = \mathrm{LSE}(\mathbf{t},\mathbf{s}) + \mathrm{LSE}(\mathbf{s},\mathbf{t}) \tag{29.17}$$

where **s** and **t** represent two different hyperspectral signature vectors to be discriminated, and LSE(**t**,**s**) denotes the resulting LSE from WSCA-SSC with **t** as the signature vector to be processed and **s** as the reference signature vector. In order for WSCA-SSC to be extended to perform signature identification and classification, similar processes will be applied using mixed signatures or subpixels as signatures to be processed. So far, the relationship between WSCA, WSCA-SST, WSCA-SSC, and also signature self-discrimination, classification, and identification can be depicted by the structure in Figure 29.8.

29.4 Synthetic Image-Based Computer Simulations

In order to demonstrate the utility of proposed WSCA, experiments using computer simulations are conducted. For computer simulations, five Airborne Visible infra Red Imaging Spectrometer (AVIRIS) reflectance data, blackbrush, creosote leaves, drygrass, red soil, and sagebrush, are used and are shown in Figure 1.8. Each of these five spectral signatures has 158 bands after water bands are removed and can be considered as a 158-dimensional hyperspectral signature where each signature component is specified by a particular spectral wavelength.



(a) Reference

(b) Self-tuned

(c) Error profile

(d) LSE

Figure 29.9 Comparison between original and corrected signatures.

29.4.1 Signature Self-Tuning and Self-Denoising

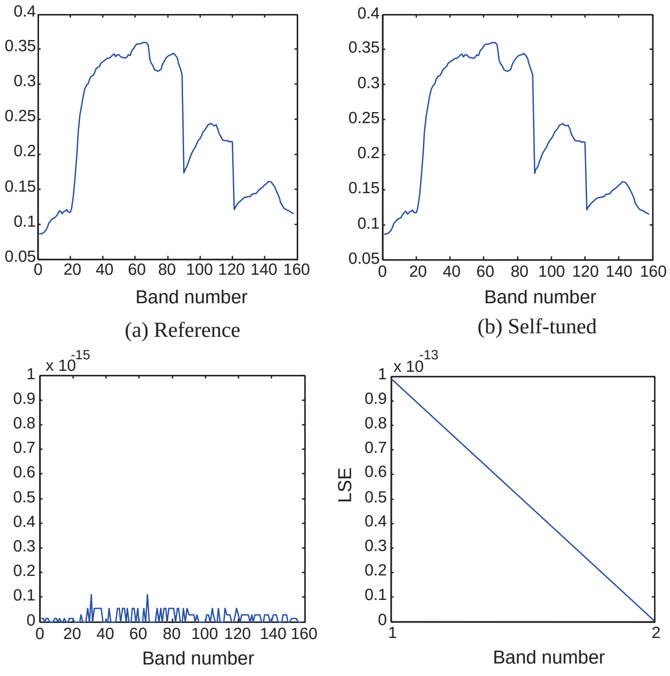
Since blackbrush, creosote leaves, and sagebrush are very close in their spectral shapes, there are two cases for us to discuss: the first case is that one signature’s details signatures are corrupted by any other signatures. The second case is that one of these three signatures are considered as a corrupted signature of the other two signatures, with both details and approximation signatures. Either case provides us with a good example to analyze WSCA in signature self-tuned performance. The experiments were divided into two categories. In the first one, the signature was self-tuned by detail signatures, and in the second one, the signature was self-denoised by both detail and approximation signatures.

Experiment 29.4.1.1 (Self-tuned by detail signature)

In this experiment, the details signature of the discrete wavelet transformed blackbrush was corrupted by the details signature of the creosote leaves. In order for the corrupted signature blackbrush to be self-tuned, the pure signature of blackbrush was used as the reference. The results along with the resulting LSE for each step of iteration are shown in Figure 29.9(a)–(c). As can be seen, the self-tuned signature was nearly the same original pure black signature.

Experiment 29.4.1.2 (Self-tuned by both detail and approximation signatures)

This experiment considered creosote leaves as a corrupted signature of blackbrush. In this case, both the detail and approximation signatures of blackbrush were replaced with those of creosote



(a) Reference

(b) Self-tuned

(c) Error profile

(d) LSE

Figure 29.10 Comparison between original and corrected signatures.

leaves. The results are shown in Figure 29.10, where the self-tuned signature resulting from WSCA matched the original pure signature with very high accuracy, which could be seen from the numerical value of resulting LSE between self-tuned and original signatures.

As for convergence rate, the results shown in Figures 29.9(d) and 29.10(d) demonstrated that with both details and approximation signatures tuned at each iteration step, the convergence speed was much faster than that obtained by only self-tuning the details signatures at each iteration step.

29.4.2 Signature Self-Discrimination, Self-Classification, and Self-Identification

In this section, both simulated mixed hyperspectral signature and subpixels were used for experiments with WSCA used as discriminator, classifier, and identifier. The discrimination power defined by (29.4) was used as a measure for discrimination, classification, and identification.

Experiment 29.4.2.1 (Signature self-discrimination)

Blackbrush, creosote leaves, and sagebrush were used here because of their close similarity. Their $\mathrm{DP}^{\mathrm{WSCA}}$ are tabulated in Table 29.1.

The results in Table 29.1 further show that both blackbrush and creosote leaves were much closer to sagebrush, which was consistent with the results of using other commonly used similarity measures such as SAM and SID.

Table 29.1 Discrimination power of WSCA

	Blackbrush	Creosote leaves	Sagebrush
Blackbrush	0	6.1629e – 013	3.2014e – 013
Creosote leaves		0	3.3804e – 013
Sagebrush			0

Table 29.2 Discrimination power of WSCA

Blackbrush	Creosote leaves	Sagebrush	Drygrass
2.6981e – 013	4.2723e – 013	1.6359e – 013	4.9961e – 013

Table 29.3 DP^{WSCA} in 10^{-13} for subpixel with different target sizes

	10%	20%	30%	40%	50%
Drygrass	0.8764	1.7519	2.6276	3.5044	4.3815
Creosote leaves	7.8858	7.0089	6.1327	5.2574	4.3810
	60%	70%	80%	90%	100%
Drygrass	5.2563	6.1337	7.0097	7.8865	0.5892
Creosote leaves	3.5051	2.6282	1.7518	8.7540	9.2024

Experiment 29.4.2.2 (Mixed pixel self-classification)

In this experiment, the mixed hyperspectral signature was simulated by equally mixing four different materials, blackbrush, creosote leaves, sagebrush, and drygrass, each with abundance of 25%. The classification results are shown in Table 29.2.

The smaller DP^{WSCA} is more similar to the two signatures. The results demonstrated that WSCA could successfully classify the equally mixed pixel as sagebrush, which can be explained by the fact that both blackbrush and creosote leaves were close to sagebrush by spectral similarity measures such as SAM and SID.

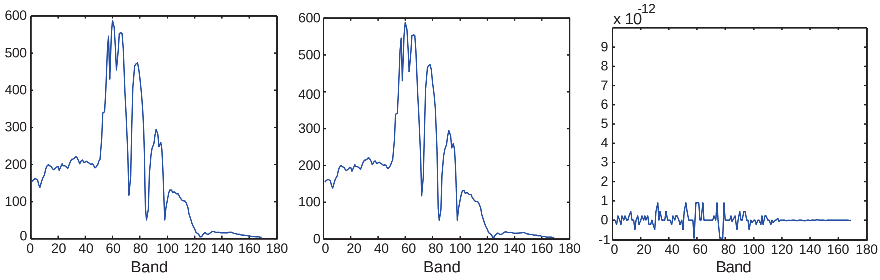
Experiment 29.4.2.3 (Subpixel self-identification)

In this experiment, the subpixels of different target sizes were simulated with creosote leave as the target, mixed with drygrass as background. For example, the subpixel of 25% target size was simulated by mixing three background pixels with one pixel of the target. The identification results are shown in Table 29.3.

The results demonstrated that WSCA could successfully identify the subpixel as target creosote leaves as long as the size of subpixel target creosote leaves was equal to or above 50%.

29.5 Real Image Experiments

So far, we have discussed much about the utility of hyperspectral signature self-clarification using wavelet discrete decomposition, which can perform self-clarification given a relatively



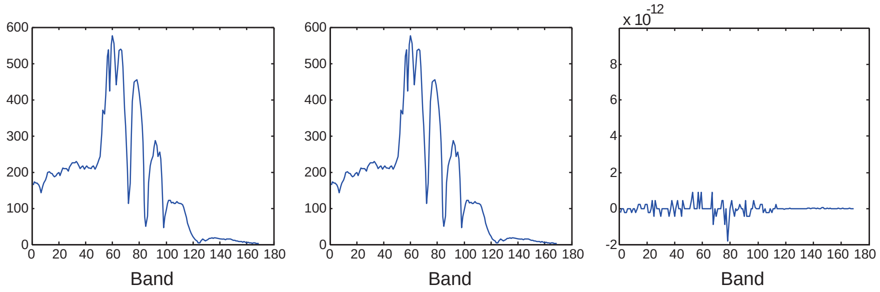
(a) $\mathbf{p}_1$ (b) Reconstructed from p_{13} (c) Error profile

Figure 29.11 Original $\mathbf{p}_1$ and self-correction version from p_{13}.

pure and clean signal as a reference. Intuitively, the farther the two hyperspectral signatures, the more the number of iterations required. So this can be used to judge if one hyperspectral signature is much closer to the another and further develop a technique to perform subpixel identification. In the following experiments, two different ways to choose reference were considered, one using the five average panel signatures, $\{\mathbf{p}_i\}_{i=1}^5$, as the reference, and the other using the average of the first two column panel signatures as the reference. Subpixel panels $\{p_{i3}\}_{i=1}^5$ were the signatures to be identified.

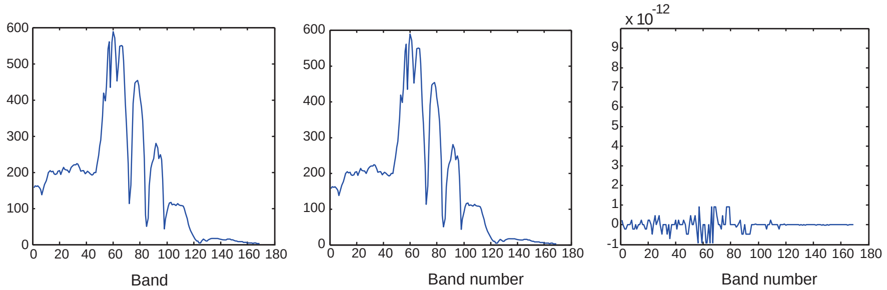
Experiment 29.5.1 (WSCA-SSC)

In this experiment, five subpixel panels p_{13}, p_{23}, p_{33}, p_{43}, and p_{53} in the last column were considered as contaminated signatures of the five panels signatures $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ in Figure 1.16 that were used as references for subpixel panels p_{13}, p_{23}, p_{33}, p_{43}, and p_{53} for self-correction. In other words, WSCA-SSC was applied here to recover the original five panels signatures $\mathbf{p}_1$, $\mathbf{p}_2$, $\mathbf{p}_3$, $\mathbf{p}_4$, and $\mathbf{p}_5$ from their corrupted versions, here considered as p_{13}, p_{23}, p$_{33}$, p_{43}, and p_{53}. Figures 29.11–29.15 show the WSCA-SSC results along with their corresponding LSEs.



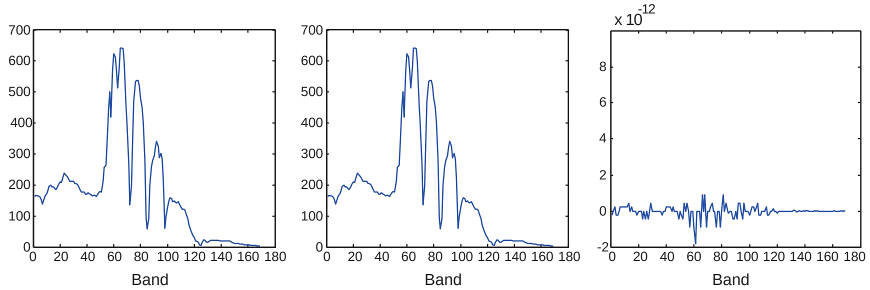
(a) $\mathbf{p}_2$ (b) Reconstructed from p_{23} (c) Error profile

Figure 29.12 Original $\mathbf{p}_2$ and self-correction version from p_{23}.



(a) $\mathbf{p}_3$ (b) Reconstructed from p_{33} (c) Error profile

Figure 29.13 Original $\mathbf{p}_3$ and self-denoised version from p_{33}.

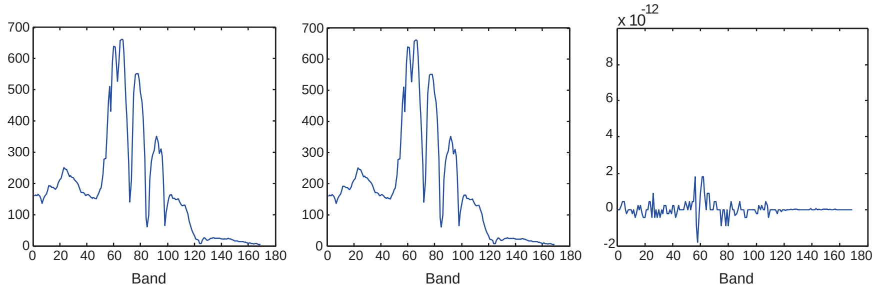


(a) $\mathbf{p}_4$ (b) Reconstructed from p_{43} (c) Error profile

Figure 29.14 Original $\mathbf{p}_4$ and self-correction version from p_{43}.

Experiment 29.5.2 (Subpixel self-identification using WSCA-SSC)

Since the panels p_{13}, p_{23}, p_{33}, p_{43}, and p_{53} in the last column have same size of 1 m × 1 m that is smaller than the 1.56 m × 1.56 m spatial resolution, they are actually subpixel panels and provide a good example with their corresponding spectral signature vectors considered to be discriminated by WSCA-SSC with the reference signature vectors chosen to be the five panel signatures in Figure 1.16. Their $\text{DP}^{\text{WSCA-SSC}}$ are tabulated in Table 29.4, where a cross indicates an incorrect identification.



(a) $\mathbf{p}_5$ (b) Reconstructed from p_{53} (c) Error profile

Figure 29.15 Original $\mathbf{p}_5$ and self-correction version from p_{53}.

Table 29.4 $\mathrm{DP}^{\mathrm{WSCA\text{-}SSC}}$ in 10^{-9} for subpixel panel identification

	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
p_{11}	2.3409	3.3172	3.9544	3.4417	3.5440
p_{12}	0.3383	1.3113	2.6041	4.1124	4.5321
p_{13}	2.2475	2.0926	3.1510	5.6057	6.2415
p_{211}	1.6062	0.7910	1.8471	4.3584	4.5801
p_{221}	1.4333	0.9931	2.0134	4.1615	4.4090
p_{22}	1.5873	0.3540	1.9674	4.8817	5.2222
p_{23}	2.3370	1.5089	2.6381	5.6607	6.1351
p_{311}	3.0492	2.5057	1.2020	5.4509	5.5516
p_{312}	3.2021	2.8915	1.6988	5.1001	5.1458
p_{32}	2.8146	2.0061	0.8634	5.1692	5.4864
p_{33}	3.4005	2.3932	2.2421	6.2933	6.6947
p_{411}	6.0148	6.7728	7.2537	2.3320	2.2207
p_{412}	5.8439	6.5981	7.2011	2.0997	2.1190
p_{42}	4.5378	5.1850	5.6902	0.62061	1.0919
p_{43}	2.7346	2.1622	2.8088	4.7516	5.3035
p_{511}	4.7269	5.2888	5.8289	1.3884	0.4717
p_{512}	7.9715	8.6424	9.0293	4.4454	3.7631
p_{52}	5.8336	6.4115	6.7650	2.3867	1.4972
p_{53}	2.6549	1.9019	2.6283	4.9607	5.4265

Table 29.5 SAM values for subpixel panel identification

	$\mathbf{p}_1$	$\mathbf{p}_2$	$\mathbf{p}_3$	$\mathbf{p}_4$	$\mathbf{p}_5$
p_{11}	0.0531	0.0873	0.0979	0.1008	0.1050
p_{12}	0.0079	0.0411	0.0658	0.1160	0.1258
p_{13}	0.0579	0.0453	0.0740	0.1516	0.1640
p_{211}	0.0450	0.0160	0.0387	0.1495	0.1570
p_{221}	0.0431	0.0151	0.0399	0.1462	0.1539
p_{22}	0.0365	0.0163	0.0462	0.1375	0.1461
p_{23}	0.0660	0.0338	0.0647	0.1643	0.1752
p_{311}	0.0810	0.0818	0.0540	0.1561	0.1571
p_{312}	0.0863	0.0876	0.0578	0.1598	0.1607
p_{32}	0.0849	0.0477	0.0469	0.1827	0.1915
p_{33}	0.1066	0.0684	0.0768	0.2028	0.2131
p_{411}	0.1538	0.1897	0.2036	0.0464	0.0408
p_{412}	0.1772	0.2127	0.2252	0.0709	0.0627
p_{42}	0.1174	0.1505	0.1676	0.0100	0.0252
p_{43}	0.0777	0.0646	0.0963	0.1429	0.1568
p_{511}	0.1387	0.1717	0.1852	0.0369	0.0177
p_{521}	0.1947	0.2294	0.2397	0.0923	0.0760
p_{52}	0.1526	0.1863	0.1978	0.0509	0.0326
p_{53}	0.0748	0.0568	0.0889	0.1470	0.1597

As can be seen from Table 29.4, WSCA-SSC makes incorrect identification on four panel pixels p_{13}, p_{411}, p_{43}, and p_{53}. In order to evaulate the performance of WSCA-SSC, SAM was used for comparison. Table 29.5 tabulates its identification results where same six incorrect-made identifications were panel pixels $p_{13}, p_{33}, p_{411}, p_{412}, p_{43}$, and p_{53}. Similar experiments were also performed by SID that also made incorrect identification on the same 6 panel pixels as did SAM. Therefore, SID results are not included in this chapter. Comparing Tables 29.4 and 29.5, both WSCA-SSC and SAM had trouble with identifying some subpixel panels p_{13}, p_{43}, and p_{53} in the third column because their size is $1/(1.56\,\mathrm{m})^2 = 0.41091\,\mathrm{m}^2$, which is smaller than half the size of a pixel. Nevertheless, WSCA-SSC still correctly identified two subpixel panels p_{23} and p_{33} compared to only one p_{23} identified by SAM. In addition, WSCA-SSC made only four incorrect identifications as opposed to six incorrect identifications made by SAM with two more errors on panel pixels p_{33} and p_{412}.

These real image experiments demonstrated that WSCA outperformed SAM in real subpixel identification. Such an advantage may result from the ability of WSCA-SSC that SAM does not have in capturing detailed profile of spectral variations in a hyperspectral signature vector across its spectral range as demonstrated in Experiment 29.6.

29.6 Conclusions

This chapter presents a new application of wavelet in hyperspectral signature characterization. In particular, a WSCA is developed for signature self-correction, self-tuning, self-denoising, self-discrimination, self-classification, and self-identification. The experimental results demonstrated that the proposed algorithm WSCA performs effectively in self-correction, self-tuning and self-denoising with high accuracy for a given reference. Most importantly, the results also show that WSCA can successfully self-classify and also self-identify mixed hyperspectral signatures and subpixel targets.

VIII Applications

As a final part to conclude this book a few applications are of particular interest to form Category C. Chapter 30 includes two applications of hyperspectral target detection, subpixel target size estimation and concealed target detection, both of which have unique issues to be addressed and cannot be resolved by spatial domain-based techniques due to the fact that the targets of interest are invisible either completely or partially to human eyes by inspection. Under such circumstances detection of these targets must rely on their spectral characteristics rather than on their spatial properties. These applications provide good examples to demonstrate unique advantages of hyperspectral imaging over traditional image processing. On the other hand, as noted in Chapter 1 (Sections 1.2 and 1.3), hyperspectral imagery shall not be considered as a straightforward extension to multispectral imagery by simply adding more spectral bands. As a matter of fact, it is spectral resolution, not total number of spectral bands, that matters. Consequently, techniques developed for hyperspectral imaging are not necessarily applicable to multispectral imaging, specifically linear spectral mixture analysis (LSMA). Chapter 31 presents two nonlinear band dimensionality expansion techniques, band expansion process (BEP), and kernel trick for kernelization, to make hyperspectral imaging techniques also effective on multispectral imagery. Since hyperspectral imaging techniques generally require a sufficient number of spectral bands to capture spectral information for effective data processing the lack of spectral bands in multispectral imagery certainly impairs their processing ability. BEP allows users to expand the dimensionality of an original multispectral image data by including new extra spectral bands created by nonlinear function to be treated as a hyperspectral image so that hyperspectral imaging techniques can still work effectively. On the other hand, due to the use of insufficient spectral bands the data sample vectors may not be effectively unmixed linearly. Instead of expanding band dimensionality suggested by BEP, the kernel trick allows users to perform kernelization on data transformation or on training samples or on classifiers to make nonlinear decisions in a kernel-transformed feature space. It turns out that these approaches do accomplish what they are designed for. Considering MR images as multispectral images is not new. In fact, Vannier et al. (1985) showed that satellite remote sensing image processing could be readily applied to solving MR classification problems if MR images could be processed as multispectral images. One of most interesting applications of hyperspectral imaging is to take advantage of techniques developed in Chapter 31 to solve magnetic resonance (MR) imaging problems. Chapter 32 is included to show how the idea of applying hyperspectral imaging can be applied to MR imaging. In particular, a long-standing partial volume effect issue in MR

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

image classification can be addressed by formatting a partial volume estimation problem as a linear spectral unmixing problem where linear spectral mixture analysis (LSMA) via nonlinear dimensionality expansion developed in Chapter 31 can be used to perform spectral unmixing with unmixed abundance fractions interpreted as partial volumes of tissue substances. In addition, to further conduct quantitative study on MR imaging 3D ROC analysis developed in Chapter 3 is also used for performance evaluation. It is believed that this chapter provides a new direction referred to as quantitative MR imaging as an alternative, which is quite different from traditional spatial domain-based structural approaches such as fuzzy c-means methods as well as statistical approaches such as finite Gaussian mixture model coupled with the Markov random field.

30

Applications of Target Detection

Since there is no such a uniformly optimal technique that works for all applications, when it comes to hyperspectral data exploitation users should ask themselves a simple question, "what are the applications in which they are interested?" and "what are the benefits and advantages that a hyperspectral imaging sensor can provide for a specific application?" In fact, an application generally determines what techniques should be used, but not the other around. This chapter presents two interesting applications as examples, which require specific techniques to accomplish what they are designed for. One is size estimation of a subpixel target embedded in a single pixel vector due to its size smaller than pixel resolution determined by ground sampling distance (GSD). Under such circumstances, the target can only be detected spectrally at subpixel level, not spatially as ordinarily conducted by classical spatial domain-based image processing techniques. Besides, subpixel detection generally does not estimate the subpixel target size. In other words, a subpixel target detector may not be able to estimate the size of the subpixel targets that it detects. Therefore, new approaches must be sought for subpixel size estimation. Another interesting application is concealed target detection where targets to be detected are concealed by either natural background variations or hidden underneath man-made objects. As a result, many algorithms developed for detecting exposed targets may not be directly applicable to detection of concealed targets. Most importantly, the prior knowledge of the concealed targets is generally not available and detecting concealed targets must be performed in an unsupervised manner due to nature of target concealment. Apparently, these two applications present great challenges in algorithm design because they are not encountered in traditional image processing and spatial domain-based techniques currently being used in the literature of classical image processing may not be applicable.

30.1 Introduction

One advantage provided by hyperspectral imaging is subpixel detection, which detects targets at subpixel scale. In many applications such as reconnaissance and surveillance, targets of interest may occur with low probabilities or may have relatively small size. The targets of this type are special species in agriculture and ecology, rare minerals in geology, vehicles in a large battlefield, etc. Under these circumstances, spatial-based image processing techniques may not be effective to extract these targets, particularly, when the size of targets is smaller than pixel resolution (i.e., (GSD). These targets are embedded in a single pixel vector and referred to as subpixel targets. In this case, spatial analysis-based techniques are unlikely to find these subpixel targets. We must rely on techniques that make use of their spectral characteristics to detect their presence at subpixel

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

scale. One such technique is constrained energy minimization (CEM) developed in Chapters 2 and 12, and also in Chang (2003a). In this chapter, we investigate an interesting issue associated with subpixel detection. If a subpixel target is detected within a single pixel vector, what is its size? In order to solve this problem, we develop an approach that enables us to reliably estimate the abundance fraction of a subpixel target. Then we can multiply the obtained abundance fraction by GSD to calculate its size. Such an approach is effective only if the true abundance fraction of a subpixel target contained in a pixel vector is estimated accurately. A fully constrained least squares (FCLS) method recently developed by Heinz and Chang (2001) for material quantification can be used for this purpose where FCLS imposes two constraints, abundance sum-to-one constraint (ASC) and abundance non-negativity constraint (ANC) on abundance fractions of image endmembers used to form a linear mixture model. In order to make FCLS work in an unknown environment, automatic target generation process (ATGP) similar to ATGP-EEA developed in Section 8.5.1 is further incorporated into FCLS to make it unsupervised, referred to as ATGP-FCLS, which allows FCLS to have ability in detecting subpixel targets without prior knowledge. This algorithm is a little bit different from unsupervised FCLS (UFCLS) developed in Section 8.5.3 for endmember extraction, but is proved to be very effective in estimating subpixel target size.

Over the past years various techniques have been developed for detecting and classifying exposed targets in hyperspectral imagery, specifically, Hyperspectral Digital Imagery Collection Experiment (HYDICE) imagery. However, in many applications, targets of interest are those concealed by natural background variations or hidden underneath man-made objects. In this case, algorithms developed for exposed targets may not be directly applicable to concealed target detection. The second part of this chapter investigates this issue and further develops an algorithm, to be called computer-aided detection and classification algorithm (CADCA) for unknown concealed targets, which can be carried out in three successive processes: (1) a band selection procedure; (2) a band ratio approach; and (3) ATGP. The idea of CADCA takes advantage of a band ratio transformation that can reduce effects caused by slopes and aspects of topography or terrain as well as eliminate differential illumination effects caused by shadows. However, such a band ratio approach requires a careful selection of bands to be used for the band ratio. This is more difficult for hyperspectral imagery than multispectral imagery since there are hundreds of bands from which we can select. Therefore, the first-stage process of CADCA is to design an effective band selection procedure that can select an appropriate band subset to be used for the band ratio in the second stage. The criterion particularly designed for such a band selection is based on the orthogonal projection correlation between two bands. If two bands have very little orthogonal projection correlation, this implies that the projections of one band onto the other should be very close. In this case, one band can represent the other band. So, eliminating one band will not sacrifice too much information provided by the other band in terms of orthogonal projection (OP). This criterion is further used to design ATGP for automatic target detection. The consistency of using the same OP criterion maximizes the overall performance of CADCA. On the whole, CADCA is a computer automated algorithm that can be implemented sequentially by a band selection procedure followed by a band ratio transformation and finally complete by ATGP, which detects and classifies concealed targets. As shown by experimental results, concealed targets can be effectively detected and classified by CADCA.

30.2 Size Estimation of Subpixel Targets

Prior to size estimation detection must be performed to find targets of interest. When an image scene is unknown such a detection must be carried out in an unsupervised manner. The algorithm to be used for this purpose is ATGP which is similar to ATGP-EEA developed in Section 8.5.1, and can be described as follows.

Algorithm for Automatic Target Generation Process

1. Initial condition:
   Select a pixel with maximal vector length as an initial target signature of interest denoted by $\mathbf{t}_0$. Let ε be the prescribed error threshold and $\mathbf{U}^{(0)} = [\mathbf{t}^{(0)}]$.
   Set $k = 1$.
2. At $k \geq 1$ iteration, apply $P_{\mathbf{U}}^{\perp}$ via (2.78) to all data sample vectors $\mathbf{r}$ and find the kth data sample vector $\mathbf{t}^{(k)}$, which has the maximum orthogonal projection defined by

$$\mathbf{t}^{(k)} = \arg\left\{\max_{\mathbf{r}}\left[\left(P_{[\mathbf{U}^{(k-1)}]}^{\perp}\mathbf{r}\right)^{T}\left(P_{[\mathbf{U}^{(k-1)}]}^{\perp}\mathbf{r}\right)\right]\right\} \tag{30.1}$$

   where $\mathbf{U}^{(k-1)} = \left[\mathbf{t}^{(0)}\mathbf{t}^{(1)}\cdots\mathbf{t}^{(k-1)}\right]$ is the data matrix and $\mathbf{U}^{(k-1)} = \emptyset$ if $k - 1 = 0$.
3. Calculate

$$\eta_k = \left(\mathbf{t}^{(k)}\right)^{T} P_{\mathbf{U}^{(k-1)}}^{\perp}\mathbf{t}^{(k)} \tag{30.2}$$

   and compare η_k to a prescribed threshold ε.
4. Stopping rule:
   If $\eta_k < \varepsilon$, go to step 5. Otherwise, continue.
5. At this stage, ATGP is terminated and the target $\mathbf{U}^{(k)} = \left[\mathbf{U}^{(k-1)}\mathbf{t}^{(k)}\right] = \left[\mathbf{t}^{(1)}\mathbf{t}^{(2)}\ldots\mathbf{t}^{(k)}\right]$ generated at this point is the desired target set, which contains k targets.

It should be noted that the stopping rule in step 4 used by ATGP described above is different from that used by ATGP-EEA in Section 8.5.1 where the former makes use of η_k in (30.2) to measure the residual from two consecutive OPs while the latter uses VD to determine the numbers of targets required to generate p.

30.3 Experiments

In this section, synthetic and real hyperspectral image experiments were conducted to substantiate and validate the utility of ATGP-FCLS in size estimation of subpixel targets. The data to be used for experiments was the HYDICE image shown in Figure 1.15(a) with ground truth map of Figure 1.15(b).

30.3.1 Synthetic Image Experiments

In order to evaluate our approach, we simulated a synthetic image scene based on Figure 1.15(a) and (b) and the five panel signatures in Figure 1.16. First of all, 400 grass samples extracted from Figure 1.15(a) were used to simulate the image background. Two panel signatures, $\mathbf{p}_1$ and $\mathbf{p}_2$ were used to simulate two sets of target panels, p_{11}, p_{12}, p_{13} and $p_{21}p_{22}$, p_{23}, respectively. The three panels in each set had size ranging from 100%, 50%, and 25% of pixel size (i.e., GSD = 1.5 m), namely, 2.25 m^2, 1.125 m^2, and 0.5625 m^2, respectively. Figure 30.1(a) and (b) shows how the target panels p_{12} and p_{13} with size being 50% and 25% of pixel size were simulated. For example, in order to simulate the panel p_{12}, we first simulated 2-pixel vectors specified by the grass signature and 2-pixel vectors specified by the panel signature $\mathbf{p}_1$ to form a 4-pixel square panel where each of

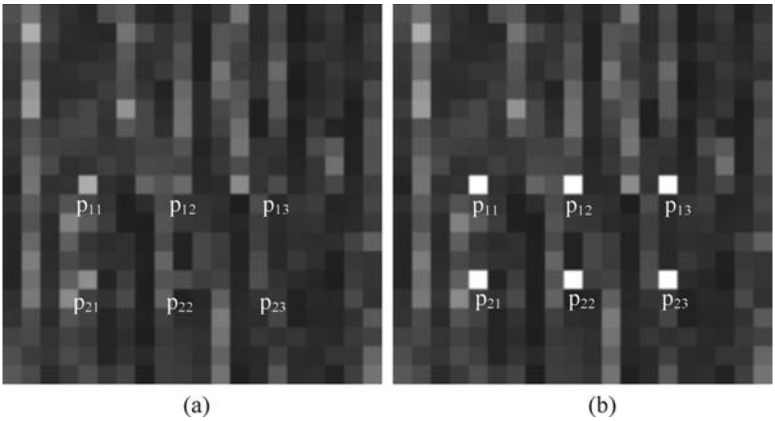


Figure 30.1 (a) Simulated image scene and (b) ground truth map of six implanted panel targets.

4-pixel vectors in the panel had size of 2.25 m^2 and the resulting 4-pixel square panel had size of 9 m^2. This 4-pixel square panel was then shrunk to its $^1\!/_4$ size by averaging all 4-pixel vectors to reduce the 4-pixel square panel with size of 9 m^2 to a 4-pixel square panel with size of 2.25 m^2 where each pixel vector was only $^1\!/_4$ size of its corresponding pixel vector as a result of $1/4(\mathbf{p}_1 + \mathbf{p}_1 + \text{grass} + \text{grass})$. This shrinking process was similar to the pyramid method developed by Burt and Adelson (1983) illustrated in Figure 4.1, where a pixel in an upper layer was obtained by averaging 4 pixels in its immediate lower layer in a pyramid. The two sets of the six simulated targets, p_{11}, p_{12}, p_{13} and p_{21}, p_{22}, p_{23}, were then implanted in the image background as shown in Figure 30.1(a) with the ground truth map of the six implanted target panels shown in Figure 30.1(b) where their precise spatial coordinates are specified in the parentheses in p_{11}(10,5), p_{12}(10,10), p_{13}(10,15), p_{21}(15,5), p_{22}(15,10), and p_{23}(15,15).

Assume that no prior knowledge was provided for the simulated image scene. So the exact locations of these six target panels were not supposed to be known *a priori* and must be found by an unsupervised method. In this case, ATGP was used to find these six targets plus other potential targets in the simulated image scene in Figure 30.1(a). As a result, eight targets were generated and shown in Figure 30.2 where they were labeled in accordance with the order that they were found by ATGP.

In order to see how effective ATGP worked, the ground truth map in Figure 30.1(b) was used to verify the eight ATGP-generated targets. It turned out that the pixels labeled by 2 and 6 corresponded to panels P$_{11}$ and P$_{21}$ and the other six targets, numbered by 1, 3, 4, 5, 7, and 8, were not panels. One comment is worthwhile. Since ATGP used the OSP projector $P^{\perp}_{[2,6]}$ to project all the image pixel vectors into the space that was orthogonal to the space linearly spanned by the targets, #2 and #6. As a result, the signatures that were similar to signatures of #2 and #6 were annihilated by $P^{\perp}_{[2,6]}$. The $P^{\perp}_{[2,6]}$-eliminated pixel vectors included the subpixel panels, p_{12}(10,10), p_{13}(10,15), p_{22}(15,10), and p_{23}(15,15), that were specified by signatures of #2 and #6. As a consequence, they were not detected by ATGP after the targets #2 and #6 were extracted. As we noted, the six targets #1, #3, #4, #5, #7, and #8 were not panel pixels, and they were not annihilated by $P^{\perp}_{[2,6]}$. Therefore, these six targets were detected by ATGP subsequently. Now, we used these eight ATGP-generated targets as *a posteriori* target information to find other potential targets in the image scene in Figure 30.1(a). Four methods were used for this purpose, LSOSP, sum-to-one constrained least squares (SCLS), NCLS, and FCLS. Their respective results are shown in Figure 30.3 where SCLS was particularly included for comparison since it is also a partially abundance-constrained least squares unmixing method developed in Heinz and Chang (2001) and Chang (2003a).

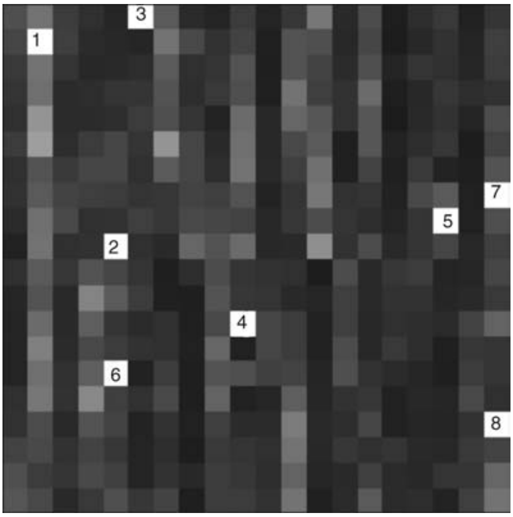


Figure 30.2 Eight targets detected by ATGP.

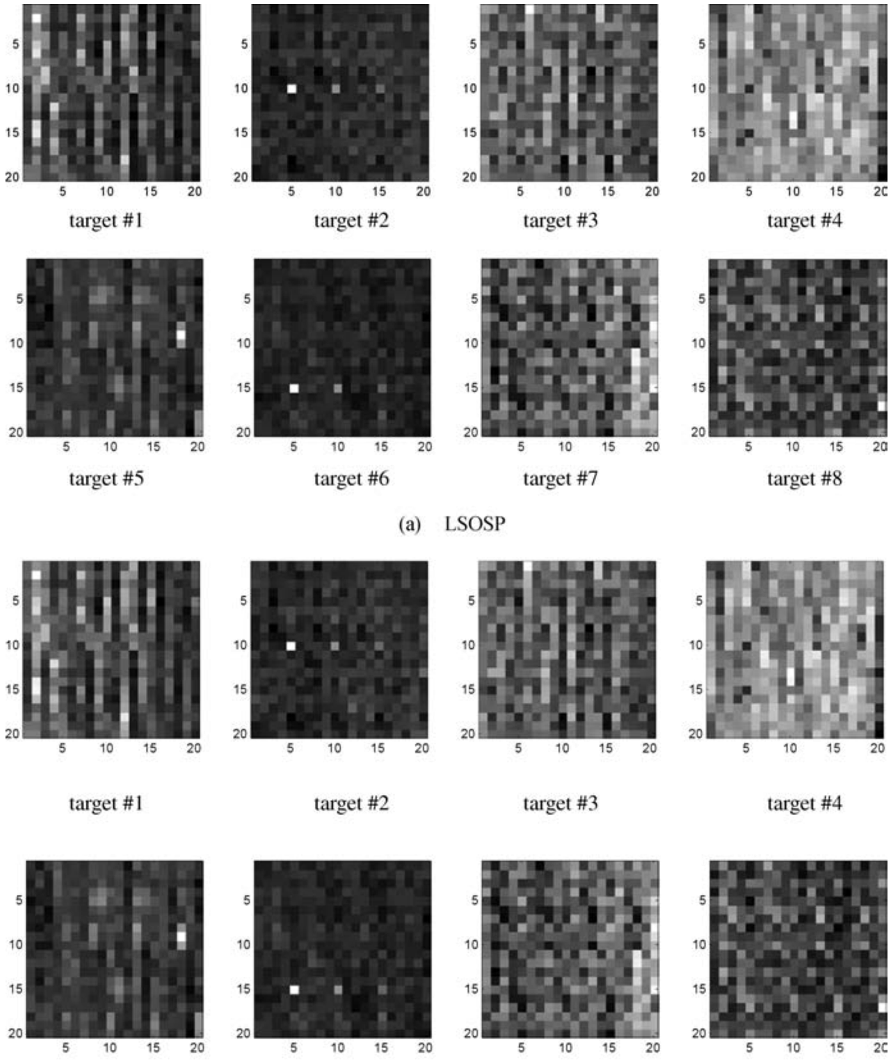
As we can see, when the detected target #2 was used as the desired target information the other seven targets (i.e., targets #1, #3, #4, #5, #6, #7, and #8) were considered as undesired targets and annihilated by LSOSP. As a result, the second images in Figure 30.3(a) and (d) only show the targets specified by the signature of target #2 where all the four methods picked up additional two target panels, p_{12} and p_{13} whose signatures were similar to the signature of target #2. Similarly, when the target #6 was used as the desired target information and the other seven targets (i.e., targets #1, #2, #3, #4, #5, #7, and #8) were considered as undesired targets and annihilated by LSOSP, all the four methods also pulled out additional two target panels, p_{22} and p_{23}, in the sixth image in Figure 30.3(a) and (d). These experiments demonstrated that LSOSP, SCLS, NCLS, and FCLS methods could be used for subpixel target detection.

Since our main focus is size estimation of subpixel targets and only the targets p_{11}, p_{12}, p_{13} and p_{21}, p_{22}, p_{23} had ground truth to validate our results only these six targets would serve as our interest for the follow-up size estimation. Of course, we could also conduct experiments for the other six found nonpanel targets. But, their results could not be used to substantiate our algorithm due to the lack of ground truth. Because of that, experiments of these six nonpanel targets, #1, #3, #4, #5, #7, and #8, are not included in this chapter.

In order to see whether the four methods, LSOSP, SCLS, NCLS, and FCLS, could be further used to estimate size of these panel targets, we calculated the abundance fractions generated by these four methods. Table 30.1 tabulates their corresponding results. From Table 30.1, FCLS yielded the best results while NCLS was the second best, but very close to FCLS. Table 30.2 also tabulates the error percentage obtained from Table 30.1 by (30.3), which computed the ratio of the estimation error of target size to the true target size:

$$E = 100\% \times \frac{|\text{true size} - \text{estimated size}|}{\text{true size}}. \tag{30.3}$$

As shown in Table 30.2, the estimated error was increased as the target size was decreased. When a target fully occupied a pixel, no estimated errors were produced by all the four methods. However, when the target size was only one-quarter of a pixel size, FCLS performed slightly better than NCLS, but significantly better than LSOSP and SCLS.



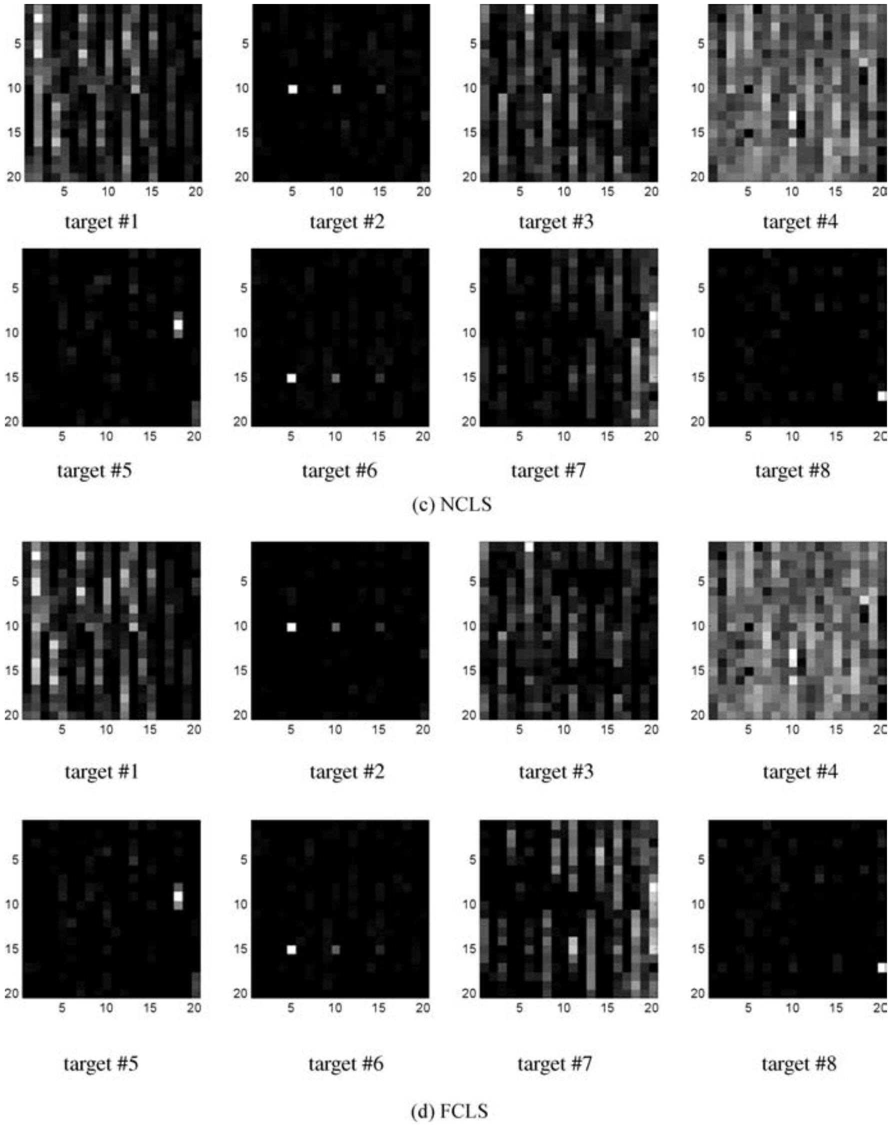
(a) LSOSP

(b) SCLS

Figure 30.3 Unmixed results of Figure 30.1(a) by (a) LSOSP; (b) SCLS; (c) NCLS; (d) FCLS.

These experiments provided evidence that fully abundance-constrained method was required to achieve better target size estimation when targets were smaller than pixel resolution.

It is worth noting that ATGP is an unsupervised method. When it is used, it assumes that no knowledge is provided *a priori*. Consequently, the knowledge of ATGP-generated targets remain unknown. In order to identify these target signatures, a database or spectral library is generally required. In our simulations, ATGP was implemented with no given prior knowledge. The eight targets extracted by ATGP were still unknown. The ground truth of the simulated image in Figure 30.1(b) was only used to identify which detected target belonged to the



(c) NCLS

(d) FCLS

Figure 30.3 (*Continued*)

implanted panels so that these targets could be further used to evaluate the effectiveness of subpixel target size estimation.

Finally, a concluding remark on the above experiment is noteworthy. Although the concept of virtual dimensionality (VD) developed in Chapter 5 can also be used to estimate the number of target pixels to be detected in this experiment, VD generally underestimates the number of unsupervised targets as will be demonstrated in the following section. This is because VD is designed to detect spectrally distinct signatures not necessarily distinct target pixels, in which case two target pixels with similar spectral signatures can be detected as distinct targets. One such type of targets is anomalies discussed in Chapter 18. So, VD was not used in this experiment. Instead,

Table 30.1 Abundance fractions estimated by LSOSP, SCLS, NCLS, and FCLS

	LSOSP	SCLS	NCLS	FCLS
p_{11} (1 pixel)	1.0000	1.0000	1.0000	1.0000
p_{12} (50% of pixel)	0.5130	0.4952	0.5142	0.4958
p_{13} (25% of pixel)	0.3387	0.3256	0.3031	0.2908
p_{21} (1 pixel)	1.0000	1.0000	1.0000	1.0000
p_{22} (50% of 1 pixel)	0.5642	0.5814	0.5189	0.4710
p_{23} (25% of 1 pixel)	0.3699	0.3917	0.2877	0.2206

Table 30.2 Error (%) resulting from LSOSP, SCLS, NCLS, and FCLS

	LSOSP	SCLS	NCLS	FCLS
p_{11} (1 pixel)	0.00	0.00	0.00	0.00
p_{12} (1 pixel)	2.60	0.96	2.85	0.83
p_{13} (1 pixel)	35.46	30.22	21.23	16.34
p_{21} (1 pixel)	0.00	0.00	0.00	0.00
p_{22} (1 pixel)	12.85	16.29	3.78	5.80
p_{23} (1 pixel)	47.95	56.70	15.06	11.77

selecting 8 as the number of targets of interest was purely an empirical choice based on the fact that there were five panel signatures and three background signatures to account for three scenarios of abundance fractions, 25%, 50%, and 75%, of background signature.

30.3.2 HYDICE Image Experiments

In this section, the HYDICE image scene in Figure 1.15(a) was used for real image experiments to further validate the idea proposed in this chapter. As noted, the panels, p_{13}, p_{23}, p_{33}, p_{43}, and p_{53}, in the third column have spatial resolution of 1 m that is smaller than the pixel size, 1.5 m. These five panels could serve as subpixel targets to evaluate the four methods, LSOSP, SCLS, NCLS, and FCLS. By assuming that no ground truth was provided, ATGP was applied to find 41 targets shown in Figure 30.4 where the threshold was set to 2.54×10^5, as the same threshold used in Heinz and Chang (2001).

These obtained 41 targets were then used as *a posteriori* target information to classify the HYDICE image scene in Figure 1.15(a). Now, if we compared these 41 detected targets against the ground truth in Figure 1.15(b), we found that the 10th, 34th, 5th, and 4th targets corresponded to p_{11}, p_{211}, p_{312}, and p_{521}, respectively. Additionally, both the 23rd and 36th targets detected two panel pixels, p_{411}, p_{412} in row 4. Since only these targets had ground truth to verify our results, Figure 30.5 only shows the classification results produced by these targets where the classification results in columns (d) and (e) were produced by the two panel pixels p_{411}, p_{412} labeled by ATGP-detected 23rd and 36th targets. Apparently, using these six targets, all the four methods demonstrated different degrees of detecting and classifying the 19 R panel pixel vectors.

Table 30.3 tabulates the abundance fractions of all the 19 R panel pixel vectors detected in Figure 30.5 by the four methods, LSOSP, SCLS, NCLS, and FCLS.

Since both panel pixels p_{411}, p_{412} vectors were detected by ATGP as the 23rd and 36th targets, there were two rows of abundance fractions for each of the three panels in row 4, p_{411}, p_{412}, p_{42}, and p_{43} in Table 30.3 where the abundance fractions in the 1st and 2nd rows were detected by using the signatures of the target #23 and target #36, respectively. Interestingly, despite the fact that

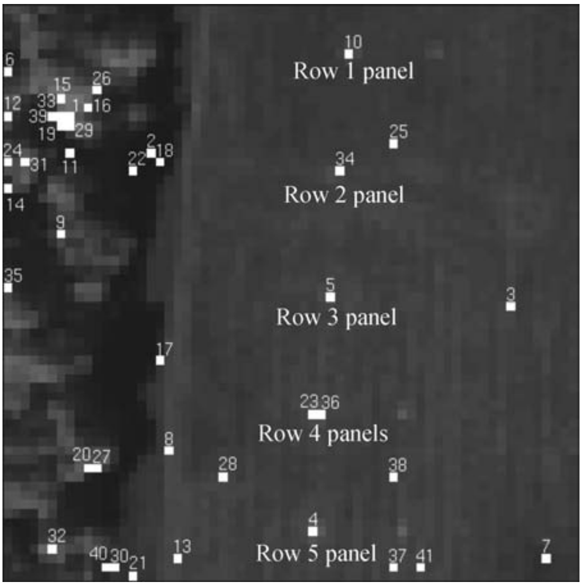
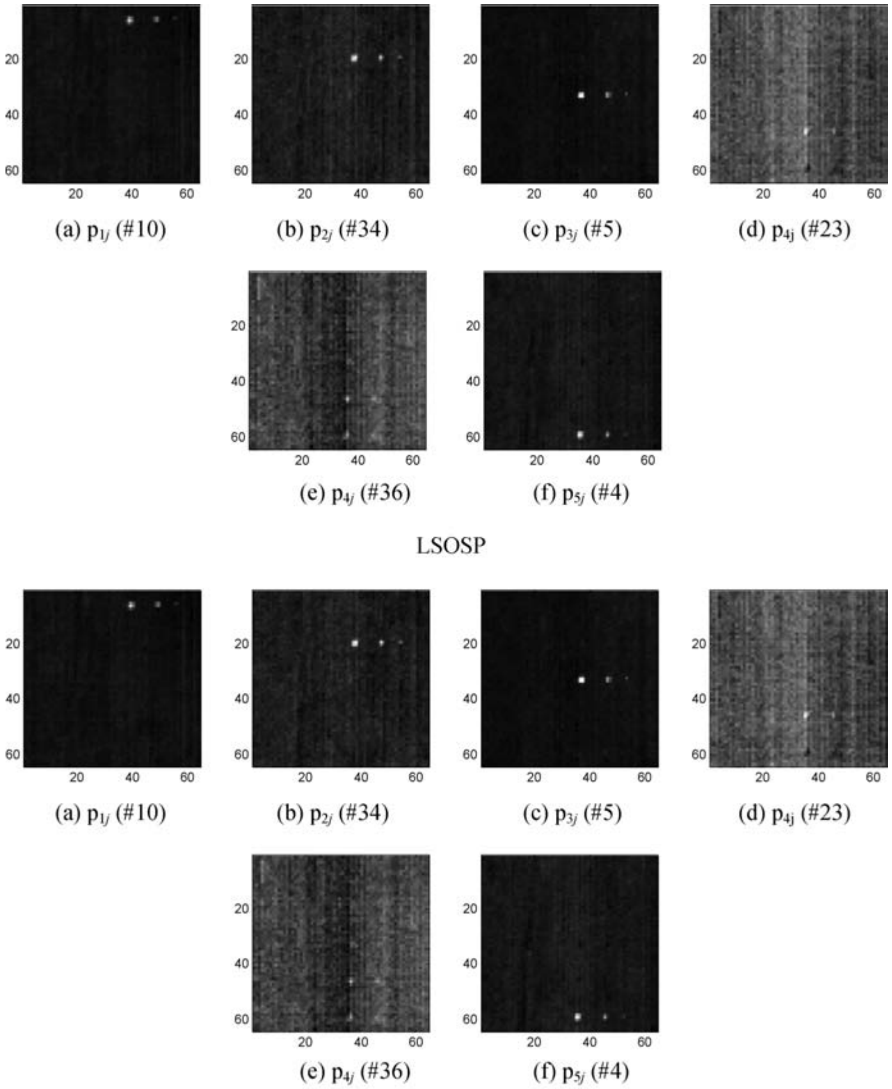


Figure 30.4 Forty-one targets detected in the scene of Figure 1.15(a) by ATGP.

ATGP detected both panel pixels p_{411}, p_{412} they were detected by target #23 and target #36 as distinct target pixels. However, only the information of the target #23 could pull out all the three panels in row 4. This phenomenon was evidenced by Table 30.3 where the abundance fractions in the second row were detected by LSOSP and SCLS using the target #36 were negative and were zeros detected by NCLS and FCLS. There were some additional interesting results, which could also be observed from Table 30.3. According to the results in Table 30.3, the panel pixel p_{11} was detected as a pure pixel, the panel pixels, p_{312} and p_{521} were detected as two-pixel panels which contained at least one pure pixel, and the panel in row 4 was made up of two pure pixels, p_{411}, p_{412}. The sizes of these four panels estimated by the four methods were very close. The only exception was the panel in row 2 which was found to be composed of two mixed pixels p_{211}, p_{221}. As a result, its panel size estimated by LSOSP and SCLS was twice much that estimated by NCLS and FCLS. It is also interesting to note that the size of all the five targets, 10th, 5th, 23rd, 36th, and 4th, found by ATGP was estimated to be of full pixel size compared to the 34th target, which had an approximate $4/5$ size of a full pixel size. This is because the 34th target pixel detected by ATGP was a Y pixel vector, not an R pixel vector like the other five 10th, 5th, 23rd, 36th, and 4th targets. In this case, the 34th target was considered as a mixed pixel vector, but was detected by ATGP as a pure pixel vector. Consequently, the four methods, which used the 34th target to produce the two R pixel vectors, p_{211}, p_{221} for the panel in row 2, resulted in inaccurate abundance fraction estimates of the R pixels. As an example, Table 30.4 tabulates FCLS estimated abundance fraction map of all the R and Y pixels that masked the panel p_{21}. As shown in Table 30.4, the 34th target appeared as a Y pixel vector with its abundance estimated by FCLS as 1.0000, which indicated that the Y pixel vector was a pure pixel vector. The two real R pixel vectors, p_{211}, p_{221} were next to its left and lined up vertically to make up the panel in row 2. They were estimated by FCLS as mixed pixels with their corresponding abundance fractions, 0.5216 and 0.3455, respectively.

Finally, according to the ground truth provided in Figure 1.15(b), it should be noted that panels, P_{21}, P_{31}, P_{41}, and P_{51} are of 2-pixel size, that is, P_{21} made up of two panel pixels, p_{211} and p_{221}; P_{31} made up of two panel pixels, p_{311} and p_{312}; P_{41} made up of two panel pixels, p_{411} and p_{412}; P_{51}



(a) p_{1j} (#10) (b) p_{2j} (#34) (c) p_{3j} (#5) (d) p_{4j} (#23)

(e) p_{4j} (#36) (f) p_{5j} (#4)

LSOSP

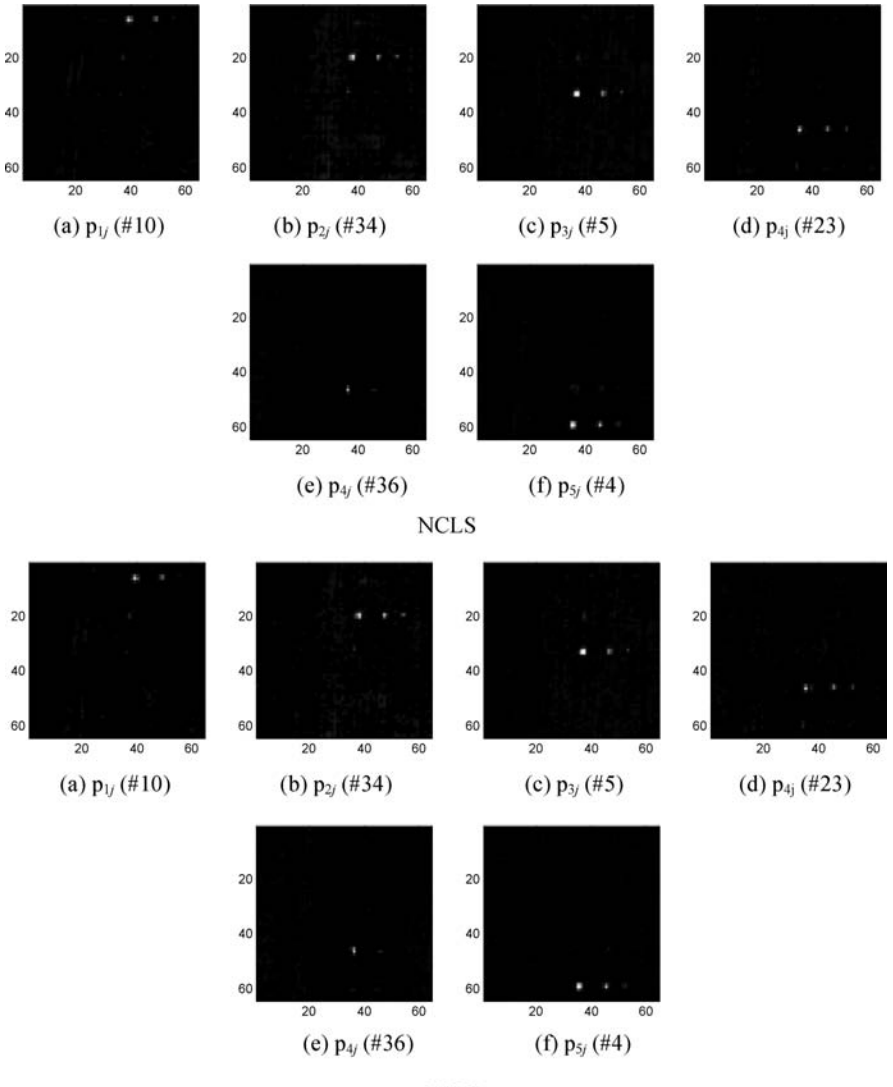
(a) p_{1j} (#10) (b) p_{2j} (#34) (c) p_{3j} (#5) (d) p_{4j} (#23)

(e) p_{4j} (#36) (f) p_{5j} (#4)

SCLS

Figure 30.5 Abundance fractions of the 15 panels in Figure 1.15(b) estimated by LSOSP, SCLS, NCLS, and FCLS.

made up of two panel pixels, p_{511} and p_{521}. Despite the fact that ATGP only detects p_{211} labeled by #34, p_{312} labeled by #5, p_{411} labeled by #23, p_{412} labeled by #36, and p_{521} labeled by #4, FCLS did manage to extract three ATGP-missed panel pixels, p_{221}, p_{311}, and p_{511} as shown in Table 30.3. It is interesting to compare the results in Figure 8.10(i) produced by ATGP-EEA, which extracted VD-determined $n_{\mathrm{VD}} = 9$ target pixels that only included three panel pixels, p_{11}, p_{312}, and p_{521} as endmember pixels. It demonstrated that ATGP-FCLS using 41 unsupervised target pixels was able to extract all 19 R pixels including six pure panel pixels, which are p_{11}, p_{211}, p_{311}, p_{411}, p_{412}, and p_{521}. This example demonstrated that ATGP implemented as an unsupervised method did not always extract pure pixels, in which case ATGP was better considered as unsupervised target detection



(a) p_{1j} (#10) (b) p_{2j} (#34) (c) p_{3j} (#5) (d) p_{4j} (#23)

(e) p_{4j} (#36) (f) p_{5j} (#4)

NCLS

(a) p_{1j} (#10) (b) p_{2j} (#34) (c) p_{3j} (#5) (d) p_{4j} (#23)

(e) p_{4j} (#36) (f) p_{5j} (#4)

FCLS

Figure 30.5 (*Continued*)

algorithm as also demonstrated in Chapter 17 than being considered as an endmember extraction algorithm. Therefore, VD understated the number of unsupervised targets. This interesting issue will be further explored in Chang (2013).

With the GSD of 1.56 m, the 15 panels with size of 3 m × 3 m, 2 m × 2 m and 1 m × 1 m shown in Figure 1.15 are supposed to occupy three different 3.70, 1.64, and 0.41 pixel sizes, respectively, where an abundance fraction of 1.0 corresponds to 1 pixel size. Obviously, the abundance fraction estimates of the R pixel vectors in Table 30.3 did not provide accurate size estimation for each of the 15 panels. This is because some of abundance fractions have spread over Y pixel vectors that surround the R pixel vectors. The leakage of such abundance fractions into Y pixel vectors must be included to account for target size estimation. The reason for this inclusion can be best explained by the following example. If we would like to reconstruct an exploded airplane from its debris

Table 30.3 Abundance fractions of all the 19 R panel pixel vectors in the 15 panels estimated by LSOSP, SCLS, NCLS, and FCLS in Figure 3.5

	LSOSP	SCLS	NCLS	FCLS
p_{11} (#10)	1.0000	1.0000	1.0000	1.0000
p_{12}	0.4701	0.4634	0.4450	0.4091
p_{13}	0.1190	0.1135	0.0804	0.0496
p_{211} (#34)	0.8144	0.8062	0.5824	0.5216
p_{221}	0.8273	0.8283	0.4516	0.3455
Total	1.6417	1.6345	1.0340	0.8671
p_{22}	0.7330	0.7701	0.6169	0.6899
p_{23}	0.3218	0.3527	0.3841	0.4177
p_{311}	0.9052	0.9052	0.8640	0.8647
p_{312} (#5)	1.0000	1.0000	1.0000	1.0000
Total	1.9052	1.9052	1.8640	1.8647
p_{32}	0.5985	0.5985	0.5292	0.5367
p_{33}	0.3997	0.3997	0.3549	0.3614
p_{411} (#23)	1.0000	1.0000	1.0000	1.0000
p_{412} (#36)	1.0000	1.0000	1.0000	1.0000
Total	2.0000	2.0000	2.0000	2.0000
p_{42}	0.4318	0.4606	0.5464	0.5306
	0.4713	0.4571	0.1616	0.1938
Total	0.9031	0.9177	0.7080	0.7244
p_{43}	0.3966	0.3953	0.2234	0.2543
	−0.0914	−0.0908	0	0
Total	0.3052	0.3045	0.2234	0.2543
p_{511}	0.7216	0.7178	0.7216	0.7217
p_{521}(#4)	1.0000	1.0000	1.0000	1.0000
Total	1.7216	1.7178	1.7216	1.7217
p_{52}	0.6646	0.6637	0.7801	0.7769
p_{53}	0.1057	0.0959	0.1537	0.1412

spread over a wide range of areas, we need to find all the pieces all over locations in order to reassemble the airplane, even though the size of the airplane is relatively small compared to a large spread area of its debris. In light of this interpretation Table 30.4 tabulates FCLS-estimated abundance fractions of 2 R and 12 Y pixel vectors surrounding the panel P_{21} where the two R pixel vectors at the center make up the panel P_{21} and the other 12 Y pixel vectors are their neighboring pixel vectors mixed with the background. According to Table 30.4, the estimated abundance fractions of the two R pixel vectors are 0.5216 and 0.3455, which indicated that the panel P_{21} was not a single pure pixel vector. So, if only abundance fractions of R pixel vectors, p_{211}, p_{221} in Table 30.3 were used to estimate the size of the panel P_{21}, its size would be 0.8671 pixel size, i.e.,

Table 30.4 FCLS-estimated abundance fraction map of R and Y pixel vectors that mask the panel P_{21}

0	0	0.1057	0.2199	0.0846
0.0171	0.1017	0.5216 R panel pixel, p_{211}	1.0000	0.1158
0	0.1550	0.3455 Red panel pixel, p_{221}	0.7520	0.0574
0.0123	0	0	0.0841	0

Table 30.5 Abundance fractions of all the R and Y panel pixel vectors in the 15 panels estimated by LSOSP, SCLS, NCLS, and FCLS in Figure 30.5

	LSOSP	SCLS	NCLS	FCLS
p_{11}	3.2094	3.1368	3.1432	3.0329
p_{12}	1.0191	0.9728	1.2892	1.2639
p_{13}	−0.2302	−0.2254	0.0885	0.0496
P_{21}	5.2066	4.9628	3.7033	3.5727
p_{22}	1.2023	1.3001	1.1969	1.4674
p_{23}	0.4698	0.5247	0.5672	0.6148
P_{31}	4.0116	4.0119	4.0922	4.0978
p_{32}	2.5240	2.5239	1.9474	2.0073
p_{33}	0.7715	0.7715	0.4934	0.5144
P_{41}	3.7346	3.7762	3.5161	4.1374
p_{42}	2.6265	2.7857	1.7554	1.9011
p_{43}	0.6861	0.7495	0.5367	0.6132
P_{51}	4.5061	4.5382	3.8530	3.6857
p_{52}	1.5076	1.5119	1.8493	1.8321
p_{53}	0.2772	0.1859	0.4666	0.4604

$0.5216 + 0.3455 = 0.8671$, which may not be accurate. However, if the estimated abundance fractions of 12 Y pixel vectors in Table 30.4 were included in size estimation, the total estimated abundance fraction would be 3.5727, which corresponds to 3.5727 pixel size, very close to its true 3.70 pixel size. This simple example suggested that in order to reliably estimate the size of panel targets, we need to include all estimated nonzero abundance fractions that contributed to the panels. It was indeed the case shown in Table 30.5, which tabulates abundance fractions of the R and Y pixel vectors in the 15 panels estimated by LSOSP, SCLS, NCLS, and FCLS in Figure 30.5.

Compared to Table 30.3, which only computed the estimated abundance fractions of R pixel vectors for target size estimation, Table 30.5 produced more accurate results by including the estimated abundance fractions of Y pixel vectors.

30.4 Concealed Target Detection

As another example for target detection applications, this section investigates a more difficult problem, concealed target detection, which has not received much attention in the past. On many occasions, a concealed target can be camouflaged by man-made objects or shaded by natural background. The main issue in detection of concealed targets is how to remove the shadow covered on or over the target surface to expose the targets for detection. One effective means is band ratio (Robinove, 1982; Crippen, 1988), which can be used for this purpose. However, due to the fact that hyperspectral images are generally collected by hundreds of contiguous spectral bands, it is very challenging to select appropriate bands to be used for band ratio. Exhausting all possible pair combinations does not seem to be practically feasible. Accordingly, one key component in detecting concealed targets is how to select proper pairs of bands to compute their band ratio.

Band selection has been widely used for various purposes for data dimensionality, data compression, feature extraction, etc. (see Chapters 6, 19, 21–23). Many criteria for band selection have been proposed in the past to find bands that retain most of information. For instance, distance measures (Bhattacharyya distance, Jeffreys–Matusita distance), information theoretic approaches (divergence, transformed divergence, and mutual information), and eigen analysis (principal components analysis (PCA)) have been applied to multispectral images for optimal band selection.

In particular, the use of the divergence measure for band selection has shown promise in hyperspectral imagery where it is also used as a band selection criterion for hyperspectral pixel classification (Chang et al., 1999). However, it requires computing divergences for all the possible combinations of band subsets. When the divergence measure is applied to hyperspectral imagery, such a direct calculation of divergences becomes formidable. In order to alleviate this problem, an alternative is also reported in Stearns et al. (1993) and Chang et al. (1999). However, according to our experiments, the divergence and other aforementioned criteria are not suitable for the band ratio. So, this section presents a new band selection that is based on a criterion, called between-bands orthogonal projection correlation (BBOPC) whose idea arises from the orthogonal subspace projection (OSP) approach developed by Harsanyi and Chang (1994). First of all, we can arrange image pixel vectors in a 2D band image, $\mathbf{B}_l$ line by line as a 1D band image pixel vector, $\mathbf{b}_l$.

BBOPC Band Selection Algorithm

1. Apply a pyramid coding to reduce the size of band images, $\Omega = \{\mathbf{B}_l\}_{l=1}^{L}$.
2. Construct a band image vector for each reduced band image. Let $\tilde{\Omega}$ be the set of entire band images, denoted by $\tilde{\Omega} = \{\mathbf{b}_l\}_{l=1}^{L}$ where $\mathbf{b}_l$ is the lth band image vector constructed from the lth band image $\mathbf{B}_l$ and L is the total number of band images. Assume that the initial band set is $\tilde{\Omega}_0 = \emptyset$.
3. Select a band image vector with the largest vector norm as the first initial band image denoted by $\mathbf{b}_{\max,1}$ where the norm is defined by the length of a vector, that is,

$$\mathbf{b}_{\max,1} = \arg\left\{\max_{\mathbf{B}_l\in\Omega}\mathbf{b}_l^T\mathbf{b}_l\right\} \tag{30.4}$$

   and let $\tilde{\Omega}_1 = \tilde{\Omega}_0 \cup \left\{\mathbf{b}_{\max,1}\right\}$.
4. At the kth stage, calculate the orthogonal projections of all band image vectors in $\tilde{\Omega} - \tilde{\Omega}_{k-1}$ and find the one with the maximal orthogonal projection, denoted by $\mathbf{b}_{\max,k}$,

$$\mathbf{b}_{\max,k} = \arg\left\{\max_{\mathbf{b}_j\in\tilde{\Omega}-\tilde{\Omega}_k}\left(P_{\tilde{\Omega}_{k-1}}^{\perp}\mathbf{b}_j\right)T\left(P_{\tilde{\Omega}_{k-1}}^{\perp}\right)\right\} \tag{30.5}$$

   where $\tilde{\Omega}_{k-1}$ be the band set selected at the $k-1$st stage. Then, let $\tilde{\Omega}_k = \tilde{\Omega}_{k-1} \cup \left\{\mathbf{b}_{\max,k}\right\}$
5. Compute and check if the orthogonal projection correlation index (OPCI) given by

$$\eta_k = \mathbf{b}_{\max,\,k}^T P_{\tilde{\Omega}-\tilde{\Omega}_k}^{\perp}\mathbf{b}_{\max,k} < \varepsilon \tag{30.6}$$

6. If (30.6) is true, the algorithm is terminated. Otherwise, go to step 4. It should be noted that at the worst case, BBPOC is always terminated after it exhausts all the bands.

30.5 Computer-Aided Detection and Classification Algorithm for Concealed Targets

The band ratio has been commonly used for multispectral data to reduce the effects caused by topological slope and aspects or to eliminate differential illumination effects caused by shadows (Jensen, 1996).

Let $\mathbf{B}_j$ and $\mathbf{B}_k$ be the jth and kth band image vectors. A band-ratioed image vector obtained by taking the ratio of $\mathbf{B}_j$ to $\mathbf{B}_k$, denoted by $\mathbf{BR}_{jk}$, is defined as follows:

$$\mathbf{BR}_{jk} = \mathbf{B}_j/\mathbf{B}_k. \tag{30.7}$$

Assume that the gray level range for all band image vectors is given by $\{g_1, g_2, \ldots, g_T\}$ in ascending order. In case there is a pixel in $\mathbf{B}_k$ taking gray level value 0, the gray level of the corresponding pixel in the band ratio image vector $\mathbf{BR}_{jk}$ of (30.7) will be simply set to $\mathbf{B}_j$ to prevent the denominator of (30.7) from taking 0. As a result, the gray level range of the $\mathbf{BR}_{jk}$ is between $1/g_T$ and g_T.

Now, combining ATGP in conjunction with the BBOPC-based band selection algorithm discussed in Section 30.4 along with the band-ratioed image obtained by (30.7), the desired CADCA for concealed targets can be implemented as follows.

Computer-aided detection and classification algorithm

1. Apply BBOPC band selection algorithm to select an appropriate band set $\tilde{\Omega}_k$ for the band ratio.
2. Apply the band ratio specified by (30.7) to generate band-ratioed images.
3. Apply ATGP to detect concealed targets.
4. Apply a target classifier to perform classification of each detected concealed targets.

30.6 Experiments for Concealed Target Detection

Two HYDICE datasets to be used in the experiments were two different image scenes taken in Maryland in August 1995 using 210 bands with spectral coverage 0.4–2.5 μm of resolution 10 nm and GSD approximately 0.78 m. Figure 30.6 shows an image scene with size of 128 × 128 pixel vectors. Three vehicles of the same type are parked underneath the trees on the left edge and aligned vertically and circled by V1, V2, and V3 from bottom to top. Except V1, which is partially revealed, the other two vehicles V2 and V3 are completely concealed.

In this experiment, BBOPC used a two-layer pyramid to reduce the image size and generated seven bands (band numbers: 16, 41, 53, 57, 63, 83, and 96) with $\varepsilon = 0.0003$ for band ratio transformation, which resulted in $7 \cdot 6 = 42$ band-ratioed images obtained by (30.7). It was then followed by ATGP where 12 target signatures were detected and generated and each target signature was classified in an individual image. It should be noted that many of the targets detected were background signatures such as grass, trees, or unknown interferers. Figure 30.7(a) only shows that three vehicles V1, V2, and V3 in Figure 30.6 were effectively detected and classified by CADCA. Since CADCA is an abundance estimation-based classifier, the resulting images were gray-scaled.

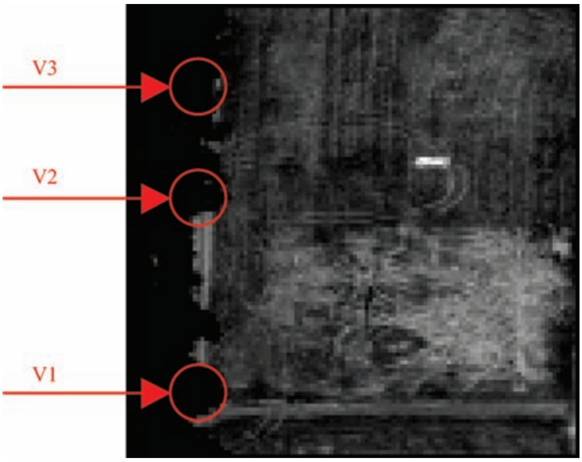
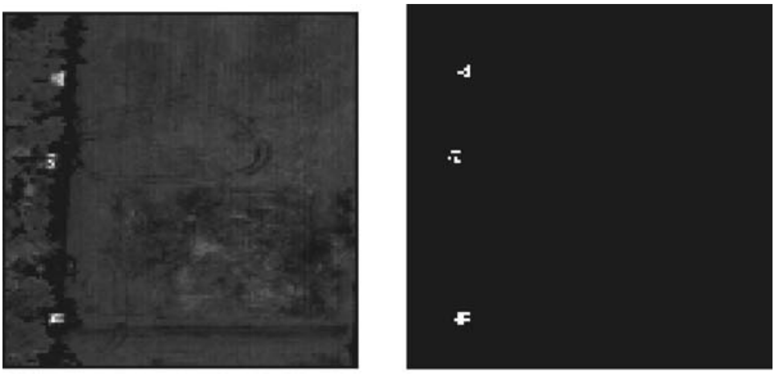


Figure 30.6 A HYDICE image scene.



(a) Gray-scaled image

(b) Thresholded image

Figure 30.7 Detected concealed targets: (a) gray-scaled image; (b) thresholded image.

In order to segment the detected and classified targets from the background, we used a Winner-Take-All (WTA) rule to threshold the gray-scaled image in Figure 30.7(a) and produced its thresholded image in Figure 30.7(b) where only concealed vehicles were shown in the image. The WTA rule used here was based on the maximum of abundance fractions resident within a pixel. It calculated the abundance fractions of 12 target signatures present in a pixel vector estimated by CADCA and assigned the pixel the target whose value yielded the maximal abundance. As a result of this WTA assignment, a gray-scaled image in Figure 30.7(a) was converted to a binary image in Figure 30.7(b) with one assigned to the target and zero otherwise.

Figure 30.8 shows another HYDICE image scene of size 128 × 128 pixel vectors and has a large grass field with tree lines running along the left edge and three vehicles are parked underneath trees where there are two vehicles, circled by V1 (bottom one) and V2 (upper one) along with the left tree line and a third circled by V3 hidden under the top tree line. In addition to these three vehicles, two different material-made objects, circled by O1 (bottom one) and O2 (upper one), are located near the center to the left. Covered under O1 and O2 are two vehicles, indicated V4 and V5, respectively. Except V2, all other four vehicles, V1, V3, V4, and V5, belong to the same type of vehicles. As shown in Figure 30.8, all five vehicles are completely concealed.

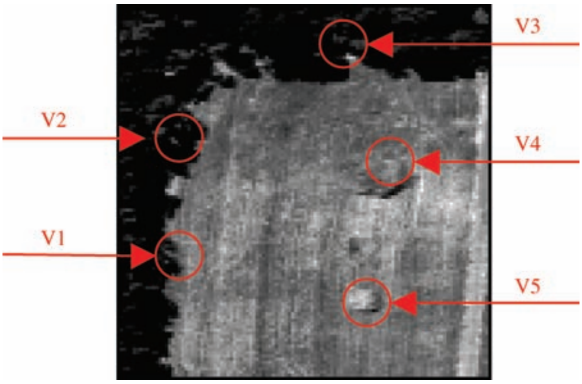


Figure 30.8 Another HYDICE image scene.

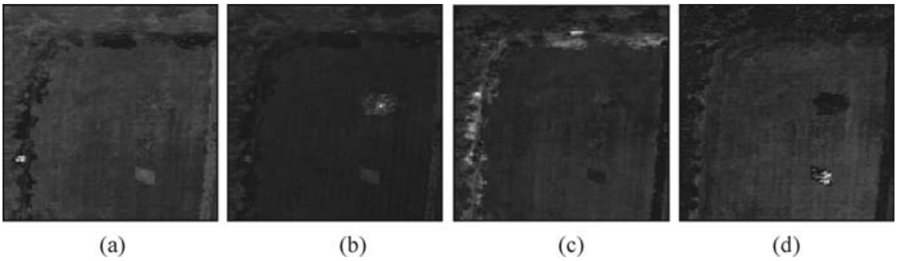


Figure 30.9 Detected concealed targets.

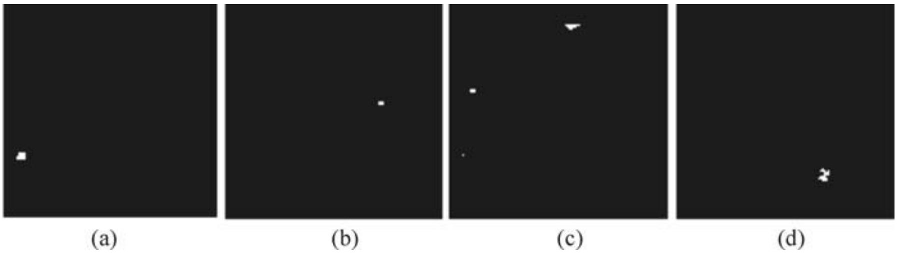


Figure 30.10 Detected concealed targets.

Only visible targets in the figure are O1 and O2. Apparently, from Figure 30.8, there are no clues about these vehicles and their corresponding locations.

Similarly, Figure 30.9(a) shows gray-scaled images resulting from applying CADCA to Figure 30.8 where seven bands (band numbers: 16, 53, 56, 60, 63, 82, and 96) were also selected by BBOPC using a two-layer pyramid and threshold $\varepsilon = 0.0001$ for band ratio transformation. The resulting $7 \cdot 6 = 42$ band-ratioed images were used for ATGP to detect and generate 24 target signatures for classification. Like Figure 30.7, most of the detected targets in Figure 30.8 were uninteresting or unwanted background signatures. As shown in Figure 30.9, V1 was detected in Figure 30.9(a) while V2 and V3 were detected in Figure 30.10(c). Further, V4 and V5 were also detected in Figure 30.9(b) and (d), respectively (see bright spots within the two objects, O1 and O2).

Figure 30.10 shows thresholded binary images of Figure 30.10(a–d) obtained by WTA where V1, {V2,V3}, V4, and V5 were detected in Figure 30.10(a), 30.10(c), 30.10(b), and 30.11(d), respectively.

As noted, V2 and V3 are different vehicles but were classified as the same type of vehicles because their spectra were very close. Furthermore, despite the fact that V4 and V5 are of the same type of vehicles, they were detected and classified in two different images because the two objects O1 and O2 used to cover these two vehicles are made by different materials. These results were interesting. Unlike V1, V2 and V3 which are shaded by trees, V4 and V5 are hidden underneath O1 and O2.

30.7 Conclusions

This chapter presents a new application of fully abundance-constrained LSMA to subpixel target size estimation. The idea is to apply an unsupervised target detection algorithm, ATGP, to find subpixel targets of interest, and then implement FCLS to estimate the abundance fractions of subpixel targets present in the image, and finally use the obtained abundance fractions to calculate their sizes. For such an approach to be effective, an accurate estimate of abundance fraction for a subpixel target is required. In this case, a fully constrained abundance LSMA such as FCLS is implemented for this purpose. Despite that abundance-constrained linear unmixing has been studied extensively for material quantification, the issue of subpixel target size estimation investigated in this chapter has never been explored in the past. Four LSMA methods are used for validation, which are an unconstrained method, LSOSP, two partially constrained least-squares methods, SCLS, NCLS, and a fully constrained method, FCLS. As demonstrated in simulated and real image experiments, the need of the fully abundance-constrained methods is evident when the target size is smaller than GSD. The target estimation error is increased as the target size is decreased. In addition to subpixel target size estimation this chapter also explores another application, the problem of concealed target detection, and further develops a computer automated method for detecting and classifying unknown concealed targets, to be called computer-aided detection and classification algorithm (CADCA). Since the targets of interest are either shaded by natural background or hidden underneath man-made objects, a band ratio transformation is used to reduce the effects caused by topographic aspects or differential illumination. In order to select appropriate band images to be used for band ratio, a BBOPC approach is also proposed for this purpose. Since the unknown concealed targets is now uncovered by the custom-designed band ratio transformation, an unsupervised target detection algorithm, ATGP is readily applied. It is worth noting that the criterion for ATGP is based on orthogonal subspace projection, which is also used for BBOPC. Such consistent design principle allows CADCA to achieve its best possible overall performance.

31

Nonlinear Dimensionality Expansion to Multispectral Imagery

Hyperspectral imaging sensors have been around more than two decades. Interestingly, there is no cut-and-dried definition available in the literature to differentiate hyperspectral imagery from multispectral imagery. A general understanding of distinction between these two is that a hyperspectral image is acquired by hundreds of "contiguous" spectral channels/bands with very fine spectral resolution, while a multispectral image is collected by tens of "discrete" spectral channels/bands with low spectral resolution. If this interpretation is used, we then run into a dilemma, "how many spectral channels are sufficiently enough for a remotely sensed image to be called a hyperspectral image?" or "how fine the spectral resolution should be for a remote sensing image to be considered as a hyperspectral image?" For example, if we take a small set of hyperspectral band images with spectral resolution 10 nm, say five band images, to form a five-dimensional image cube, do we still consider this newly formed five-dimensional image cube as a hyperspectral image or simply a multispectral image? If we adopt the former definition based on the number of bands, this five-dimensional image cube should be viewed as a multispectral image. On the other hand, if we adopt the latter definition based on spectral resolution, the five-dimensional image cube should be considered as a hyperspectral image. So, which one is correct? Thus far, it seems that there is no general consensus to settle this issue. This chapter makes an attempt to address this issue from a perspective of linear spectral mixture analysis (LSMA) via the pigeon-hole principle described in Chapter 1 so that a multispectral imaging (MSI) can be explored in such a way that hyperspectral imaging (HSI) can also be applicable to multispectral imagery. Two approaches are introduced in this chapter for such an exploration, both of which can be considered as reverse operations of data dimensionality reduction discussed in Chapter 6. One is band expansion process (BEP) originated from the band generation process (BGP) proposed by Ren and Chang (2000a), which can be viewed as a reverse process of dimensionality reduction by band selection (DRBS) in Chapter 6. The resulting LSMA is called band dimesnionality expansion (BDE)-based LSMA. The other is feature dimensionality expansion (FDE) derived from the kernel-based approaches discussed in Chapters 2 and 15. The resulting LSMA is then called kernel-based LSMA.

31.1 Introduction

It is known that a hyperspectral imager can discriminate and quantify materials more effectively via much better spectral resolution than a multispectral imager can. However, an interesting issue

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

seems to be overlooked and has never been addressed, “how to define and differentiate hyperspectral imagery from multispectral imagery”. Until we can settle this issue, the algorithm design for hyperspectral imagery cannot be effective. This has been the case over the past years where hyperspectral imagery has been considered and viewed as a natural extension of multispectral imagery under a common sense that hyperspectral imagery has more spectral bands with finer resolutions than multispectral imagery with low spectral resolution. With this intuitive generalization, in early days a general approach to designing HSI algorithms has been the one that extends algorithms developed for multipsectral imagery in a straightforward fashion. One of such techniques is the maximum likelihood-based classification and estimation (Landgrebe, 2003). Unfortunately, using this multispectral-to-hyperspectral extension may not be a best way to design algorithms for hyperspectral image analysis as already discussed and addressed in Chapter 1 (Section 1.2). We believe that one of main causes may be due to the fact that there is no specific criterion or definition that can be used to distinguish a hyperspectral image from a multispectral image in a rigorous and mathematical means. Because of that, we do not know how to take advantage of benefits provided by hyperspectral imagery, which cannot be found in multispectral imagery so that these advantages can be best utilized in designing and developing HSI algorithms. This chapter investigates this issue by exploring the connection between HSI and MSI techniques.

Assume that there are p spectral signatures in a remote sensing image with L being the total number of spectral channels used for data acquisition and collection. From a linear spectral mixture analysis (LSMA) point of view, LSMA is performed by solving a linear system consisting of L equations specified by L spectral channels/bands and p unknowns corresponding to p image endmembers. When $L > p$, the system is an over-determined system, in which case the considered remote sensing image is defined as a *hyperspectral* image. Conversely, if $L < p$, the system is an underdetermined system, in which case we can define the considered remote sensing image as a *multispectral* image. More specifically, if we consider that there are the p image endmembers to serve as a set of p basis vectors for L equations, LSMA with $L > p$ is called under-complete linear spectral mixture analysis (UC-LSMA) due to the insufficient number of basis vectors to represent the data. Conversely, if $L < p$, LSMA is called over-complete linear spectral mixture analysis (OC-LSMA) because there are more basis vectors than what we need to represent the data. At the first glimpse, the way that UC-LSMA and OC-LSMA are defined above seems to be out of reach and not intuitive. However, the following rationales should provide the ground to support the above definitions.

First, the relationship between L and p needs to be explored. On one hand, L is the number of spectral channels/bands and determines how many equations required to be used. On the other hand, p is the number of image endmembers (considered as signal sources) and determines how many unknown signal sources resident in the data to be solved via the L equations. By virtue of the pigeon-hole principle described in Chapter 1 (Section 1.3), one spectral channel/band is represented by an equation and can be viewed as a pigeon-hole to characterize, specify, and accommodate one spectral distinct signal source, that is, image endmember, which can be considered as a pigeon. For UC-LSMA where L is greater than p, it means that more pigeon-holes than pigeons can be used for pigeon accommodation. Obviously, there are $\binom{L}{p} = \frac{L!}{p!(L-p)!}$ ways to use one pigeon-hole to accommodate p pigeons, one pigeon-hole for one pigeon. On the other hand, for OC-LSMA where L is less than p, there are more pigeons than pigeon holes. So, in this case, more pigeons fly into fewer pigeon holes in which case at least one pigeon hole must accommodate more than one pigeon. When it occurs, all the pigeons in a single pigeon-hole cannot be separated and discriminated one from another. Interestingly, similar definitions to UC-LSMA and OC-LSMA can also be found in independent component analysis (ICA) (Hyvarinen et al., 2001) where an under-complete ICA (UC-ICA) is developed to blindly separate p statistically independent signal sources using L data samples with $L > p$, while an over-complete ICA (OC-ICA) is to blindly separate p statistically independent signal sources using L data samples with $L < p$. These two definitions provide a base of how UC-LSMA and OC-LSMA are defined as above. More specifically, UC-LSMA is developed for a hyperspectral image, which solves an overdetermined system with no solutions. To address this issue, one commonly used technique is dimensionality reduction described in Chapter 6, which reduces an over-determined system to a solvable system that is, $L = p$ that can produce a solution. In contrast, OC-LSMA is generally developed for a multispectral image which solves an underdetermined system with many solutions. In this case, OC-LSMA needs dimensionality expansion to augment the system to find a best solution in some sense of optimality with $L = p$. Accordingly, UC-LSMA and OC-LSMA essentially deal with completely opposite problems.

Over the past years, we have seen many efforts devoted to expanding MSI techniques to solve HSI problems (Richards and Jia, 1999; Landgrebe, 2003) and have achieved some success. However, there is little work, which does the other way around by taking advantage of HSI techniques to solve MSI problems. In doing so, we need to resolve the issue of OC-LSMA used for MSI which is an insufficient number of spectral bands. This chapter developed two approaches based on a concept of dimensionality expansion. One is band dimensionality expansion (BDE) that expands original data dimensionality by creating new spectral band images via nonlinear functions. It is originated from the band generation process (BGP) proposed by Ren and Chang (2000a) and can be considered as band expansion process (BEP), which is essentially a reverse process of dimensionality reduction by band selection (DRBS) presented in Chapter 6. The other approach is feature dimensionality expansion (FDE) via nonlinear kernels discussed in Chapter 15. However, it should be noted that unlike BEP/BDE, FDE actually expands features extracted from the original data space into a higher dimensional feature space via nonlinear kernels. Such FDE is quite different from BDE, which produces new spectral bands generated from the original spectral bands by nonlinear functions. Nevertheless, both BDE and FDE are developed to resolve the same issue caused by OC-LSMA, that is, there are no sufficient spectral channels/bands in a remote sensing image cube to be used to accommodate more signal sources that the image can handle. Furthermore, BDE and FDE also share similar ideas in the sense that the dimensionality expansion in FDE and BDE must be carried out by nonlinear functions. In other words, the use of nonlinear functions in the BDE and nonlinear kernels in FDE is a key to success in making hyperspectral imaging techniques applicable to multispectral imagery because OC-LSMA is linear.

31.2 Band Dimensionality Expansion

As shown in Chang and Brumbley (1999a, 1999b), the performance of OSP was considerably degraded if it was used for multispectral image classification due to an insufficient number of spectral bands to be used for orthogonal subspace projection. In order to address this issue, a recent effort in extending OSP to multispectral imagery was investigated by Ren and Chang (2000a) where an extended version of OSP, called Generalized OSP (GOSP), was developed by including a new technique, referred to as BGP, which allows users to expand spectral band dimensionality via nonlinear functions.

31.2.1 Rationale for Developing BDE

The idea of the BGP developed in Ren and Chang (2000a) arises from a second-order random process specified by the first-order and second-order statistics. If we view the original spectral bands image as the first-order statistical images, we can generate a set of second-order statistical spectral bands that capture correlation between spectral bands. These correlated images provide useful second-order statistical information among bands, which is missing in the set of the original spectral bands. The desired second-order statistics including auto-correlation, cross-correlation, and nonlinear correlation can be used to create nonlinearly correlated images. If there is a need of statistics of high orders, the same process can be carried out for this purpose. The concept of producing second-order or high-order correlated spectral bands coincides with moment generating functions used to generate various orders of moments for a random process. Despite that such a BDE may not have real physical meaning; it does provide a significant advantage to cope with the issue of the insufficient number of spectral band images. To shed light on this idea, we consider the following two examples.

As the first example, a farmer would like to give his own 17 cows to his three sons according to proportions of 17 cows, 1/2, 1/3, and 1/9. Obviously, there is no way to do so by simple arithemitics. However, if we introduce one hypothetic cow into 17 cows to make up a total of 18 cows, the problem is the well-solved because 1/2, 1/3, and 1/9 of 18 cows are 9 cows, 6 cows, and 2 cows, which are summed up exactly 17 cows. This simple example illustrates the idea of introduction of new band images by the BEP.

Another good example, which is more technical, is ternary Huffman coding from information theory. Assume that there is an information source X with six source alphabets {*a*,*b*,*c*,*d*,*e*,*f*} specified by their respective probabilities given by $\{0.3,\ 0.25,\ 0.2,\ 0.15,\ 0.05,\ 0.05\}$. The source entropy is

$$\begin{aligned} H(X) = -[&0.3 \log 0.3 + 0.25 \log 0.25 + 0.2 \log 0.2 + 0.15 \log 0.15 \\ &+0.05 \log 0.05 + 0.05 \log 0.05] \end{aligned} \tag{31.1}$$

Suppose that these source alphabets are encoded by three source code word alphabets denoted by $\{0, 1, 2\}$. What is the optimal Huffman code for {*a*,*b*,*c*,*d*,*e*,*f*}? Since the number of source alphabets is even, there is an unused node in a ternary Huffman coding tree. The best way is to leave this unused node as a leave rather than an internal node so that this unused node will produce the longest codeword as shown in Figure 31.1 rather than at the first internal node with shortest coding length shown in Figure 31.2.

To achieve the desired code in Figure 31.1, we create a hypothetic source alphabet by introducing a dummy source alphabet x in Figure 31.1 so that this dummy alphabet x will be treated as a

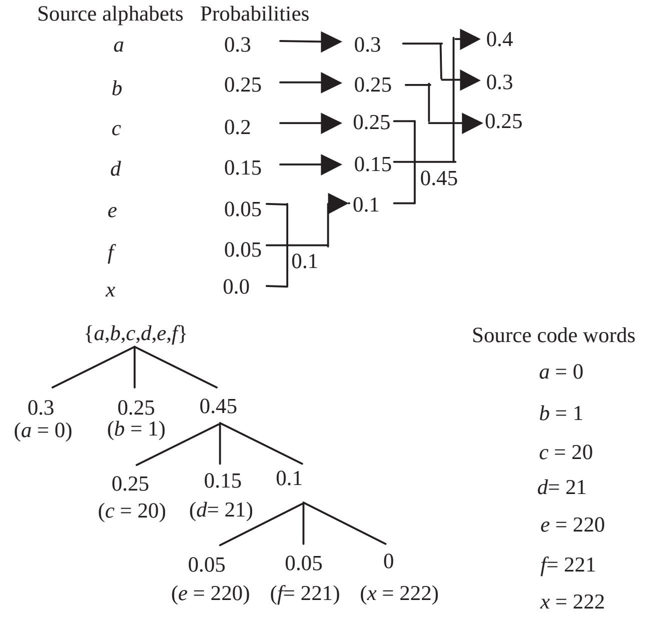


Figure 31.1 Huffman coding with an introduced dummy source alphabet x.

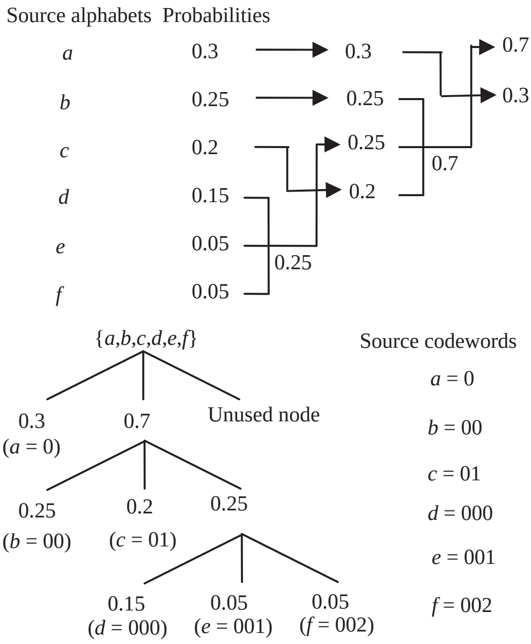


Figure 31.2 Huffman coding without introducing a dummy source alphabet x.

source alphabet with zero probability, thus it will have longest source code word, which does not exist. This idea is similar to the previous cow example described above where a hypothetic cow was introduced to make 17 cows a total of 18 cows so that the various proportions can be calculated based on the 18 cows. In the end such introduced hypothetic cow or dummy alphabet only serves as part of calculation but will not be counted for real.

31.2.2 Band Expansion Process

BEP presented in this section is essentially the same as BGP. It is renamed to reflect more accurately what the process does. Nevertheless, these two terminologies can be used interchangeably.

Let $\{B_l\}_{l=1}^{L}$ be the set of all original spectral band images. The first set of second-order-statistics spectral band images is generated based on autocorrelation. They are constructed by multiplying each individual spectral band image by itself, that is, $\left\{B_l^2\right\}_{l=1}^{L}$. A second set of second-order-statistical spectral band images are made up of all cross-correlated spectral band images, which are produced by correlating any pair of two different spectral band images, that is, $\{B_lB_k\}_{l=1,k=1,l\neq k}^{L,L}$. Adding these two sets of second-order-statistics spectral band images to $\{B_l\}_{l=1}^{L}$ produces a total of $L+L+\binom{L}{2}=\frac{L^2+3L}{2}$ spectral band images. If more spectral band images are needed, other nonlinear functions may also be used to generate the so-called nonlinear correlated spectral band images. For example, we calculate third-order, fourth-order auto- or cross-correlated spectral band images or use a square-root function to produce $\left\{\sqrt{B_l}\right\}_{l=1}^{L}$ or a logarithm function to produce $\{\log(B_l)\}_{l=1}^{L}$ to stretch out lower gray level values. In what follows, we describe several ways to generate second-order, third-order, fourth-order auto- and cross-correlated spectral band images and some nonlinear correlated spectral band images.

Algorithm for Band Expansion Process

Step 1. First-order band image: $\{B_l\}_{l=1}^{L}$ = set of original spectral band images.
Step 2. Second-order correlated spectral band images:

i. $\{B_l^2\}_{l=1}^{L}$ = set of auto-correlated spectral band images.

ii. $\{B_kB_l\}_{k=1,l=1,k\neq l}^{L,L}$ = set of cross-correlated spectral band images.

In case a re-scaling is needed, auto- or cross-correlated spectral band images can be normalized by the variances of spectral band images such as $\left(\sigma_{B_l}^2\right)^{-1}\{B_l^2\}$ and $(\sigma_{B_k}\sigma_{B_l})^{-1}\{B_kB_l\}$.

Step 3. Third-order correlated spectral band images:

i. $\{B_l^3\}_{l=1}^{L}$ = set of auto-correlated spectral band images.

ii. $\{B_k^2B_l\}_{k=1,l=1,l\neq k}^{L,L}$ = set of two cross-correlated spectral band images.

iii. $\{B_kB_lB_m\}_{k=1,l=1,m=1,k\neq l\neq m}^{L,L,L}$ = set of three cross-correlated spectral band images.

Similarly, like Step 2, if a rescaling is needed, auto- or cross-correlated spectral band images can be normalized by the variances of spectral band images such as $\left(\sigma_{B_l}^3\right)^{-1}\{B_l^3\}$, $\left(\sigma_{B_k}^2\sigma_{B_l}\right)^{-1}\{B_k^2B_l\}$, and $(\sigma_{B_k}\sigma_{B_l}\sigma_{B_m})^{-1}\{B_kB_lB_m\}$.

Step 4. Nonlinear correlated spectral band images:

i. $\{\sqrt{B_l}\}_{l=1}^{L}$ = set of spectral band images stretched out by the square-root.

ii. $\{\log(B_l)\}_{l=1}^{L}$ = set of spectral band images stretched out by the logarithmic function.

It is worth noting that all the spectral band images generated by BEP are produced nonlinearly. These spectral band images should offer useful information for data analysis because they provide useful nonlinear spectral information to help to improve performance. However, we should point out that according to our experience, using the cross-correlated spectral band images generated by Step 2(ii) is generally sufficient to accommodate the need of BEP. Additionally, using the set of auto-correlated spectral band images produced by Step 2(i) may sometimes cause nonsingularity problems in matrix computation because they are self-correlated and usually very close to the original spectral band images. It is suggested that they should not be used alone and can only be used in conjunction with cross-correlated spectral band images. This can be well explained by the fact that a covariance matrix including variances and co-variances provides more information than a diagonal matrix, which only includes variances.

31.3 Hyperspectral Imaging Techniques Expanded by BDE

In this section, three well-known HSI techniques, orthogonal subspace projection (OSP), constrained energy minimization (CEM), and RX-detector (RXD) described in Chapter 12 are considered as candidates to be expanded by BDE as MSI techniques. This is because each of these three techniques requires a different level of target knowledge. For OSP to work effectively, the complete knowledge of image endmembers including background must be provided *a priori*. Since such full target knowledge required by OSP is nearly impossible to obtain, specifically for image background, CEM is then developed to cope with this dilemma where only targets of interest are required to know in advance while the complete image background can be discarded. When obtaining partial knowledge of interesting targets becomes infeasible, RXD may be used to serve as this purpose for anomaly detection. In what follows, OSP, CEM, and RXD will be generalized to BEP-OSP, BEP-CEM, and BEP-RXD by including BEP as a preprocessing for BDE.

31.3.1 BEP-Based Orthogonal Subspace Projection

The BEP-based OSP introduced in this section operates in two phases. The first phase implements BEP to create additional new spectral bands from the original spectral bands. The objective of BEP is to make use of nonlinear correlation functions to produce a new set of second-order statistical bands. It is then followed by the second phase carried out by OSP. The procedure to implement the BEP-based orthogonal subspace projection (BEP-OSP) is summarized as follows:

Phase (1) Band Generation Process

i. Produce a set of second-order correlated spectral bands using auto-correlation or cross-correlation functions.
ii. Produce a set of nonlinearly correlated spectral bands using nonlinear functions, square root, or logarithm functions.
iii. Form a new set of bands by including both original spectral bands and spectral bands generated by steps (i) and (ii).

Phase (2) OSP

i. Apply OSP to the new set of spectral bands produced by BEP.

31.3.2 BEP-Based Constrained Energy Minimization

In analogy with BEP-OSP, CEM was also extended by the concept of BEP to BEP-based constrained energy minimization (BEP-CEM) in Chang et al. (2000).

Phase (1) Band Generation Process

i. Produce a set of second-order correlated spectral bands using autocorrelation or cross-correlation functions.
ii. Produce a set of nonlinearly correlated spectral bands using nonlinear functions, square root, or logarithm functions.
iii. Form a new set of bands by including both original spectral bands and spectral bands generated by step (i) and (ii).

Phase (2) CEM

i. Apply CEM to the new set of bands produced by BEP.

31.3.3 BEP-Based RX-Detector

Despite the fact that using BEP to develop BEP-OSP and BEP-CEM was investigated in Ren and Chang (2000a) and Chang et al. (2000), respectively, it is interesting to note that the concept of using BEP as BDE for RXD to derive a BEP-Based RX-detector (BEP-RXD) has never been explored. In what follows, we summarize its implementation.

Phase (1) Band Generation Process

i. Produce a set of second-order correlated spectral bands using auto-correlation or cross-correlation functions.
ii. Produce a set of nonlinearly correlated spectral bands using nonlinear functions, square root, or logarithm functions.
iii. Form a new set of bands by including both original spectral bands and spectral bands generated by Steps (i) and (ii).

Phase (2) RXD

i. Apply RXD to the new set of bands produced by BEP.

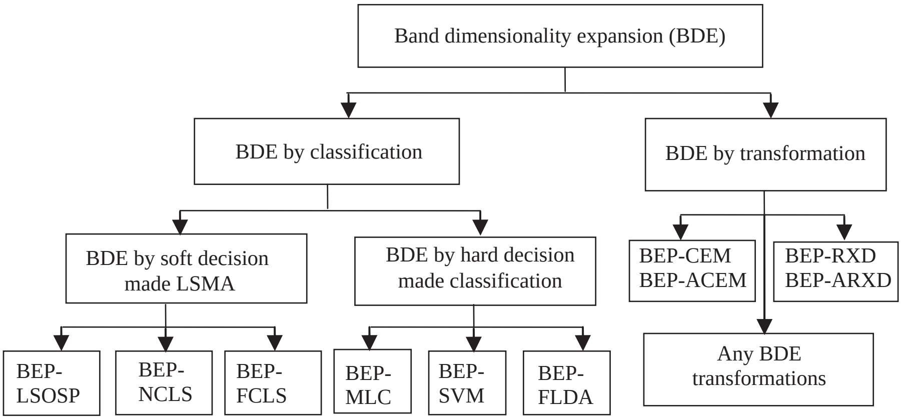


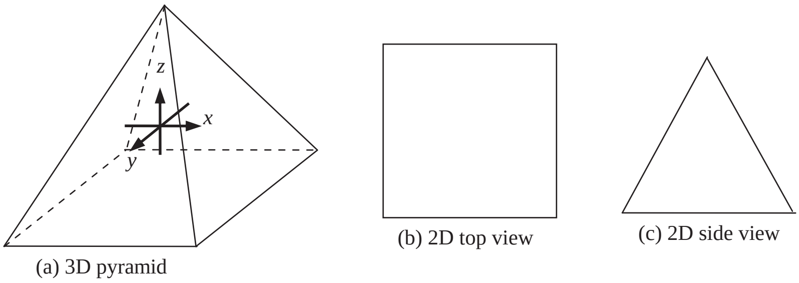
Figure 31.3 Band dimensionality expansion techniques for multispectral imagery.

One note on BEP-RXD is worth being mentioned. In addition to CEM and RXD, adaptive RXDs (ARXD) and adaptive CEM (ACEM), both of which using adaptive window sizes to form local covariance/covariance matrices to expand BEP to their corresponding generalized versions in exactly the same way as BEP-RXD is done, referred to as BEP-ARXD and BEP-ACEM. Figure 31.3 summarizes the utility of BDE in various operators and transformations where BEP-MLC is a BEP-based maximum likelihood classifier, which operates MLC on the BEP-expanded data space.

31.4 Feature Dimensionality Expansion by Nonlinear Kernels

Unlike BDE, which expands band dimensionality to resolve the issue in use of an insufficient number of spectral bands, FDE makes a rather different attempt by expanding features used by classification to a high dimensional feature space via a nonlinear kernel so that the linear nonseparability issue arising in the original data space can be resolved in a new feature space. Its idea can be briefly illustrated by the following example. Assume that a three-dimensional (3D) pyramid shown in Figure 31.5(a) is viewed in a two-dimensional space. It must require two views, a 2D top view shown in Figure 31.5(b) and a 2D side view shown in Figure 31.5(c) to completely envision what a pyramid looks like in a 3D space.

Using the pyramid in Figure 31.4(a) as an example, solving a linear nonseparable problem in an original data space is similar to viewing a pyramid in a two-dimensional data space, which requires two different views, top and side views, to capture its shape, which can only be visualized in a three-dimensional space. This is equivalent to expanding two top and side views to a single view of a pyramid in a three-dimesnional space shown in Figure 31.4(a), which becomes linearly separable problem.



(a) 3D pyramid

(b) 2D top view

(c) 2D side view

Figure 31.4 A three-dimensional pyramid can be viewed by jointly 2 two-dimensional top and side views.

Two kernel-based approaches are generally developed in the literature. One is FDE by transform, which utilizes a nonlinear kernel function to map a component analysis-based transformation such as PCA, ICA into a higher dimensional feature space so that nonlinear separable data features can be recognized by a linear classifier in this new feature space. The main hurdle of this approach is excessive computational complexity resulting from kernelization where all data sample vectors are considered as feature vectors to be kernelized. The second approach is FDE by classification, which only kernelizes the training sample vectors used to train classifiers such as SVM and FLDA. As a consequence, the issue of computational complexity caused by data sample vectors in FDE by transform is mitigated by computational complexity which is only involved with the training sample vectors. In what follows, these two approaches are presented in two separate subsections.

31.4.1 FDE by Transformation

Kernel-based approaches have found great success in many applications in the sense that feature vectors extracted from the original data space are nonlinearly transformed to a higher feature dimensional space to resolve the issue of linear nonseparable problems. This section presents a kernel version of the most widely used transformation, principal components analysis (PCA) developed by Scholkopf et al. (1999b).

For presenting, we reiterate the PCA described in Section 6.2.1.1. Assume that $\{\mathbf{r}_i\}_{i=1}^N$ is a set of L-dimensional image pixel vectors, and $\boldsymbol{\mu}$ is the mean of the sample pool S obtained by $\boldsymbol{\mu} = (1/N)\sum_{i=1}^N \mathbf{r}_i$. Let $\mathbf{X}$ be the sample data matrix formed by $\mathbf{X} = [\mathbf{r}_1\mathbf{r}_2 \cdots \mathbf{r}_N]$. Then, the sample covariance matrix of the sample pool S can be calculated by $\mathbf{K} = (1/N)\left[\mathbf{X}\mathbf{X}^T\right] = (1/N)\left[\sum_{i=1}^N (\mathbf{r}_i - \boldsymbol{\mu})(\mathbf{r}_i - \boldsymbol{\mu})^T\right]$. If we further assume that $\{\lambda_l\}_{l=1}^L$ is the set of eigenvalues obtained from the covariance matrix $\mathbf{K}$ and $\{\mathbf{v}_l\}_{l=1}^L$ are their corresponding unit eigenvectors, that is, $||\mathbf{v}_l|| = 1$, we can define a diagonal matrix $\mathbf{D}_\sigma$ with variances $\left\{\sigma_l^2\right\}_{l=1}^L$ along the diagonal line as

$$\mathbf{D}_\sigma = \begin{bmatrix} \sigma_1^2 & 0 & \mathbf{0} \\ 0 & \ddots & 0 \\ \mathbf{0} & 0 & \sigma_L^2 \end{bmatrix} \tag{31.2}$$

and an eigenvector matrix $\boldsymbol{\Lambda}$ specified by $\{\mathbf{v}_l\}_{l=1}^L$ as

$$\boldsymbol{\Lambda} = [\mathbf{v}_1\mathbf{v}_2 \cdots \mathbf{v}_L] \tag{31.3}$$

such that

$$\mathbf{D}_\sigma = \boldsymbol{\Lambda}^T\mathbf{K}\boldsymbol{\Lambda} \tag{31.4}$$

Using the eigenvector matrix $\boldsymbol{\Lambda}$, a linear transform $\xi_{\boldsymbol{\Lambda}}$ defined by

$$\xi_{\boldsymbol{\Lambda}}(\mathbf{r}) = \boldsymbol{\Lambda}^T\mathbf{r} \tag{31.5}$$

transforms every data sample $\mathbf{r}_i$ to a new data sample, $\tilde{\mathbf{r}}_i$ by

$$\tilde{\mathbf{r}}_i \equiv \boldsymbol{\Lambda}^T\mathbf{r}_i \tag{31.6}$$

As a result, the mean of new $\xi_{\boldsymbol{\Lambda}}$-transferred data samples $\{\tilde{\mathbf{r}}_i\}_{i=1}^N$ becomes $\tilde{\boldsymbol{\mu}} = (1/N)\sum_{i=1}^N \tilde{\mathbf{r}}_i$ and its resulting covariance matrix is reduced to a diagonal matrix given by

$$\tilde{\mathbf{K}} = (1/N)\sum_{i=1}^N (\tilde{\mathbf{r}}_i - \tilde{\boldsymbol{\mu}})(\tilde{\mathbf{r}}_i - \tilde{\boldsymbol{\mu}})^T = \boldsymbol{\Lambda}^T\mathbf{K}\boldsymbol{\Lambda} = \mathbf{D}_\sigma \tag{31.7}$$

Equation (31.7) implies that the $\xi_{\boldsymbol{\Lambda}}$-transferred data matrix $\tilde{\mathbf{X}} = [\tilde{\mathbf{r}}_1\tilde{\mathbf{r}}_2 \cdots \tilde{\mathbf{r}}_N]$ has been de-correlated or whitened by the matrix $\boldsymbol{\Lambda}$, which is referred to as a whitening matrix (Poor, 1994). The transform $\xi_{\boldsymbol{\Lambda}}$ defined by (31.6) is generally called principal component transform and the lth component of $\hat{\mathbf{X}}$ is formed by

$$\xi_{\mathbf{v}_l}(\mathbf{X}) = \mathbf{v}_l^T\mathbf{X} \tag{31.8}$$

and is called the lth principal component (PC) which consists of $\left\{\mathbf{v}_l^T\mathbf{r}_i\right\}_{i=1}^N$ that are $\xi_{\mathbf{v}_l}$-transferred data samples corresponding the lth eigenvalue λ_l. The PCA is a process that implements the transform $\xi_{\boldsymbol{\Lambda}}$ defined by (31.5) to obtain a set of PCs via (31.6) or (31.8) with all $1 \leq l \leq L$.

PCA is a process that implements the transform $\xi_{\boldsymbol{\Lambda}}$ defined by (31.5) to obtain a set of PCs via (31.6) or (31.8) with all $1 \leq l \leq L$. To achieve DR, only those PCs specified by eigenvectors that correspond to first q largest eigenvalues will be retained, while the PCs specified by eigenvectors corresponding to the remaining (L–q) smaller eigenvalues will be discarded. In what follows, we follow the approach proposed in Scholkopf et al. (1999b) to extend the PCA to the so-called Kernel PCA (K-PCA) via a nonlinear function.

Assume that the original data space $\mathbf{X}$ is made up of N L-dimensional data sample vectors $\{\mathbf{r}_i\}_{i=1}^N$, $\mathbf{X} = \{\mathbf{r}_i\}_{i=1}^N$ and ϕ is a nonlinear function which maps the data space $\mathbf{X}$ into a feature space $\mathbf{F}$, with dimensionality yet to be determined by a kernel, that is,

$$\phi : \mathbf{X} \rightarrow \mathbf{F}. \tag{31.9}$$

We further assumed that the mapped data samples into the feature space are centered, that is, $\sum_{i=1}^N \phi(\mathbf{r}_i) = 0$. Now we can define the sample covariance matrix in the feature space $\mathbf{F}$ by the $\{\phi(\mathbf{r}_i)\}_{i=1}^N$

$$\bar{\mathbf{K}} = (1/N)\sum_{i=1}^N \phi(\mathbf{r}_i)\phi(\mathbf{r}_i)^T \tag{31.10}$$

Using the same argument outlined by (31.1–31.5), let $\bar{\lambda}_1 \geq \bar{\lambda}_2 \geq \cdots \geq \bar{\lambda}_L$ be the eigenvalues of $\bar{\mathbf{K}}$ specified by (31.10) and $\{\bar{\mathbf{v}}_l\}_{l=1}^L$ be their corresponding eigenvectors with unit length, that is, $\bar{\mathbf{v}}_l^T\bar{\mathbf{v}}_l = 1$. Then,

$$\bar{\mathbf{K}}\bar{\mathbf{v}}_l = \lambda_l\bar{\mathbf{v}}_l \tag{31.11}$$

Multiplying both sides of (31.11) by $\phi(\mathbf{r}_k)$ results in

$$\phi(\mathbf{r}_k)\bar{\mathbf{K}}\bar{\mathbf{v}}_l = \lambda_l\phi(\mathbf{r}_k)\bar{\mathbf{v}}_l \tag{31.12}$$

Since $\bar{\mathbf{v}}_l$ lies in the space linear spanned by the data samples in the space $\mathbf{F}$, $\{\phi(\mathbf{r}_i)\}_{i=1}^N$, it can expressed by

$$\bar{\mathbf{v}}_l = \sum_{i=1}^N \bar{\gamma}_{li}\phi(\mathbf{r}_i) \tag{31.13}$$

where $\bar{\gamma}_l = (\bar{\gamma}_{l1}, \bar{\gamma}_{l2}, \ldots, \bar{\gamma}_{lN})^T$is a combinitorial coefficient vector of $\bar{\mathbf{v}}_l$. Substituting (31.10) and (31.13) into (31.12) yields

$$N\bar{\lambda}_l\bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\gamma}}_l = \bar{\boldsymbol{\Phi}}^2\bar{\boldsymbol{\gamma}}_l \tag{31.14}$$

where

$$\bar{\Phi}_{ij} = \phi(\mathbf{r}_i)^T\phi(\mathbf{r}_j) = \phi(\mathbf{r}_i)\cdot\phi(\mathbf{r}_j) \tag{31.15}$$

To find a solution to (31.14), we only need to solve

$$N\bar{\lambda}_l\bar{\boldsymbol{\gamma}}_l = \bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\gamma}}_l \tag{31.16}$$

Using (31.16) and $\bar{\mathbf{v}}_l^T\bar{\mathbf{v}}_l = 1$ gives rise to

$$1 = \bar{\mathbf{v}}_l^T\bar{\mathbf{v}}_l = \sum_{i,j=1}^{N}\bar{\gamma}_{li}\bar{\gamma}_{lj}\big(\phi(\mathbf{r}_i)\cdot\phi(\mathbf{r}_j)\big) = \big(\bar{\boldsymbol{\gamma}}_l\cdot\bar{\boldsymbol{\Phi}}\bar{\boldsymbol{\gamma}}_l\big) = \bar{\lambda}_l(\bar{\boldsymbol{\gamma}}_l\cdot\bar{\boldsymbol{\gamma}}_l) \tag{31.17}$$

To find the lth principal component in the feature space $\mathbf{F}$, we project all data samples $\phi(\mathbf{r})$ in $\mathbf{F}$ onto the lth eigenvector $\bar{\mathbf{v}}_l$ via the following equation:

$$(\bar{\mathbf{v}}_l\cdot\phi(\mathbf{r})) = \sum_{i=1}^{N}\bar{\gamma}_{li}(\phi(\mathbf{r}_i)\cdot\phi(\mathbf{r})) \tag{31.18}$$

It should be noted that to calculate (31.15) and (31.18), we only need to perform dot products without actually using explicit forms of the nonlinear function ϕ, which is referred to as kernel trick in Section 2.4.1.

31.4.2 FDE by Classification

Two types of FDE by classification are considered, one using sample spectral correlation and the other using intrapixel spectral correlation.

31.4.2.1 FDE by Classification using Sample Spectral Correlation

The FDE by classification using sample spectral correlation take advantage of spectral correlation among sample vectors to improve and enhance classification. This type of classifiers includes hard decision-made classifiers such as MLC and soft decision-made detectors, RXD and CEM, all of which include either sample spectral covariance matrix or sample spectral correlation matrix as additional spectral information to improve their performance in solving non-linear separable problems. Their kernel counterparts are referred to as kernel-based MLC (KMLC), kernel-based FLDA (K-FLDA), kernel-based RXD (KRXD), and kernel-based CEM (KCEM). Since the sample spectral covariance/correlation matrix used by this type of classifiers for FDE is formed by all the data sample vectors, its size determines computational complexity, which turns out to be the same issue encountered in FDE by transform. However, an advantage of these classifiers used for FDE over FDE by transform is that these classifiers can be made adaptive classifiers by considering various window sizes to capture local spectral correlation to replace the global sample spectral correlation. As a result, the number of the sample vectors embraced by a local window is only limited to the window size which can significantly ease computational complexity. The resulting classifiers are referred to as Kernel-based adaptive MLC (KAMLC), kernel-based adaptive RXD (KARXD), Kernel-based adaptive CEM (KACEM).

Two interesting facts are noteworthy.

1. Unlike other classifiers used for FDE which use interpixel sample spectral covariance or sample correlation matrix, Fisher's linear discriminant analysis (FLDA) discussed in Chapter 2.3.1.1 uses

the interpixel within-class and between-class scatter matrices to perform classification. Since the between-class scatter matrix has only rank of $p-1$ where the p is the number of classes to be classified. This implies that the rank of Fisher's ratio is $p-1$. So, in this case, KFLDA discussed in Section 2.4.2 does not require a full rank to calculate the within-class scatter matrix. This fact significantly reduces computational complexity by using singular value decomposition (SVD) discussed in Chapter 6 in a similar manner that a local window is used to reduce the size of the sample covariance/correlation matrix.

2. It should be also noted that the kernel-based support vector machines (KSVM) in Section 2.4.3 can be considered as a special case of this type of FDE by classification since it does not use any interpixel sample spectral correlation.

31.4.2.2 FDE by Classification using Intrapixel Spectral Correlation

The FDE by classification using intrapixel spectral correlation has been discussed in Chapter 15 where three KLSMA-based classifiers, kernel-based orthogonal subspace projection (KOSP), kernel-based least-squares orthogonal subspace projection (KLSOSP), kernel-based non-negativity constrained least squares (KNCLS), and kernel-based fully constrained least squares (KFCLS) are derived. These classifiers make use of intrapixel spectral correlation provided by the signature matrix **M** in (2.78). Since these classifiers are operated on a single pixel vector basis, their kernel-based counterparts are determined by the number of signatures, p, used in the **M** not the data size and signatures used to unmix data sample vectors. Accordingly, the kernelization trick is actual performed on the classifiers themselves not on training samples or on data sample vectors. This type of FDE by classification is the third kernelization approach described in Section 15.2.4. As a consequence, their computational complexity is significantly reduced because the value of the p is very small compared to the data size. In this case, all the KLSMA classifiers developed in Chapter 15 are easy to implement.

Finally, Figure 31.5 summarizes two approaches to FDE where various transformations used in FDE by transform and various classifiers used by FDE by classifiers are described.

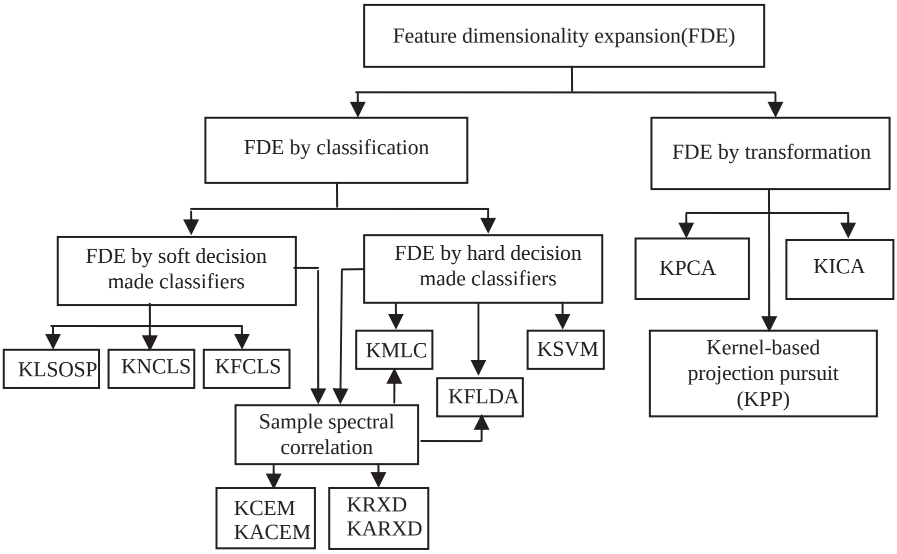

Feature dimensionality expansion(FDE)
FDE by classification
FDE by transformation
FDE by soft decision made classifiers
FDE by hard decision made classifiers
KPCA
KICA
KLSOSP
KNCLS
KFCLS
KMLC
KSVM
KFLDA
Kernel-based projection pursuit (KPP)
Sample spectral correlation
KCEM
KACEM
KRXD
KARXD

Figure 31.5 Feature dimensionality expansion techniques for multispectral imagery.

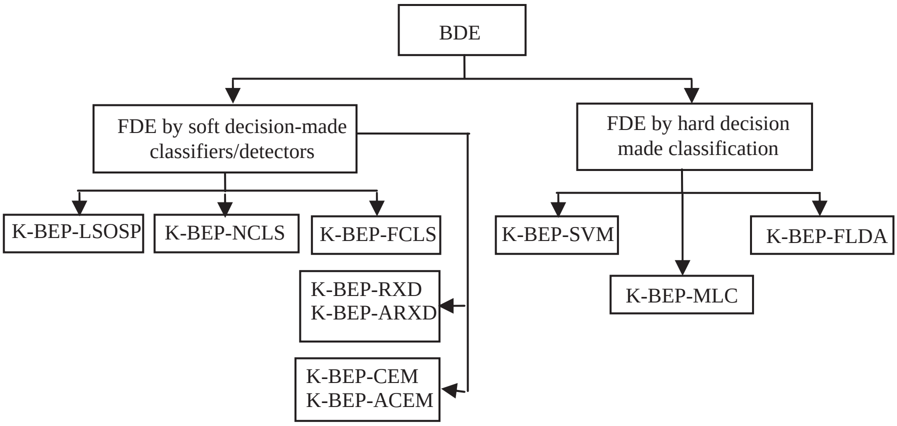


Figure 31.6 BDE in conjunction with FDE.

31.5 BDE in Conjunction with FDE

It is obvious that BDE can also be implemented in conjunction with FDE by taking advantages provided by both BDE and FDE. Figure 31.6 depicts various combinations of BDE with FDE.

31.6 Multispectral Image Experiments

The multispectral image data used for experiments is shown in Figure 31.7. The image scene was collected by the Satellite Pour l'Observation de la Terra (SPOT) system in three spectral bands, two of which are from visible region of electromagnetic spectrum, band 1: 0.5–0.59 μm and band 2: 0.61–0.68 μm and the third band is from near-infrared region of electromagnetic spectrum, 0.79–0.89 μm. The ground sampling distance is 20 m. The image scene has size 256×256 was taken over Northern Virginia where in the scene there are the Falls Church High School, the Little River Turnpike, a lake at the upper right corner and the Mill Creek Park.

According to the ground truth obtained from visiting the site as well as provided by the Google Earth, there are at least four signatures, buildings, roads/parking lots, water, and vegetation in the scene with training samples selected from the four marked areas shown in Figure 31.8 to represent these four different classes. The signatures used to form the signature matrix **M** for LSMA were

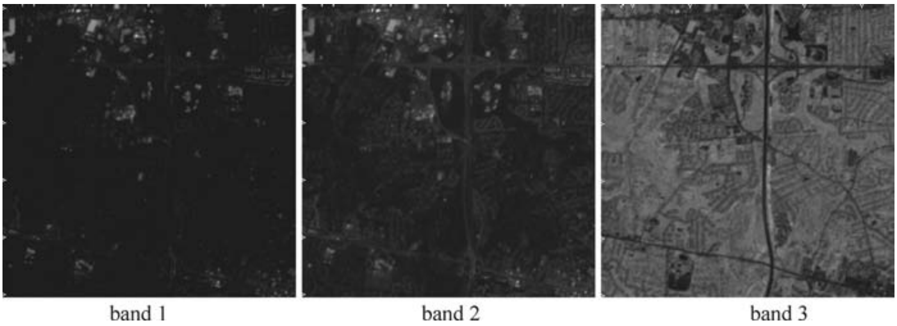


Figure 31.7 Three spectral band images of SPOT data.

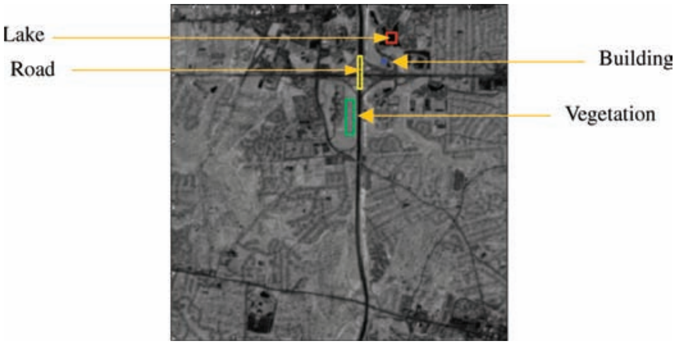
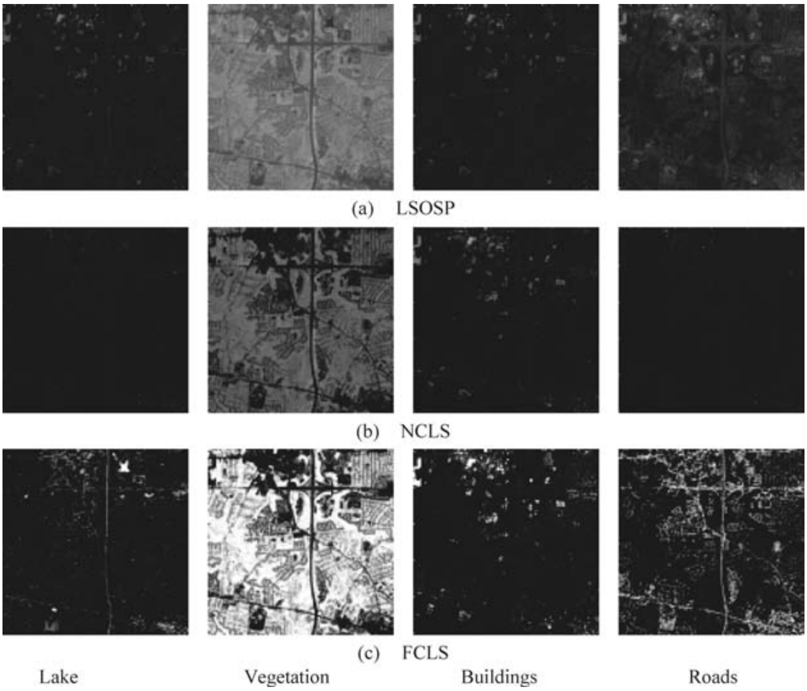


Figure 31.8 Training samples selected from four areas in Northern Virginia.

obtained by sample means of training samples from the four marked areas, that is, $\mathbf{M} = [\mathbf{m}_{\text{lake}}, \mathbf{m}_{\text{vegetation}}, \mathbf{m}_{\text{building}}, \mathbf{m}_{\text{roads}}]$ where $\mathbf{m}_{\text{area}}$ is the sample mean of the area.

To carry out BEP, two sets of BEP bands were used, autocorrelated bands and cross-correlated bands, where both autocorrelated bands and cross-correlated bands were normalized by their corresponding variances of spectral band images. Six experiments were conducted to evaluate the effectiveness of BEP and kernelization in conjunction with LSMA in performance analysis where LSMA, BEP6 + LSMA, BEP9 + LSMA, KLSMA, BEP6 + KLSMA, and BEP9 + KLSMA were


(a) LSOSP

(b) NCLS

(c) FCLS

Figure 31.9 Unmixed results of SPOT data produced by (a) LSOSP, (b) NCLS, and (c) FCLS.

used to perform spectral unmixing with BEP6 and BEP9 defined as

$$\begin{aligned} \text{BEP6} &= \text{original three spectral bands} + \text{three cross-correlated spectral bands} \\ \text{BEP9} &= \text{original three spectral bands} + \text{three cross-correlated spectral bands} \\ &\quad + \text{three auto-correlated spectral bands} \end{aligned}$$

Experiment 31.1 (LSMA)

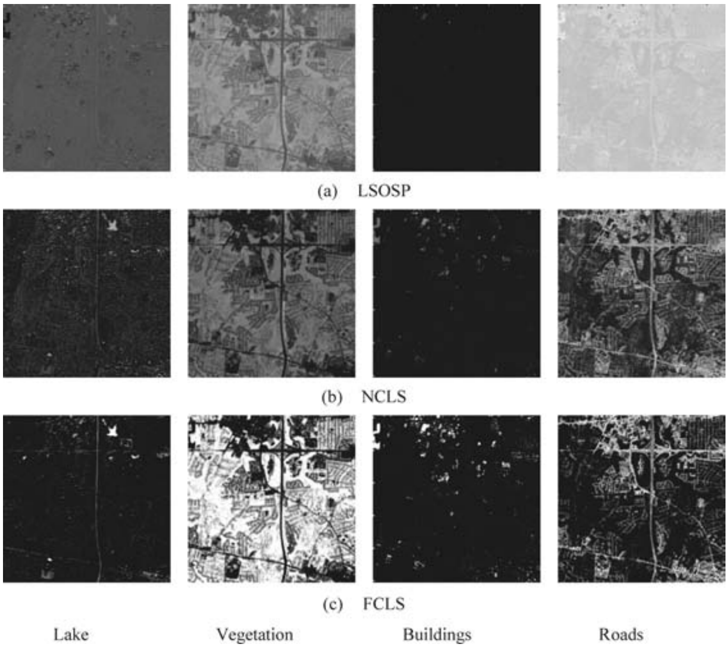
In this experiment three LSMA techniques, LSOSP, NCLS, and FCLS were operated on the three-band original SPOT data without nonlinear dimensionality expansion. Figure 31.9(a)–(c) shows the unmixed results of the SPOT data produced by LSOSP, NCLS, and FCLS, respectively, where FCLS obviously outperformed the other two, LSOSP and NCLS.

Experiment 31.2 (BEP6 + LSMA)

This experiment was conducted in the same way that Experiment 31.1 was performed except that LSOSP, NCLS, and FCLS were applied to BEP6 image data. Figure 31.10(a)–(c) shows the unmixed results produced by LSOSP, NCLS, and FCLS, respectively, where FCLS was still the best followed by NCLS as the second best with LSOISP being the worst. Compared to results shown in Figure 31.9, FCLS performed nearly the same with no obvious improvements, LSOSP was indeed slightly improved, but NCLS was improved drastically. This experiment shows that three LSMA-based techniques could be benefited from three extra cross-correlated spectral bands, which actually provided useful spectral information that could improve the unmixing ability.

Experiment 31.3 (BEP9 + LSMA)

The only difference between this experiment and Experiment 31.2 was the data which included three more auto-correlated spectral bands in addition to the three cross-correlated spectral bands. Figure 31.11(a)–(c) shows the unmixed results produced by LSOSP, NCLS, and FCLS, respectively.


(a) LSOSP

(b) NCLS

(c) FCLS

Figure 31.10 Unmixed results of SPOT data produced by (a) LSOSP, (b) NCLS, and (c) FCLS with using BEP 6.

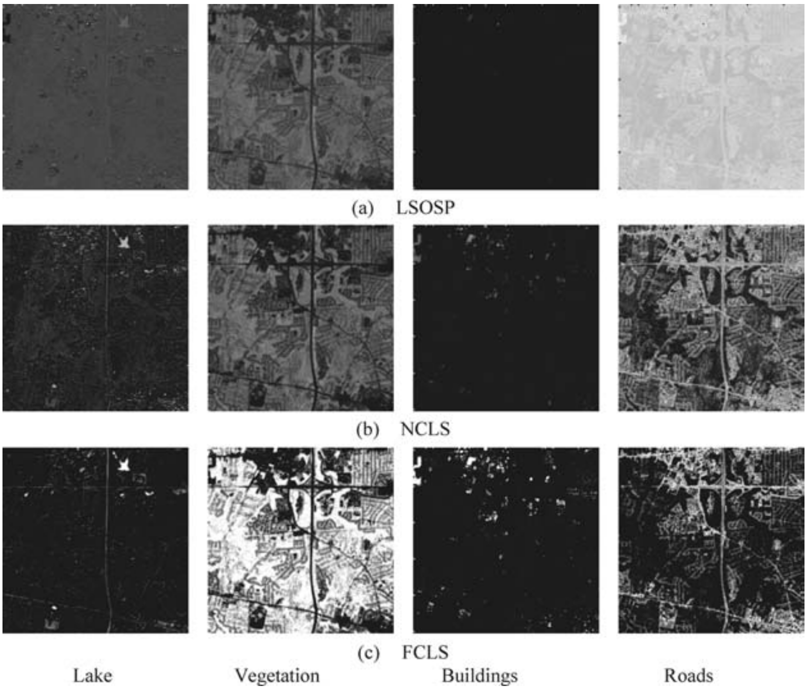


Figure 31.11 Unmixed results of SPOT data produced by (a) LSOSP, (b) NCLS, and (c) FCLS with using BEP 9.

Compared to the results in Figure 31.10, the improvement from BEP6 to BEP9 was very little and hardly recognized. This implies that including autocorrelated spectral bands did not really help much once cross-correlated spectral bands are used.

Experiment 31.4 (KLSMA)

This experiment was conducted to see how much improvement that LSMA could be benefited from kernelization where RBF kernel was used with three different values of σ selected empirically. Figures 31.12–31.14 show the unmixed results produced by KLSOSP, KNCLS, and KFCLS, respectively, where in all cases FCLS remained the best, but interestingly, LSOSP and NCLS performed very closely.

If we compare the results against their counterparts without kernelzation, the improvement resulting from kernelization was very significant, specifically, LSOSP.

In the previous four experiments, only one nonlinear dimensionality technique was implemented, either BEP or kernelization but not both. The following two experiments was performed to see whether or not unmixing ability can be further improved if both BEP and kernelization are used for nonlinear dimensionality expansion.

Experiment 31.5 (BEP6 + KLSMA)

This experiment implemented KLSMA on the image data produced by BEP6. Figures 31.15–31.17 show the unmixed results produced by KLSOSP, KNCLS, and KFCLS, respectively.

Compared to the results shown in Figs 31.12–31.14, the difference between “KLSMA without BEP” and “KLSMA with BEP6” was not very visible. In other words, the advantages provided by

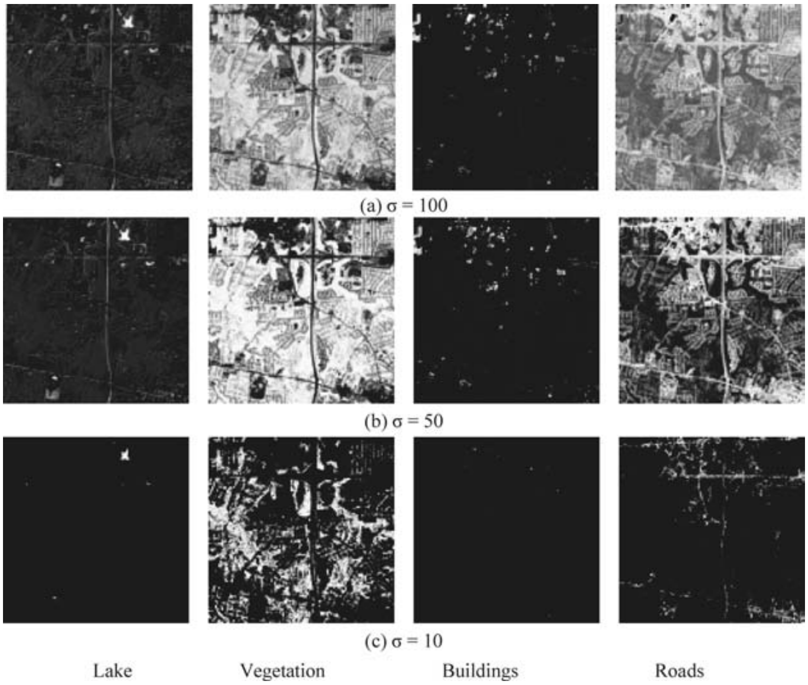


Figure 31.12 Unmixed results of SPOT data produced by KLSOSP with different values of σ.

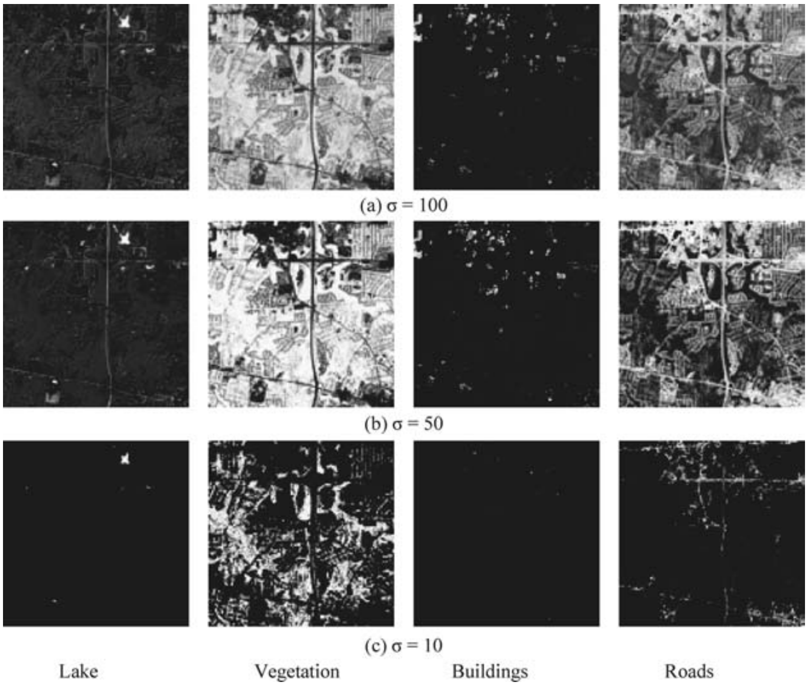


Figure 31.13 Unmixed results of SPOT data produced by KNCLS with different values of σ.

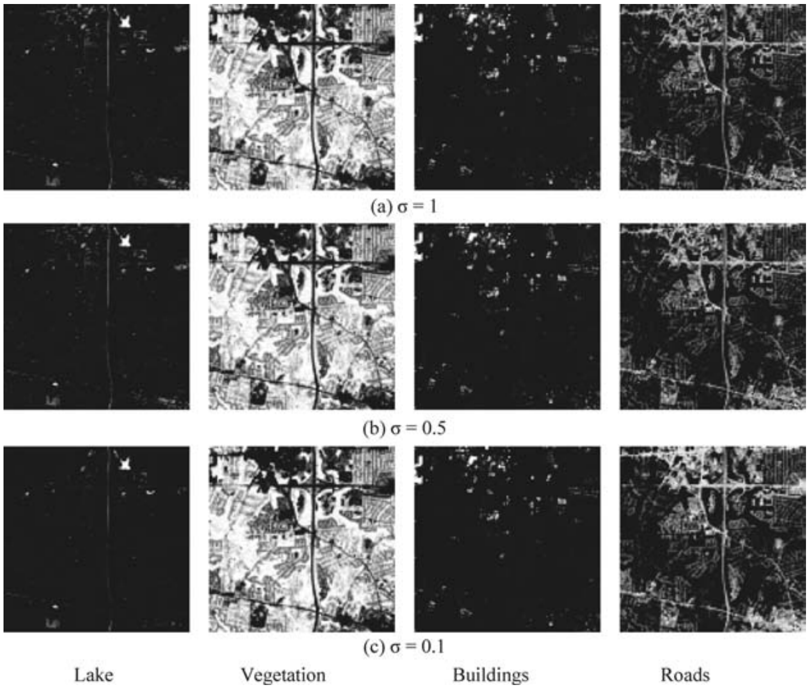


Figure 31.14 Unmixed results of SPOT data produced by KFCLS with different values of σ.

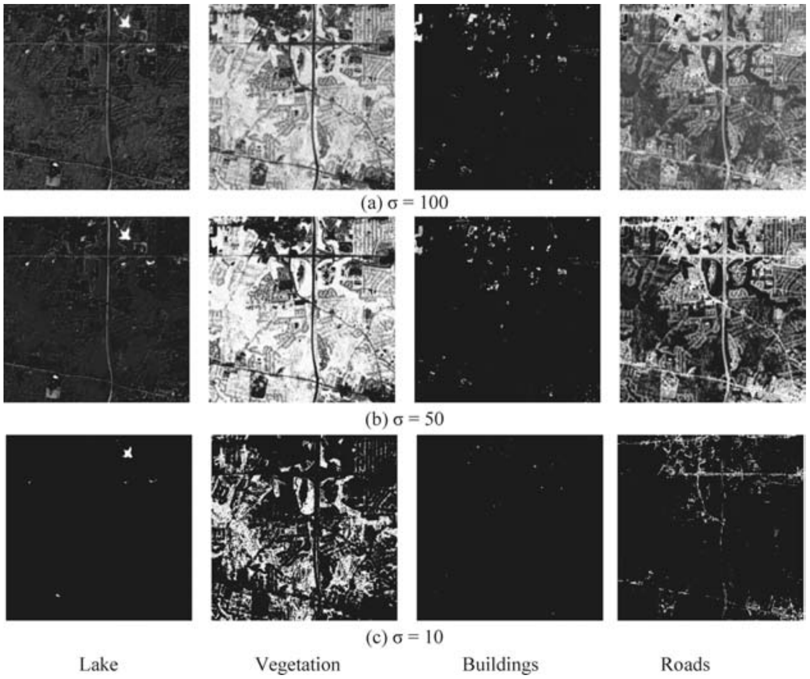
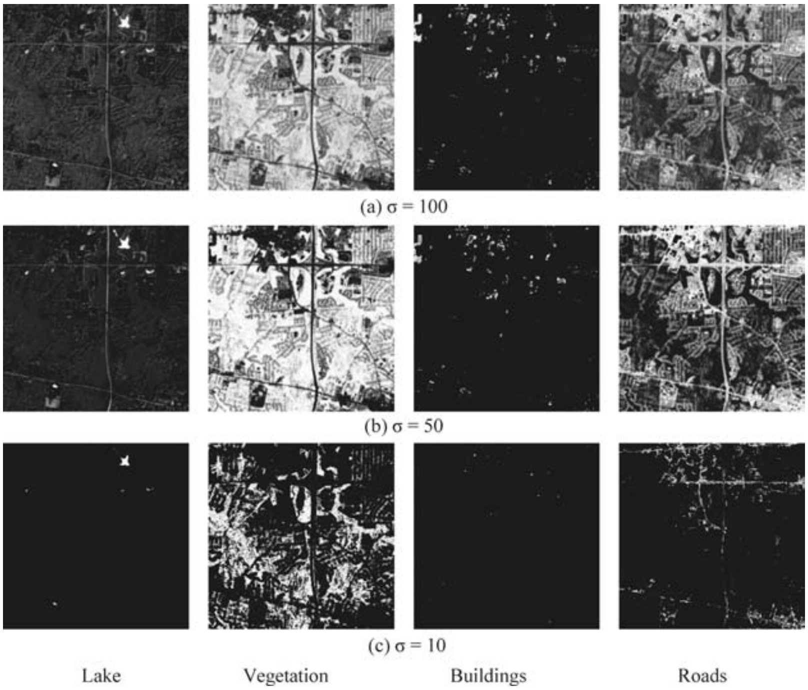


Figure 31.15 Unmixed results of SPOT data produced by KLSOSP using BEP6 with different values of σ.

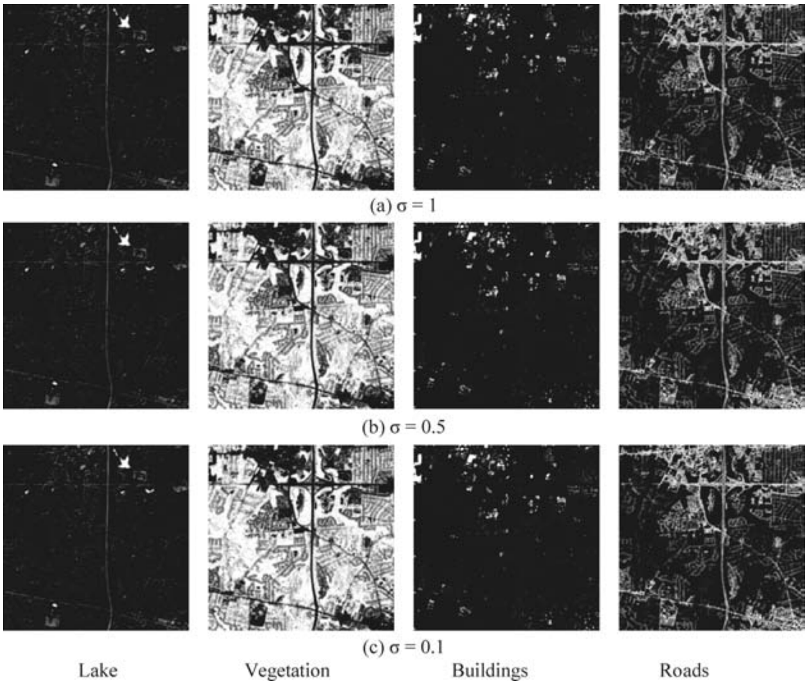


(a) σ = 100

(b) σ = 50

(c) σ = 10

Figure 31.16 Unmixed results of SPOT data produced by KNCLS using BEP6 with different values of σ.



(a) σ = 1

(b) σ = 0.5

(c) σ = 0.1

Figure 31.17 Unmixed results of SPOT data produced by KFCLS using BEP6 with different values of σ.

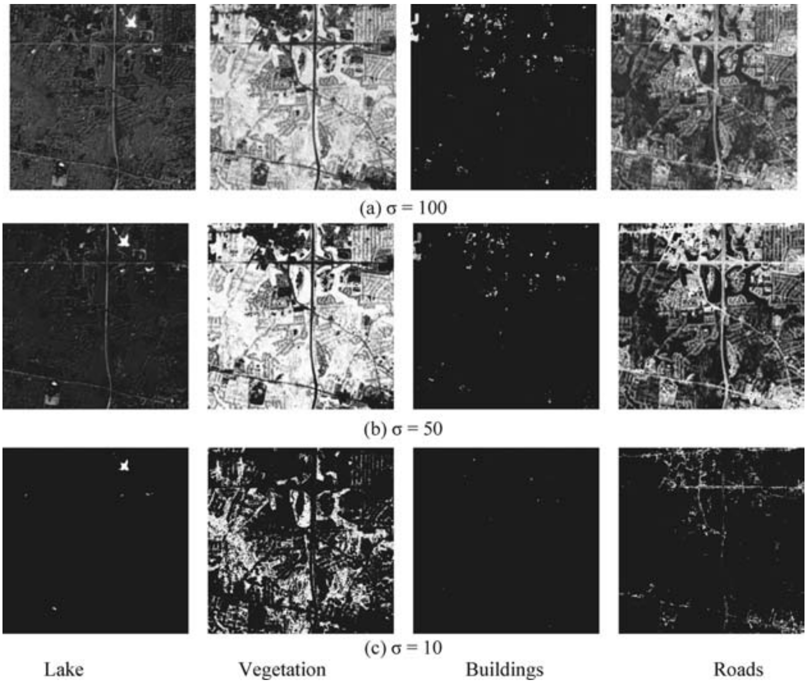
BEP seemed to vanish after kernelization. This indicated that kernelization is more effective than BEP when considering came spectral unmixing. This made sense because pixels in multispectral images are generally heavily mixed, and the pixel mixtures may not be linear either.

Experiment 31.6 (BEP9 + KLSMA)

This experiment was performed by taking both advantages, BEP9 and kernelization. Figures 31.18–31.20 show the unmixed results produced by KLSOSP, KNCLS, and KFCLS, respectively. As we can see by comparing the results to those in Experiment 31.5 BEP9 seemed not to provide any better benefit than BEP6 did, a similar phenomenon observed in Experiments 31.2 and 31.3 regardless of whether kernelization was used.

Based on the above six experiments, we can draw the following conclusions.

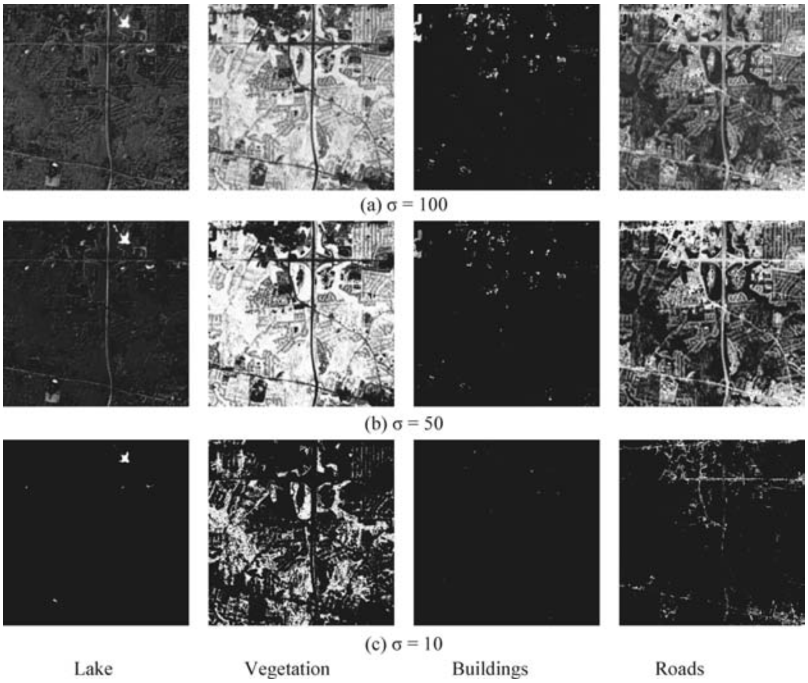
1. When LSMA is used, BEP could significantly improve unmixing performance.
2. Including cross-correlated spectral bands was sufficiently enough to implement BEP.
3. Kernelization could also significantly improve unmixing performance when the data dimensionality was small.
4. KLSMA seemed to outperform "BEP + LSMA," particularly in both LSOSP and NCLS.
5. Once kernelization was used, the advantage of using BEP vanished. This implied that the kernelization had better ability in improving spectral unmixing than BEP did. As a result, using kernelization in conjunction with BEP did not really provide much benefit in unmixing improvement.


(a) σ = 100

(b) σ = 50

(c) σ = 10

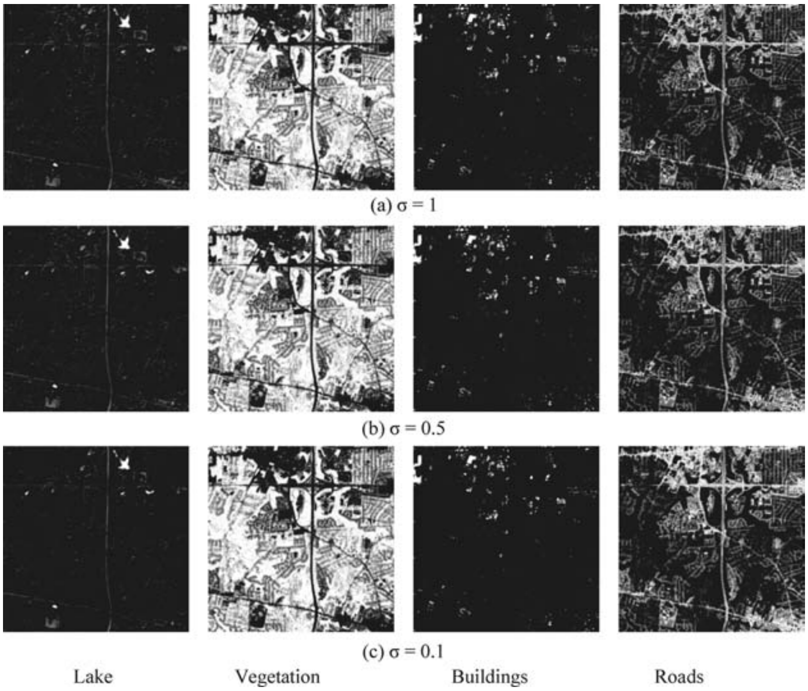
Figure 31.18 Unmixed results of SPOT data produced by KLSOSP using BEP9 with different values of σ.


(a) σ = 100

(b) σ = 50

(c) σ = 10

Figure 31.19 Unmixed results of SPOT data produced by KNCLS using BEP9 with different values of σ.


(a) σ = 1

(b) σ = 0.5

(c) σ = 0.1

Figure 31.20 Unmixed results of SPOT data produced by KFCLS using BEP9 with different values of σ.

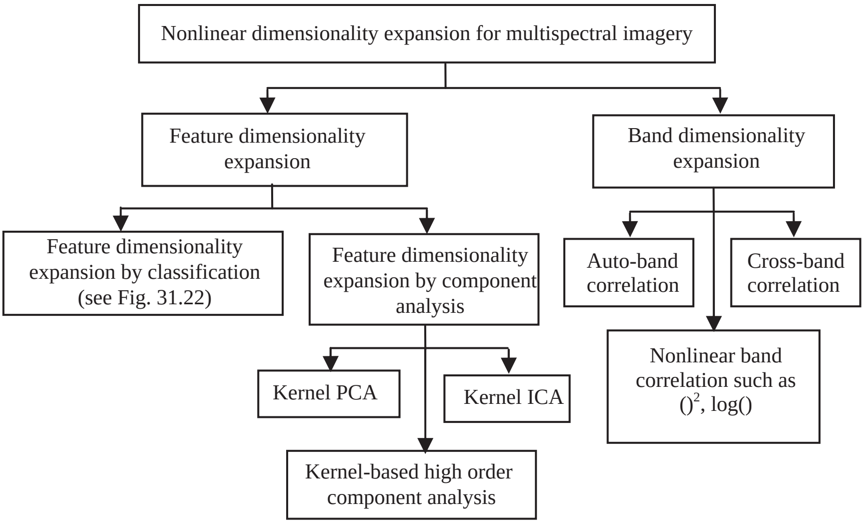

Nonlinear dimensionality expansion for multispectral imagery
Feature dimensionality expansion
Band dimensionality expansion
Feature dimensionality expansion by classification (see Fig. 31.22)
Feature dimensionality expansion by component analysis
Auto-band correlation
Cross-band correlation
Kernel PCA
Kernel ICA
Nonlinear band correlation such as ()², log()
Kernel-based high order component analysis

Figure 31.21 Nonlinear dimensionality expansion techniques to multispectral imagery.

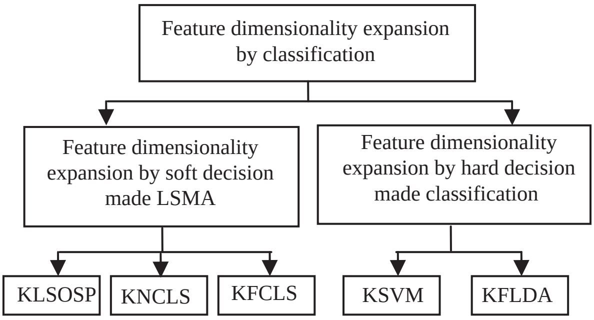

Feature dimensionality expansion by classification
Feature dimensionality expansion by soft decision made LSMA
Feature dimensionality expansion by hard decision made classification
KLSOSP
KNCLS
KFCLS
KSVM
KFLDA

Figure 31.22 Feature dimensionality expansion by classification techniques to multispectral imagery.

To conclude this section, it is worth mentioning that there is another application of applying BEP + LSMA and KLSMA to magnetic resonance (MR) image classification presented in the following chapter, Chapter 32. The superior performance of KLSMA to LSMA in quantitative analysis will be shown in this chapter and has been also demonstrated in Wong (2011) by experiments using synthetic magnetic resonance (MR) brain images provided by McGill University (http://www.bic.mni.mcgill.ca/brainweb/). Those who are interested in quantitative studies and analyses are encouraged to find details in this reference.

31.7 Conclusion

When hyperspectral imagery (HSI) was available for data processing in early 1990s, a common approach is to extend existing multispecral imaging techniques in a straightforward manner for processing HSI with a general understanding that hyperspectral imagery is an extension of multispectral imagery by including more spectral bands with better spectral resolutions. Unfortunately, this is generally not true. One main reason is that issues to be resolved in HSI such as subpxiels, mixed pixels, and endmembers are quite different from those in multispectral imagery (MSI) such as land cover/use classification, geographical information system (GIS), etc. The work in Chang (2003a) was developed to design statistical signal processing algorithms for subpixel detection and mixed pixel classification from a viewpoint of HSI. Some of them such as orthogonal subspace projection (OSP) and constrained energy minimization (CEM) have been shown to be promising in LSMA. However, as noted in Ren and Chang (2000a) such hyperspectral imagery-based developed techniques suffer from an issue of intrinsic dimensionality constraint, which does not necessarily guarantee that the same success can also be applied to multispectral imagery. This chapter investigates this issue and further develops two approaches to nonlinear dimensionality expansion to MSI. One is band dimensionality expansion (BDE) which creates new spectral band images resulting from implementing nonlinear functions on the original MSI data via a band generation process (BEP). As a result of BEP, the expanded MSI data will have sufficient band dimensionality and can be treated as if it is a hyperspectral image so that hyperspectral imaging techniques are readily applied. As an alternative to BDE, another dimensionality expansion is referred to as FDE which makes use of the kernel trick to perform kernelization on features derived from data sample vectors or training samples data or classifiers themselves in which case, no band dimensionality expansion is required, but features have been nonlinearly expanded in a higher dimensional feature space. Figure 31.21 describes these two types of nonlinear dimensionality expansion to multispectral imagery, BDE and FDE where various techniques developed each approach in this chapter are detailed in block diagrams in Figures 31.21 and 31.22.

32

Multispectral Magnetic Resonance Imaging

Due to recent advances in instrument technology, applications of remote sensing are no more confined to geoscience and earth science but have rather expanded to other areas such as medical diagnosis, food safety and inspection, law enforcement, defense, homeland security, and so on. Chapter 31 explores an expansion of hyperspectral imaging (HSI) techniques to multispectral imaging (MSI). This chapter deals with another new expansion of HSI to magnetic resonance (MR) imaging. Specifically, the problem of partial volume estimation (PVE) will be studied here. In the past years, two general approaches, a finite Gaussian mixture (FGM) model-based statistical approach and a fuzzy c-means (FCM)-based structural approach have been implemented in conjunction with Markov random field (MRF) to investigate PVE problems. This chapter develops a third spectral approach, which is completely different from the two above-mentioned approaches. It is a new PVE approach, which is based on linear spectral mixture analysis (LSMA) discussed in Part III of this book and Chapter 31. It assumes that an MR image voxel is linearly mixed with tissues of different types via a linear mixing model from which it can be further unmixed into abundance fractions of these tissues in terms of their partial volumes. To further effectively explore intravoxel spectral information within an MR image, two nonlinear expansions using nonlinear band dimensionality and nonlinear kernels developed in Chapter 31 are also used to explore inherent nonlinear spectral information, which can help increase the accuracy of LSMA-unmixed partial volume estimates. In order to conduct a quantitative study and analysis of PVE, the synthetic images provided by McGill's BrainWeb MR image database are used for experiments where a 3D receiver operating characteristics (3D ROC) analysis introduced in Chapter 3 is used for performance evaluation.

32.1 Introduction

Magnetic resonance imaging (MRI) is an advanced medical instrument technology, which provides high contrast of image intensity of information about soft tissues that can be used for image analysis (Sebastiani and Barone, 1991; Wright, 1997). Therefore, one of the fundamental tasks of MRI is tissue classification, which is generally accomplished by segmentation. Technically speaking, classification and segmentation are two completely different concepts. The classification generally requires a set of training samples to perform class membership assignment, which can be carried

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

out in a supervised or an unsupervised manner depending upon how training samples are produceed *a priori* using prior knowledge or *a posteriori* obtained directly from the data. On the other hand, segmentation intends to group data samples into a finite number of homogeneous regions, *p*, according to a certain criterion such as similarity of image intensities, custom-designed features, and so on. Therefore, segmentation is usually performed in an unsupervised manner without assuming any prior knowledge other than the value of *p*. Many efforts have been devoted to design and development of segmentation algorithms, most of them based on the concept of c-means (k-means) clustering and their various fuzzy versions (Bezdek, 1981; Bezdek et al., 1984). As a result, it is generally referred to as an automatic or unsupervised processing. In order for segmentation to perform classification, a set of training samples is generally required.

Due to the fact that MR image data are obtained by three image pulse sequences, T1, T2, and proton density (PD), MR image data are indeed a 3D image cube where each image pixel is actually a three-dimensional vector, referred to as voxel, of which each component is a pixel specified by one particular image sequence. Consequently, one very challenging issue encountered in MR image classification is the so-called partial volume effect in the sense that a data sample can be assigned to more than one class (Choi et al., 1991; Santago and Gage, 1993; Laidlaw et al., 1998; Harris et al., 1999; Ruan et al., 2000; Shattuck et al., 2001; Leemput et al., 2003; Siadat and Soltanian-Zade, 2007; Klauschen et al., 2009). Interestingly, a similar issue in PVE, called mixing effect, also occurs and presents a great challenge to remote sensing image processing where such mixing effect results from two types of mixing activities encountered in a remote sensing image pixel vector: macrospectral mixture (Singer and McCord, 1979) and intimate mixture (Hapke, 1981), both of which model a mixed pixel as a linear and nonlinear combination of a finite number of basic signatures, called endmembers, respectively. Nevertheless, Hapke's nonlinear mixing model can be linearized by a method suggested by Johnson (1983). Accordingly, LSMA has become a major technique in remote sensing image classification to perform linear spectral unmixing (Keshava and Mustard, 2002). In essence, what is PVE to MRI is the same as what is spectral unmixing to MSI.

The similarity between MRI and MSI was first realized by Vannier et al. (1985) who suggested that many multispectral satellite image processing techniques could readily be applied to MR image data, although they did not particularly specify what these techniques were. Recently, hyperspectral imaging has emerged as an advanced technique in remote sensing (Chang, 2003a). Many efforts have also been made in applying hyperspectral imaging techniques to MRI and achieved some success has been achieved in MR tissue classification (Wang, et al., 2001, 2002, 2003; Chen et al., 2005; Wong and Chang, 2008; Wong, 2008). Unfortunately, the issue of PVE was not specifically addressed there. This is because the connection with PVE and spectral unmixing was not established then and was not realized either.

There are many approaches proposed to solve PVE. Two mainstreams have been studied over the past years. One is parametric methods via statistical approaches (Choi et al., 1991; Santago and Gage, 1993; Laidlaw et al., 1998; Harris et al., 1999; Ruan et al., 2000; Shattuck et al., 2001; Leemput et al., 2003; Siadat and Soltanian-Zade, 2007; Klauschen et al., 2009). An early attempt along this line was made by Choi et al. who used a linear mixture to model an MR image voxel as a finite Gaussian mixture where signal sources in the mixture were assumed to be Gaussian distributed so that Dempster et al.'s expectation-maximization (EM) algorithm (Dempster et al., 1977) could be used to find mixing coefficients in the mixture to solve PVE. This approach is referred to as two-step framework in Siadat and Soltanian-Zadeh (2007), which assumed a density mixture model in the first step followed by a histogram fitting in the second step (i.e., Gaussian mixture). Another is nonparametric methods, which can be considered as structural approaches and are mainly fuzzy c-means (FCM)-based techniques

(Wells, III, et al., 1996; Pham and Prince, 1999; Ahmed et al., 2002; Liew and Yan, 2003; Siyal and Yu, 2005). In order to further take into account spatial information, both approaches (parametric and nonparametric methods) include Markov random field (MRF) (Zhang et al., 2001; Scherrer et al., 2009) to capture spatial properties within a custom-designed neighboring system, called cliques. Such EM-MRF-based and FCM-MRF-based approaches along with their various adaptive versions have accounted for most research works currently reported in MR image classification and analysis. Although as good as they can be theoretically, each of them suffers from several drawbacks in practical applications. In the case of parametric methods, assumption of finite Gaussian mixture is made, which is generally not true in real applications. In particular, the noise assumed in the mixture to account for the model error is generally not Gaussian (Gravel et al., 2004). Second, the computational complexity of using EM in conjunction with MRF to find mixing components is exceedingly high and expensive. This is probably the main reason that only two different tissue types have been considered in many works (Siadat and Soltanian-Zadeh, 2007). With regard to nonparametric methods, the main issue concerns the used initial condition that requires selecting seed samples to initialize a clustering process is generally randomly generated that usually results in inconsistent and unreproducible results. Most importantly, if there are more than three image sequences used for data collection, including flair images, the 3×3 cliques-based neighboring system used by MRF in both approaches must be extended to dimensionality higher than 3, which presents a great challenge, if not impossible. Finally, the number of classes to be considered by both approaches has been limited to three or four based on the fact that there are only three brain tissues of interest, white matter (WM), gray matter (GM), and cerebral spinal fluid (CSF), in MR brain image analysis. This constraint limits the use of two popular software algorithms, SPM 5/8 and FAST, which will break down if the number of classes exceeds three for SPM 5/8 and six for FAST. However, it is obvious that there are more tissue types in brain MR images in addition to WM, GM, and CSF, such as brain skull, fat, blood vessels, and muscles, which certainly have significant impacts on PVE and cannot be simply ignored and discarded.

In order to resolve the above-mentioned PVE issues, this chapter develops a third approach from a hyperspectral imaging perspective (Chang, 2003a). It is an LSMA-based technique, which is completely different from the classical FCM-MRF-based structural (spatial) and the EM/FGM-MRF-based statistical approaches. It is an approach similar to Choi et al.'s PVE approach (1991) in the sense that both make use of a linear mixing model to cope with PVE effects, but quite different in other aspects. Unlike mixing models used in MRI that assume data samples can be modeled by a finite Gaussian mixture where the EM algorithm is implemented to find partial volumes for tissue types in the mixture, the proposed LSMA does not make this assumption. Instead, it uses the least-squares error (LSE) as an optimal criterion to unmix various tissue types by finding their abundance fractions present in the mixture in terms of their corresponding partial volumes where no probability distributions are involved to describe the tissue types. As a result, it does not require probability distribution estimation as many parametric methods do, such as Gaussian distributions. Second, the number of tissues of various types can be arbitrary and determined by practical needs. Specifically, the number of tissue types can go beyond three and six, which are the upper limits imposed on two popular software packages commonly used in the literature, SPM 5/8 with the software available at http://www.fil.ion.ucl.ac.uk/spm/software/SPM8/ and FAST (FMRIB's Automated Segmentation Tool) in Smith (2000), Smith et al. (2004) and Zhang et al. (2001) with the software available at http://www.fmrib.ox.ac.uk/fsl/, respectively. Third, LSMA is a pixel-based approach, which relies only on spectral properties to perform various tasks, such as subpixel detection, mixed pixel classification, quantification, and so on. Consequently, its computational complexity is very low. Fourth, it can use other tissue types of no interest such as brain skull or fat to suppress background via unmixing so as to increase the unmixing ability to improve accuracy of partial volume estimation (PVE), a task which is difficult for EM-MRF and FCM-MRF approaches to accomplish. However, since this LSMA-based approach does not include MRF to account for spatial correlation captured by a neighboring system, LSMA must further explore more spectral properties to make up this deficiency. In doing so, an approach called band expansion process (BEP), similar to band generation process (BGP) proposed by Ren and Chang (2000a), is incorporated into LSMA to generate a new set of extra spectral bands from the original MR images by a nonlinear function. By virtue of BEP, the original MR images can include nonlinear spectral information in expanded MR images. LSMA is then implemented in conjunction with BEP to perform PVE and is called BEP-LSMA (Wong, 2008; Wong and Chang, 2008). As an alternative, LSMA can also explore nonlinear spectral information via a nonlinear kernel to allow spectral unmixing to convert nonlinear decisions for partial volume estimation to linear decisions in a high-dimensional feature space instead of the original MR images. Such an approach is called kernel-based LSMA (KLSMA).

Several advantages of LSMA using nonlinear expansions over the mixed model-based EM approach are (1) no Gaussian assumption made for tissue types or noise or model error; (2) no probability distribution estimation required; and (3) low computational complexity. On the other hand, there are also advantages of LSMA using nonlinear expansions over FCM-based approaches. First, unlike FCM which is primarily developed for data clustering with fuzzy introduced to make soft decisions that can be used for PVE, LSMA is mainly designed for PVE. Second, FCM-based methods generally cannot find subtle objects such as abnormal tissue types if their samples are relatively few, in which case these small objects are very much likely to be assigned to a large cluster owing to a small size of clusters made up of such small objects. Third, if there are tissues of the same type scattering in more than one disconnected regions, which are far away from each other, these tissue samples are also very likely to be clustered into two separate connected regions according to their spatial locations in which case they will be considered as different objects. All the above-mentioned issues are not encountered in LSMA, which is performed on a single voxel basis and does not use sample spatial correlation to perform its tasks. Fourth, in PVE the volumes of tissue types of interest, referred to as abundance fractions in LSMA, are considered as parameters to be estimated. Therefore, parametric mixing model approaches generally perform better than nonparametric FCM clustering approach. Most importantly, LSMA can use undesired tissue types to suppress their interfering effects on classification of desired tissue via unmixing to enhance desired tissue type discriminability, while FCM can only group undesired tissue types as separate clusters for segmentation but not for suppression.

32.2 Linear Spectral Mixture Analysis for MRI

A fundamental task of MRI is tissue classification. Traditionally, it takes advantage of intervoxel spatial correlation to perform spatial domain-based classification by voxel membership assignment. In other words, a spatial domain-based classification technique performs no more than class labeling for data samples via a clustering or segmentation process. As a result, MR image voxels must be classified by hard decisions, that is, discrete decisions. Unfortunately, since many tissue substances in MR images may be indeed mixed by more than one tissue substance within a single voxel, it is more realistic and effective to classify an MR image voxel by soft decisions based on the proportion of each of these tissue substances present in the particular voxel. In the past, two mainstreams are investigated for such soft decisions, referred to as PVE. One is a parametric approach that makes use of a mixing model to estimate partial volumes of each of tissue substances present in a voxel, also referred to as a mixel (Choi et al., 1991; Santago and Gage, 1993; Laidlaw et al., 1998; Harris et al., 1999; Ruan et al., 2000; Shattuck et al., 2001; Leemput et al., 2003; Siadat and Soltanian-Zade, 2007; Klauschen et al., 2009). The other is a nonparametric approach, which uses FCM-based clustering techniques for the same purpose (Wells, III, et al., 1996; Pham and Prince, 1999; Ahmed et al., 2002; Liew and Yan, 2003; Siyal and Yu, 2005).

Recently, a third approach has been investigated by Vannier et al. (1985) where MR images are considered as multispectral images and each spectral image results from a particular magnetic pulse sequence designed for data acquisition. In light of this interpretation, LSMA discussed in Part III is readily applied to MR image analysis. It assumes that there are three spectral images obtained by three magnetic resonance tissue parameters, spin-lattice (T1), spin–spin (T2) relaxation times, and PD where each data sample **r** is a three-dimensional image voxel. Suppose that there are a number of principal tissue substances $\{\mathbf{m}_j\}$ such as GM, WM, and CSF, to form a set of basic constituents. LSMA models each MR image voxel **r** as a linear mixture of $\{\mathbf{m}_j\}$ with appropriate abundance fractions, $\{\alpha_j\}$. The goal of LSMA is to use these tissues substances $\{\mathbf{m}_j\}$ to unmix **r** by solving a linear inverse problem via a linear mixing model to find the unknown abundance fractions, $\{\alpha_j\}$. The estimated abundance fraction $\hat{\alpha}_j$ is considered as a partial volume of the *j*th tissue substances $\mathbf{m}_j$. By virtue of such unmixed partial volumes $\{\hat{\alpha}_j\}$, various tasks of image analysis can be performed for MR images. Several early attempts have been made using LSMA for MR brain images and shown promise in solving certain problems, such as subpixel detection, mixed pixel classification, and quantification that cannot be resolved by traditional spatial domain-based image processing techniques (Wang et al., 2001, 2002, 2003; Chen et al., 2005; Wong and Chang, 2008; Wong, 2008). Nevertheless, LSMA also suffers from an inherited drawback of using a limited number of image pulse sequences to obtain MR images. For a spatial domain-based technique, this may not be a problem since it mainly relies on intervoxel spatial correlation among data sample voxels to perform its tasks. But for a technique such as LSMA, which makes exclusive use of spectral information to characterize tissue substances, the lack of spectral band images can certainly impair its ability in performing tasks. In order to address this issue, an interesting approach proposed by Ren and Chang (2000a) introduced the concept of BGP that allowed users to create additional nonlinear spectral information from the original set of spectral images via nonlinear functions. This idea was successfully applied to MR images (Ouyang et al., 2008a) as band dimensionality expansion (BDE) where BGP was renamed as BEP. As an alternative to BDE that expands band dimensionality of the original data space, an approach that used nonlinear kernels to expand linear classifiers to resolve the linear nonseparability issue was also investigated in Wong (2011). These two approaches are rather different. BDE is particularly designed to address the issue arising in an insufficient number of spectral band images, whereas the kernel approach expands capability of linear classifier to deal with linearly nonseparable problems more effectively in a kernel-transformed feature space. Both are proved to be promising in multispectral imaging (Wong and Chang, 2008; Wong, 2008; Liu et al., 2009). Interestingly, the idea of using LSMA via a nonlinear kernel to expand feature dimensionality has not been explored for MR image analysis in the past. Despite that BDE was investigated in Ouyang et al. (2008a), BDE was demonstrated ineffective if the SVM was implemented alone, but has become very useful and effective when the SVM was implemented coupled with the independent component analysis (ICA).

This chapter considers three LSMA methods: abundance-unconstrained least-squares orthogonal subspace projection (LSOSP), partially abundance nonnegativity-constrained least-squares (NCLS), and abundance-fully constrained least-squares (FCLS) for nonlinear dimensionality expansion. The first technique is to operate the three LSMA-based methods, LSOSP, NCLS, and FCLS, on band dimensionality-expanded image data with additional new band images generated by BEP, referred to as BEP-LSOSP, BEP-NCLS, and BEP-FCLS, respectively. A second technique is to expand LSOSP, NCLS, and FCLS into a high dimensional feature space via nonlinear kernels, referred to as K-LSOSP, K-NCLS, and K-FCLS. A third technique is to combine both BEP and kernelization to implement K-LSOSP, K-NCLS, and K-FCLS on BEP-expanded data, referred to as K-BEP-LSOSP, K-BEP-NCLS, and K-BEP-FCLS. The goal of this chapter is to introduce these approaches to demonstrate how the commonly known and well-established linear spectral unmixing can be reinvented to be used as effective MRI analysis techniques in tissue characterization in terms of PVE.

Since LSMA is basically an estimation technique, i.e., estimating abundance fractions to perform classification, it is a soft classifier as opposed to commonly used hard classifiers that perform classification via class labeling assignment. As a result, the traditional receiver operating characteristic (ROC) analysis designed for hard classifiers for performance evaluation in medical diagnosis cannot be directly applicable to LSMA, which makes soft decisions on estimated abundance fractions to perform classification. To deal with this issue, the three-dimensional receiver operating characteristic (3D ROC) analysis developed in Chapter 3 is used for performance evaluation. In addition to detection probability P_{D} and false alarm probability P_{F}, an additional parameter τ is introduced in a third dimension as a threshold to convert abundance fractions to hard decisions for class membership assignment. In order to substantiate LSMA-based methods, two image sets are used for experiments. One dataset is set of synthetic brain MR images obtained from a website http://www.bic.mni.mcgill.ca/brainweb/ that provided ground truth which can be used to conduct quantitative study and analysis. The second dataset is a set of real MR images obtained in Taichung Veterans General Hospital to validate the utility of LSMA-based methods in real applications so that it paves a way for developing computer assisted software to help radiologists in doing better PVE of MR tissue substances.

32.2.1 Orthogonal Subspace Projection to MRI

In order to see how the concept of LSMA can be applied to brain MRI, we assume that a multispectral brain MR image contains three major brain tissue substances, WM, GM, CSF, along with background tissue substances such as fatness, water, or blood vessels, to form a set of desired endmembers, $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p\}$ that will be used to describe a linear model. For any brain image pixel $\mathbf{r}$, it can be represented by a linear combination of $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p\}$ with their respective coefficients $\{\alpha_1, \alpha_2, \ldots, \alpha_p\}$, which are unknown and yet to be found. In this case, these coefficients are referred to as abundance fractions, which indicate how much abundance of each of p endmembers attributed to the sample $\mathbf{r}$. Despite that the eigenimaging filter (Windham et al., 1988) was the first work to envision the idea that a sequence of MR images can be weighted to produce a composite for feature extraction, it is interesting to note that a natural connection of MRI to spectral unmixing has not been investigated and explored until recent works reported (Wang et al., 2001, 2002) where a hyperspectral image processing technique, orthogonal subspace projection (OSP), was successfully applied to MR image classification and spectral characterization. In this section, OSP will be discussed in detail.

OSP was originally developed for hyperspectral image classification (Harsanyi and Chang, 1994) and discussed in Chapter 2. It models an image pixel as a linear mixture of finite number of known signatures assumed to be present in the image. More specifically, suppose that L is the number of spectral bands (channels). In our case, an MR image sequence is actually a collection of coregistered L spectral bands. So, the ith image pixel in an MR image sequence can be considered as an L-dimensional pixel vector.

Let $\mathbf{r}$ be an $L \times 1$ column pixel vector in a hyperspectral image. Assume that there is a set of p MR substances of interest present in the image and $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p$ are their corresponding signatures, called MR tissue signatures. Let $\mathbf{S}$ be an $L \times p$ MR signature matrix denoted by $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p]$, where $\mathbf{s}_j$ is an $L \times 1$ column vector represented by the signature of the jth MR tissue substance present in the MR image sequence. Let $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_p)^T$ be a $p \times 1$

abundance column vector associated with **r**, where α_j denotes the fraction of the *j*th MR tissue signature $\mathbf{s}_j$ present in the pixel vector **r**. Then the signature of **r** can be represented by a linear mixture model described by (2.75) with the signature matrix **M** replaced by the MR tissue substance signature matrix **S** as follows:

$$\mathbf{r} = \mathbf{S}\boldsymbol{\alpha} + \mathbf{n} \tag{32.1}$$

where **n** is noise or can be interpreted as either a measurement error or a model error. It should be noted that (32.1) uses different notations of $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p$ and **S** to represent *p* MR tissue substance signature matrix, to differentiate, respectively, the MRI from $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ and **M** in (2.75), which has been used for remote sensed imagery in previous chapters.

Equation (32.1) assumes that the knowledge of MR tissue substance signatures, **S,** must be given *a priori*. Without loss of generality, we further assume that $\mathbf{d} = \mathbf{s}_p$ is the desired MR tissue signature to be detected in (32.1) and $\mathbf{U} = \left[\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{p-1}\right]$ is the undesired signature matrix made up of the remaining $p - 1$ undesired MR tissue substance signatures in **S**. Then, we rewrite (32.1) as:

$$\mathbf{r} = \mathbf{d}\alpha_p + \mathbf{U}\boldsymbol{\gamma} + \mathbf{n} \tag{32.2}$$

where $\boldsymbol{\gamma}$ is the abundance vector associated with **U**. Equation (32.2) allows us to design the following orthogonal subspace projector, denoted by $P_{\mathbf{U}}^{\perp}$ to annihilate **U** from **r** prior to detection of **d** get rid of boldface.

$$P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\mathbf{U}^{\#} \tag{32.3}$$

where $\mathbf{U}^{\#} = \left(\mathbf{U}^T\mathbf{U}\right)^{-1}\mathbf{U}^T$ is the pseudoinverse of **U**. Applying $P_{\mathbf{U}}^{\perp}$ in (32.3) to (32.2) results in

$$P_{\mathbf{U}}^{\perp}\mathbf{r} = P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p + P_{\mathbf{U}}^{\perp}\mathbf{n} \tag{32.4}$$

Using the signal-to-noise ratio (SNR) as the criterion for optimality, the optimal solution to (32.4) is given by (2.77), which is a matched filter $\mathrm{M}_{\mathbf{d}}$ defined by

$$\mathrm{M}_{\mathbf{d}}\left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right) = \kappa\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} \tag{32.5}$$

where the matched signal is specified by **d** and κ is a constant. Setting $\kappa = 1$ in (32.5) yields the following OSP filter $\delta^{\mathrm{OSP}}(\mathbf{r})$:

$$\delta^{\mathrm{OSP}}(\mathbf{r}) = \mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)\alpha_p + \mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{n} \tag{32.6}$$

The OSP specified by (32.6) detects only the abundance fraction, which does not reflect true amount of the desired MR tissue signature **d**. In order to reliably estimate the abundance fraction of **d**, a least-squares OSP, $\delta^{\mathrm{LSOSP}}(\mathbf{r})$, is given by

$$\delta^{\mathrm{LSOSP}}(\mathbf{r}) = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r} = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}\delta^{\mathrm{OSP}}(\mathbf{r}) \tag{32.7}$$

where the constant $\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$ is included in (2.79) to account for the least-squares estimation error (Chang, 1998, Chang et al., 1998b).

Due to mathematical tractability, LSMA is widely implemented as an unconstrained technique that does not impose any constraint on the abundance fractions $\alpha_1, \alpha_2, \ldots, \alpha_p$ of the MR tissue substance signatures $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p$. However, it has been shown in the literature, for example,

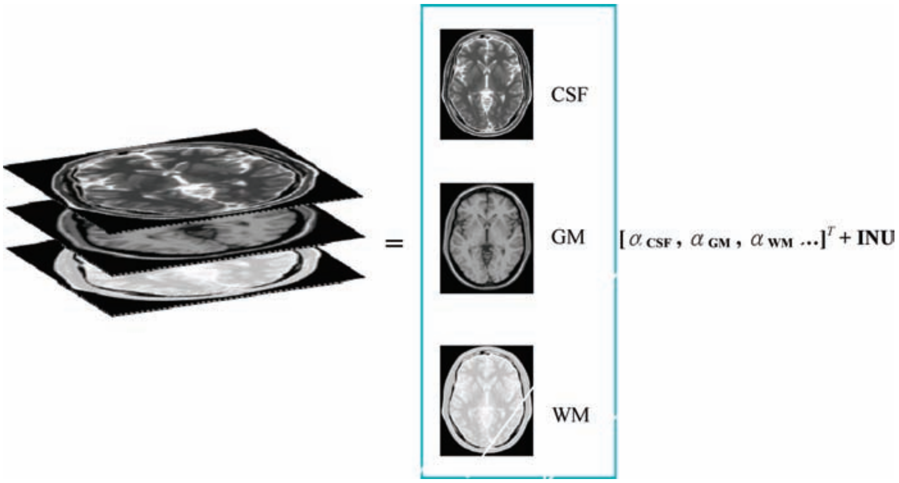


Figure 32.1 An example of a 3-band MR image cube acquired by T1, T2, and PD unmixed into three images to show brain tissue substances: CSF GM, WM respectively.

constrained LSMA can improve unconstrained LSMA in many aspects (Chang, 2003a) such as subpixel detection, mixed pixel classification, identification, and specifically quantification when accurate abundance fraction estimation is required. This is also of particular importance for finding accurate partial volumes of MR tissue substances such as CSF, WM and GM as shown in Figure 32.1. As constrained LSMA is considered, two abundance constraints can be imposed on $\alpha_1, \alpha_2, \ldots, \alpha_p$ in (32.1): abundance sum-to-one constraint (ASC), that is, $\sum_{j=1}^{p} \alpha_j = 1$ and abundance nonnegativity constraint (ANC), $\boldsymbol{\alpha} \geq \mathbf{0}$, that is, $\alpha_j \geq 0$ for all $1 \leq j \leq p$. Such constrained LSMA is referred to as abundance-constrained LSMA (AC-LSMA) considered in Chapter 14. A general and common approach to solving AC-LSMA is to estimate abundance fractions in the sense of LSE while satisfying the imposed constraints, which are referred to as LSE-based AC-LSMA. More specifically, the model in (32.1) is interpreted as least-squares error problem:

$$(\mathbf{r} - \mathbf{S}\boldsymbol{\alpha})^T(\mathbf{r} - \mathbf{S}\boldsymbol{\alpha}) \tag{32.8}$$

with **n** modeled as least-squares error, while constraining the abundance fractions $\alpha_1, \alpha_2, \ldots, \alpha_p$ on the model in (32.1) to find least-squares error solutions. As a result, three types of LSE-based abundance-constrained LSMA are generally considered (Chang, 2003a): sum-to-one constrained least-squares (SCLS) that implements only the ASC, NCLS that implements only the ANC, and FCLS that implements both ASC and ANC. Despite the fact that constrained LSMA may require more sophisticated algorithmic implementations, the payoff is sometimes great and worthwhile, specifically for material quantification. Most importantly, it generally produces more accurate abundance fraction estimation, specifically for PVE.

32.2.2 Band Expansion Process-Based OSP

In order for OSP to perform effectively, one key requirement is to assume that there is sufficient data dimensionality for OSP to carry out orthogonal subspace projections. In other words, OSP utilizes spectral bands to capture spectral characterization profiled in an image pixel. If the data dimensionality is too small, the spectral profiles provided by spectral bands may not have sufficient spectral information for data analysis. It seems natural to hyperspectral imagery since it generally has hundreds of spectral bands to be used for data collection. However, it is not the case for MR images, which usually have only a few spectral bands that can be spared to be used for data analysis. In this case, the performance of OSP will be considerably degraded due to an insufficient number of spectral bands. If we use one-band MR image to accommodate one brain tissue substance, the number of substances to be accommodated should not exceed the number of MR band images, which is generally three. Unfortunately, there are always more than three brain tissue substances such as GM, WM, and CSF, fatness, blood, water, and so on, plus noise. In this case, the band expansion process-based OSP (BEP-OSP) described in Section 31.3.1 seems appropriate to be used for this purpose.

A procedure of implementing the BEP-OSP is briefly described in Chapter 31 (Section 31.2.2) by two stages, BEP followed by OSP as follows.

BEP-OSP

Stage 1: Use the BEP to expand spectral dimensionality.
Stage 2: Implement an unconstrained OSP/LSOSP or constrained OSP such as FCLS to perform linear spectral unmixing.

32.2.3 Unsupervised Orthogonal Subspace Projection

According to (32.1), OSP needs complete prior knowledge of target signal sources $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p$. In many practical applications, such prior knowledge is extremely difficult to obtain, if not impossible. In order to mitigate this dilemma, OSP is extended to an unsupervised version of OSP where the target knowledge can be automatically generated directly from the data in an unsupervised manner. The algorithm to be used for this purpose is the automatic target generation process (ATGP) described in Section 30.2, which was previously developed to find potential target pixels that can be used to generate a signature matrix used in an unsupervised manner.

Now, by virtue of ATGP, we can extend OSP in (32.6) or LSOSP in (32.7) to an unsupervised orthogonal subspace projection (UOSP), referred to as ATGP-OSP, which produces its own signature matrix $\mathbf{S} = \left[\mathbf{t}_0, \mathbf{t}_1, \ldots, \mathbf{t}_{p-1}\right]$ for (32.1) that is obtained directly from the image and is made up of the p target pixel vectors $\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{p-1}$ generated by ATGP. The implementation of ATGP-OSP can be briefly summarized as follows.

ATGP-OSP algorithm

1. Preset the value of p, number of signatures of interest.
2. Apply the ATGP to generate p target signatures, $\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{p-1}$.
3. Form a desirable signature matrix, $\mathbf{S} = \left[\mathbf{t}_0, \mathbf{t}_1, \ldots, \mathbf{t}_{p-1}\right]$ for (32.1).
4. Due to the unavailability of prior knowledge of signatures, LSOSP classifier, $\delta^{\mathrm{LSOSP}}(\mathbf{r})$ described by (32.7), must be applied to classify each of p signatures $\mathbf{t}_0, \mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_{p-1}$. In other words, LSOSP classifier must be performed p times to classify all the p signatures. In this case, in order to classify the jth signature $\mathbf{t}_j$, the $\mathbf{d}$ and $\mathbf{U}$ in $\delta^{\mathrm{LSOSP}}(\mathbf{r})$ are specified by $\mathbf{t}_j$ and $\mathbf{U}_j = \left[\mathbf{t}_0 \ldots \mathbf{t}_{j-1}\mathbf{t}_{j+1} \ldots \mathbf{t}_{p-1}\right]$.

32.3 Linear Spectral Random Mixture Analysis for MRI

Recall that (32.1) is essentially a deterministic model, which assumes that signal sources $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p$ used for mixing are deterministic sources and the abundance fractions $\alpha_1, \alpha_2, \ldots, \alpha_p$

are unknown constants and need to be estimated by solving unconstrained or constrained optimization problems. The OSP and FCLS developed in Section 32.2.1 provide unconstrained and constrained solutions, respectively.

In this section, we look into an alternative to the linear mixing model used in (32.1) in a very different way. In particular, we assume that the signal sources $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p$ in (32.1) are no longer deterministic sources, but rather random sources. More specifically, the signal sources $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p$ are considered as mutual statistically independent random processes. Given the nature of nonstationarity present in MR images, this assumption seems reasonable. On the other hand, two signal sources, which are considered to be distinct, are supposed to be uncorrelated in some sense. Mutual statistical independency provides, a good criterion to define distinct signal sources. With these assumptions in mind, a recently developed and well-established technique, called ICA, plays a key role in materializing such a new development.

More specifically, the key idea of ICA assumes that data are linearly mixed by a set of separate statistically independent signal sources, and these signal sources can be demixed according to their statistical independency measured by mutual information. In order to validate its approach, an underlying but very crucial assumption is that at most one signal source in the mixture model can be allowed to be a Gaussian source. This is due to the fact that a linear mixture of Gaussian sources is still a Gaussian source. More precisely, by modifying the model in (32.1), we let $\mathbf{x}$ be a mixed signal source vector expressed by

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{32.14}$$

where $\mathbf{A}$ is an $L \times p$ mixing matrix and $\mathbf{s}$ is a p-dimensional signal source vector with p signal sources needed to be separated. It should be noted that the noise $\mathbf{n}$ in (32.1) is now absorbed as an independent random source in the mixing model (32.14). Two scenarios are of interest in implementing ICA. One is the case that the mixing matrix $\mathbf{A}$ in (32.14) has more dimensions than it requires for blind signal separation, that is, $L > p$. In this scenario, ICA has fewer bases (i.e., signal sources) than the samples provided (i.e., observations in the observable vector $\mathbf{x}$) and, thus, referred to as under complete ICA (UC-ICA), which implies that ICA has under representative bases. In this case, UC-ICA can be considered as a counterpart of under complete LSMA (UC-LSMA), discussed in Chapter 31. According to system theory, UC-ICA described by (32.14) is actually an over determined system, in which case there exists no solution to (32.14). In order to resolve this dilemma, a DR is generally used to reduce dimensionality of the mixing matrix $\mathbf{A}$ from L to p to make (32.14) solvable. At the other extreme, if (32.14) has fewer samples than the sources to be demixed, that is, $L < p$, ICA of this type is called over complete ICA (OC-ICA), which implies that it has over representative bases to solve an under determined system for (32.14). This OC-ICA turns out to be a counterpart of over complete LSMA (OC-LSMA), defined in Chapter 31. As a consequence, there are many solutions to (32.14), and it requires selecting best ICs to perform classification. Interestingly, there is very little work reported about how to cope with OC-ICA, particularly how to address the issues caused by insufficient ICs and the use of random initial projection vectors, which result in inconsistent ICs. However, due to the nature of OC-ICA, only a limited number of ICs are available to be used for signal source separation. When the number of signal sources is greater than the number of ICs, some of ICs are forced to accommodate more than one signal source in which case there is no way to use such a particular IC to characterize signal sources. Another issue is the use of random initial projection vectors, which also cause random mixtures of signal sources as well as noise in each of ICs. Unfortunately, such severe disadvantages that have been overlooked and never been addressed effectively in the past will be investigated in this chapter and solutions will also be provided.

32.3.1 Source Separation-Based OC-ICA for MR Image Analysis

ICA has shown great success in functional magnetic resonance imaging (fMRI), which is a method that provides functional information of MR images in time series as a temporal function. Recently, a new application of ICA to MR image (MRI) analysis was investigated by Nakai et al. (2004) for contrast enhancement of GM and WM. A major difference between fMRI and MRI analyses is the mixing matrix to be used by ICA for signal source separation. Since the samples for fMRI are collected along a temporal sequence with the number of samples, denoted by L, generally greater than the number of sources to be separated, denoted by p, the ICA implemented in fMRI is actually under complete in the sense that ICA deals with underrepresentation of a mixed model, referred to as UC-ICA. Under such circumstance, ICA intends to solve an over determined system with L equations specified by the number of samples and p unknowns represented by signal sources to be separated, and there will be no solutions for $L > p$ in general because more than one IC must be used to accommodate a single signal source. This may be one reason that UC-ICA generally requires dimensionality reduction. By contrast, the samples used for MRI analysis are a stack of images obtained by different pulse sequences specified by three magnetic resonance tissue parameters: spin-lattice (T1), spin–spin (T2) relaxation times, and PD. So, generally speaking, these three images, T1 weighted, T2 weighted, and PD weighted, can be used for MRI analysis. If the number of signal sources to be separated, p, is greater than the number of different combinations of pulse sequences, L, with $L < p$, then one IC must be used to accommodate more than one signal source. In this case, ICA must deal with an under determined system using an over complete representation of a mixed model, referred to as OC-ICA. Accordingly, fMRI and MRI analyses are essentially different applications and approaches developed for one application usually cannot be directly applied to another. However, in order to take advantage of ICA implemented as UC-ICA in the same way that it is applied to the fMRI, Nakai et al. assumed that the number of sensors, L, is greater than or equal to the number of sources, p, where the number of sensors corresponds to the combinations of acquisition parameters echo time (TE) and repetition time (TR), and a signal source is represented by a tissue cluster characterized by a unique combination of T1 and T2 relaxation times. Using the changes in signal intensity of each tissue cluster reflected by combinations of TR and TE before and after an ICA transform, the contrast resulting from effects of ICA can be used to perform image evaluation for a particular tissue such as GM and WM. Unfortunately, Nakai et al.'s ICA approach overlooked a crucial and important issue. If we interpret the number of pulse sequences used in MR acquisition, denoted by L, and brain tissue substances such as GM, WM, CSF, muscle, skin, fat, and so on, as signal sources to be separated, denoted by p, the L is actually less than p, not greater than p. Consequently, the problem to be solved for MRI analysis is indeed an under determined system with $L < p$ that violates the key assumption made in Nakai's et al.'s ICA approach as well as in most ICA-based approaches used for fMRI. Interestingly, little work has been done regarding the use of OC-ICA to perform MRI analysis. Another issue that was not addressed by Nakai et al. is the use of random initial projection vectors by ICA to produce independent components (ICs). The problem with this random approach results from the fact that the final sets of projection vectors produced by two distinct random initial projection vectors are not necessarily the same. As a consequence, an ICA transform implemented by the same user in different runs or different users at the same time will produce different sets of ICs. This serious inconsistency undermines repeatability of ICA and makes ICA unstable. In addition, due to the use of random initial projection vectors, the order of ICA-generated ICs is completely random and does not necessarily indicate the significance or importance of an IC in its appearing order. In other words, an IC generated earlier need not be more important than ones generated later. Consequently, image evaluation must wait until all ICs are generated.

Most importantly, since the representation of a mixing model used by ICA is over complete, there are no sufficient ICs to accommodate brain tissue substances in addition to the WM, GM, and CSF. Namely, many single ICs can be used to separate more than one signal sources so that there is no unique solution to select which IC is the best for particular signal source. What is worse is that due to use of random initial projection vectors, brain tissue substances are also forced to be randomly mixed in different ICs. These two reasons, that is, (1) many solutions for OC-ICA and (2) the use of random initial projection vectors, are exactly the cause of inconsistent ICs in final results. For example, WM, GM, and CSF may be randomly accommodated in a single IC as will be demonstrated in our experiments in this chapter. Under such a circumstance, there is no best way to select a single IC to discriminate these three brain tissue substances from one another. This inevitable phenomenon is caused by the use of random initial projection vectors by ICA and the lack of ICs resulting from the inherent nature in OC-ICA. This chapter aims at addressing these two issues and further develops a rather different approach to implementing ICA, OC-ICA to improve Nakai's et al.'s ICA approach, which was actually UC-ICA.

32.3.2 Band Expansion Process Over complete ICA for MR Image Analysis

As noted in Section 32.3.1, OC-ICA suffers from two major issues. One is the same problem as OSP does, which is insufficient data dimensionality where a dimension expansion (DE) is required for OC-ICA to make a mixing matrix a square matrix as opposed to the UC-ICA, which requires DR for the same purpose. The BEP developed in Section 32.2.2 to resolve this issue for OSP can be directly applicable to OC-ICA. In other words, BEP expands dimensions by creating new nonlinearly correlated images with original MR images. These newly generated images combined with the original set of MR images provide a sufficient number of images required for the ICA to perform blind source separation where ICA to be used in this section is the FastICA developed by Hyvarinen and Oja (http://ida.first.fraunhofer.de/∼anton/software.html).

The second issue is essentially the same problem arising in both UC-ICA and OC-ICA, that is, inconsistency caused by the use of random initial conditions by any ICA algorithm. In order to cope with this problem, a new concept, called prioritized ICA (PICA), introduced in Chapter 6 is used to prioritize ICA-generated ICs according to a custom-designed criterion. The three PICA-based algorithms developed in Wang and Chang (2006a) as well as in Section 6.4: eigenvector-prioritized PICA (Eigen-PICA), high-order statistics-prioritized PICA (HOS-PICA), and ATGP-prioritized PICA (ATGP-PICA), will be used to implement the PICA. Since the data dimensionality is L, there are L ICs generated by the ICA, denoted by $\{\mathrm{IC_i}\}_{i=1}^{L}$.

32.3.2.1 Eigenvector-Prioritized ICA

A simplest way to prioritize ICs is to use eigenvalues as a priority measure. The idea of the eigen-PICA comes from the PCA where the PCs are ordered by data variance. In this case, the initial set of projection vectors used by FastICA is determined by a set of eigenvectors of the data matrix.

Eigen-PICA algorithm

1. Apply the BEP to expand data dimensionality if there is a need such as in the case of OC-ICA.
2. Find a set of eigenvectors of the data matrix $\{\mathbf{v}_j\}_{j=1}^{L}$ arranged in order of magnitude of their corresponding eigenvalues.
3. Use each of $\{\mathbf{v}_j\}_{j=1}^{L}$ generated in step 1 as an initial projection vector, FastICA produce $\{\mathrm{IC}_i\}_{i=1}^{L}$ in accordance with priorities determined by the magnitude of eigenvalues.

It should be noted that eigenvalues are derived from the sample data covariance matrix and represents second-order statistics. As a result, on some occasions demonstrated in our experiments, using eigenvectors specified by eigenvalues may not be as effective as other criteria described in the following.

32.3.2.2 High-Order Statistics-Based PICA

The HOS-PICA is to prioritize FastICA-generated ICs whose significance is measured by high-order statistics. Two types of high-order statistics are of major interest: the third-order statistics, referred to as skewness, and the fourth-order statistics, referred to as kurtosis. The algorithm to implement HOS-PICA is summarized as follows.

HOS-PICA algorithm

1. Apply BEP to expand data dimensionality if there is a need such as in the case of OC-ICA.
2. FastICA is used to randomly generate a unit vector as an initial projection vector to produce each of ICs.
3. Calculate the following criterion for IC$_i$ that combines third and fourth orders of statistics for ζ_i:

   $$J(\mathrm{IC}_i) = (1/12)\left[\kappa_i^3\right]^2 + (1/48)\left[\kappa_i^4 - 3\right]^2,\ J(\mathrm{IC}_i) = \kappa_i^3 \text{ or } J(\mathrm{IC}_i) = \kappa_i^4 \tag{32.22}$$

   where $\kappa_i^3 = E\left[\zeta_i^3\right] = \frac{1}{MN}\sum_{n=1}^{MN}\left(z_n^i\right)^3$ and $\kappa_i^4 = E\left[\zeta_i^4\right] = \frac{1}{MN}\sum_{n=1}^{MN}\left(z_n^i\right)^4$ are sample means of third and fourth orders of statistics in the IC$_i$. It should be noted that (32.22) is taken from (5.35) in Hyvarinen et al. (2001), which is used to measure the negentropy by high-order statistics.
4. Prioritize the $\{\mathrm{IC_i}\}_{i=1}^L$ in accordance with the magnitude of *J*(IC$_i$).

32.3.2.3 ATGP-Prioritized PCA

Using ATGP as an initialization algorithm, a set of *L* projection vectors can be used as an initial condition for ICA.

ATGP-PICA algorithm

1. Apply BEP to expand data dimensionality if there is a need such as in the case of OC-ICA.
2. Use ATGP to produce set of projection vectors $\{\mathbf{t}_j\}_{j=1}^L$ arranged in their appearing orders.
3. Use each of $\{\mathbf{t}_j\}_{j=1}^L$ generated in step 1 as an initial projection vector, FastICA produce $\{\mathrm{IC}_i\}_{i=1}^L$ in accordance with priorities determined by order that $\{\mathbf{t}_j\}_{j=1}^L$ were generated by the ATGP.

As a concluding remark, the three PICA-based algorithms presented in this section are derived from different perspectives and have their own merits. The HOS-PICA was developed to prioritize the order of ICs resulting from the use of random initial projection vectors. The process of IC prioritization by HOS-PICA is performed after all the ICs are generated. Unlike HOS-PICA, eigen-PICA and ATGP-PICA do not need to generate all ICs prior to IC prioritization. More specifically, both eigen-PICA and ATGP-PICA do not wait for all ICs to be generated. Instead, they prioritize ICs while generating the ICs. The priorities of ICs are already determined by the order that initial projection vectors are generated by an initialization algorithm, not by a criterion used by HOS-PICA. As a consequence, once the set of initial projection vectors is determined, the IC priority is also determined accordingly.

32.4 Kernel-Based Linear Spectral Mixture Analysis

The BEP presented in Section 32.2.2 is developed to resolve the issue of insufficient band dimensionality so that LSMA techniques such as OSP can still be effective by incorporating expanded images. This section presents a complete opposite approach, called kernel-based LSMA techniques developed in Chapter 15 for MR image analysis. Instead of expanding the original band dimensionality, we introduce nonlinear kernels into LSMA-based classifiers in Section 32.2.1 to make them perform linear decisions in a high dimensional feature space to solve linear non-separable problems in the original MR image data space by mapping the original MRI data space to a new feature space to deal with the issue of linear non-separable features. As a result, the four LSMA-based classifiers, OSP, LSOSP, NCLS and FCLS in Chapter 12 can be further extended to their kernel versions, Kernel Orthogonal Subspace Projection (K-OSP), Kernel LSOSP (K-LSOSP), Kernel Non-negative Constrained Least Squares (K-NCLS) and Kernel Fully Constrained Least Squares (K-FCLS) as discussed in Chapter 15 for hyperspectral imaging and in Section 31.4 of Chapter 31 for multispectral imaging. For details we refer readers to these two chapters. Since OSP and KOSP are originally developed for abundance detection not estimation, they are not of particular interest in PVE. Only three kernel versions of LSMA are studied for MR image experiments.

32.5 Synthetic MR Brain Image Experiments

In this section, a series of experiments were conducted via synthetic images to substantiate the utility of OC-ICA in MR image analysis and to demonstrate its advantages over the traditional ICA.

The synthetic images to be used for experiments in this section are the axial T1, T2, and proton density MR brain images (with 5-mm section thickness, 0% noise, and 0% intensity nonuniformity) resulting from the MR imaging simulator of McGill University, Montreal, Canada (available at www.bic.mni.mcgill.ca/brainweb/). The image volume provided separate volumes of tissue classes, such as CSF, GM, WM, bone, fat, and background. The use of these web MR brain images is to allow researchers to reproduce our experiments for verification. Figure 32.2(a) shows three MR brain images with provided specifications where the first image is obtained by modality = PD, protocol = ICBM, phantom_name = normal, slice_thickness = 5 mm, noise = 0%, INU (intensity nonuniformity) = 0%, the second image by modality = T1, protocol = ICBM, phantom_name = normal, slice_thickness = 5 mm, noise = 0%, INU = 0%, and the third image by modality = T2, protocol = ICBM, phantom_name = normal, slice_thickness = 5 mm, noise = 0%, INU = 0%. Figure 32.2(b) provides the ground truth also available on website www.bic.mni.mcgill.ca/brainweb/ for brain tissue substances in the images in Figure 32.2(a), which will be used to verify the results obtained for our experiments.

Despite the fact that there are many other brain tissue substances such as fat, bone, and blood that also constitute the brain, the three main tissue substances, CSF, GM, and WM are usually of major interest for clinical diagnosis, and these tissue signatures can be generally obtained by their anatomical structures. It has been shown in Hahn and Peitgen (2000) that the skull had significant effects on tissue classification. In this case, the skull signature was specifically included in the signature matrix as a separate and unwanted signature for signature removal. In addition, all signatures other than CSF, GM, WM, and skull were of no interest in brain tissue classification; they could be considered as the background signatures as a whole for background suppression. Accordingly, to implement the LSMA, the signature matrix $\mathbf{M}$ is specified by $\mathbf{M} = [\mathbf{CSF\ GM\ WM\ skull\ B}]$, where the background signature $\mathbf{B}$ was obtained by averaging the signatures of skin, muscle, fat, and glial matter, connective with their training samples extracted in Figure 32.3 according to the ground truth provided in Figure 32.2(b). In the following

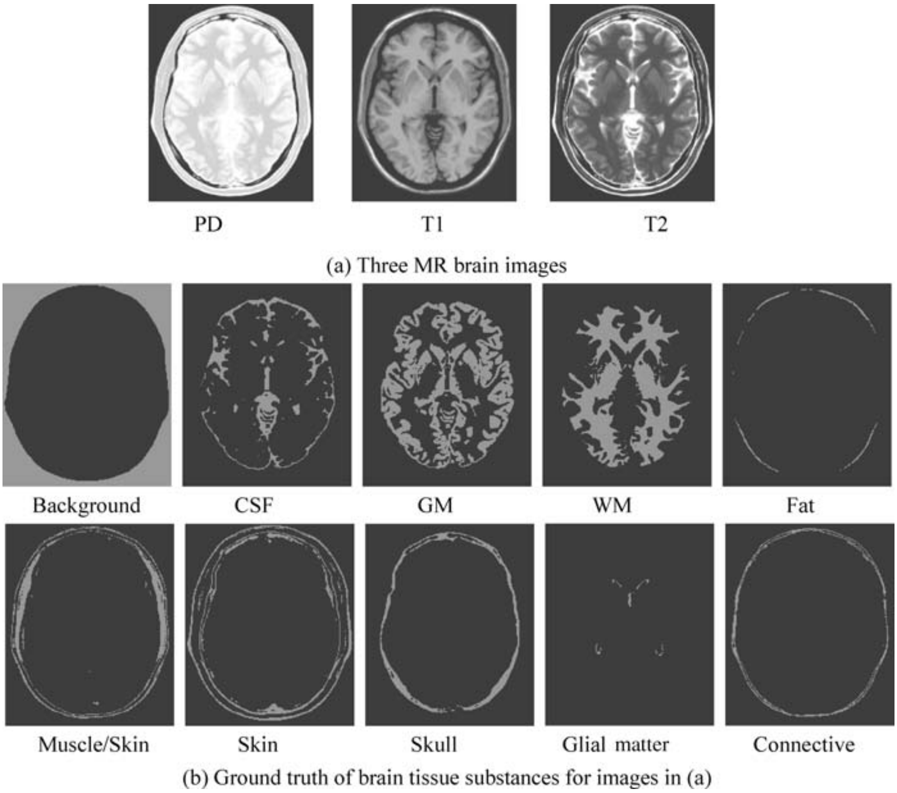

PD
T1
T2
(a) Three MR brain images
Background
CSF
GM
WM
Fat
Muscle/Skin
Skin
Skull
Glial matter
Connective
(b) Ground truth of brain tissue substances for images in (a)

Figure 32.2 Synthetic MR images along with ground truth of brain tissue substances in images.

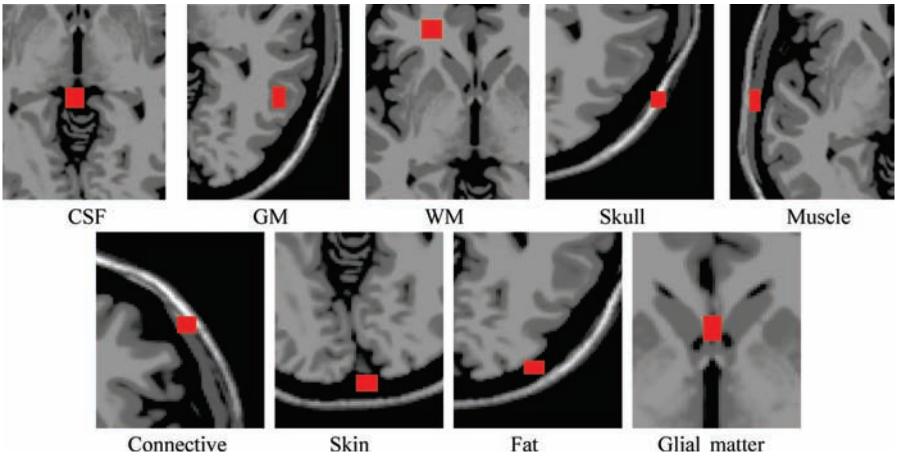


Figure 32.3 Tissues training sample regions for CSF, GM, WM, skull, muscle, connective, skin, fat, and glial matter of synthetic brain MR images.

experiments, LSOSP, NCLS, and FCLS are implemented for spectral unmixing using one of the three major brain tissue substances, CSF, GM, and WM, as the desired signature **d** with other signature plus skull and background signatures as unwanted signatures used for background suppression.

Two experiments were conducted by operating LSMA on (1) the three original MR images shown in Figure 32.2 without using BEP and (2) the three original MR images combined with six BEP-generated bands (auto correlated and cross-correlated bands).

Since it has been shown in Wong (2008) that operating LSMA on the three original MR images combined with three BEP-generated bands (i.e., cross-correlated bands) produced results slightly worse than those combined with six BEP-generated bands (i.e., auto correlated and cross-correlated bands), the figures produced by experiments using three cross-correlated band images are not included here, but rather referred to Wong (2008) to avoid unnecessary duplication.

Experiment 32.1 Operating LSMA on the original three MR images

Figures 32.4–32.7 show classification of CSF, GM, and WM in terms of their unmixed results as PVE produced by the LSOSP, NCLS, and FCLS using the original three MR images with 0%, 5%, 10%, and 20% noise INU fields.

By visual inspection, NCLS produced the best classification in terms of PVE via unmixed results of CSF, GM, and WM in the original MR images. In particular, WM was classified with less false classification compared to LSOSP and FCLS. FCLS was able to classify CSF and GM tissues; however, the background of the MR image was falsely classified into WM. On the other hand, LSOSP showed the worst classification results. Although the important brain tissues CSF, GM, and WM could be distinguished from one another, the contrast of these tissues against the background was not as clear as the constrained techniques, NCLS and FCLS. Furthermore, Figures 32.4–32.7 show that the PVE results were gradually reduced as the intensity of INU fields was increased from 0% to 20%. Visual assessment also showed that the unconstrained method, LSOSP, had less effect from the INU fields than NCLS and FCLS had. By contrast, NCLS was the one that had most effects from the INU fields. For example, the top portion of the GM tissue in

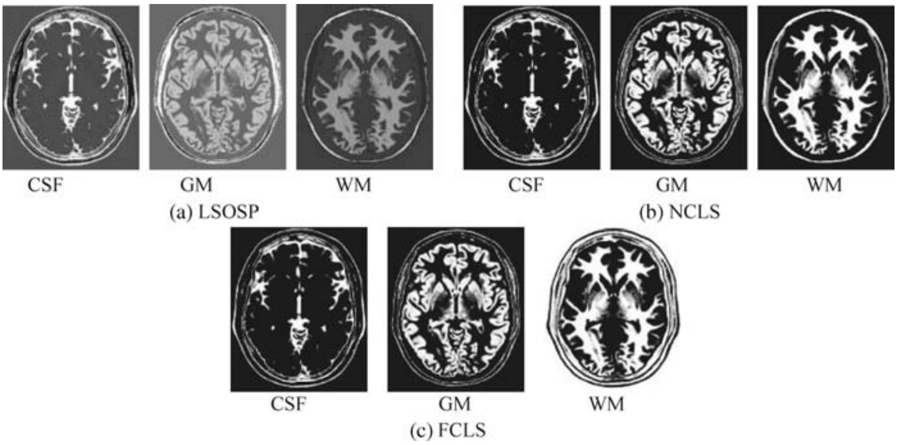


Figure 32.4 Classification of CSF, GM, and WM by their unmixed results in the original synthetic MR images with 0% noise INU fields produced by (a) LSOSP, (b) NCLS, and (c) FCLS.

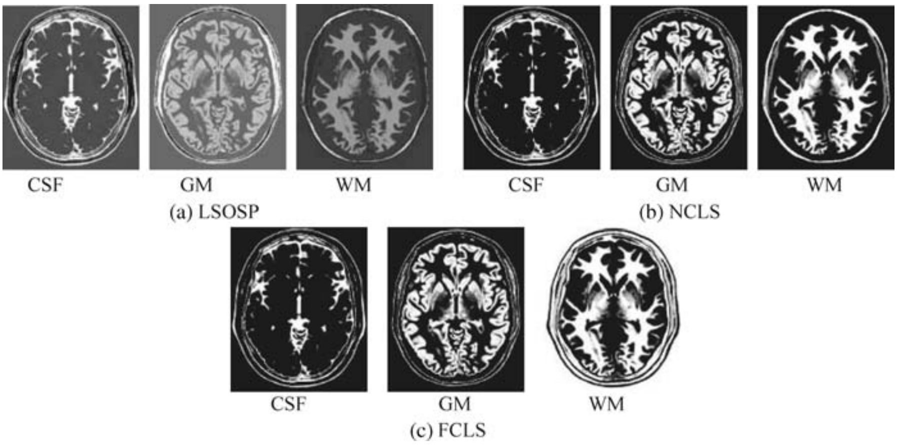


Figure 32.5 Classification of CSF, GM, and WM by their unmixed results in the original synthetic MR images with 5% noise INU fields produced by (a) LSOSP, (b) NCLS, and (c) FCLS.

Figure 32.7(b) could clearly identify the corruption effect resulting from the INU field. In order to conduct a quantitative study of PVE by unmixed abundance fractions, the 3D ROC analysis developed in Chapter 3 was used for performance evaluation. Figures 32.8–32.11 show the mean classification rates of 3D ROC curves along with three respective 2D ROC curves for INU fields of various intensities: 0%, 5%, 10%, and 20%.

As shown in Figures 32.8–32.11, NCLS clearly outperformed the other two methods: LSOSP and FCLS.

Experiment 32.2

Operating LSMA on the three original MR images combined with six BEP-generated bands (autocorrelated and cross-correlated bands) as shown in Figure 32.12 where the signature

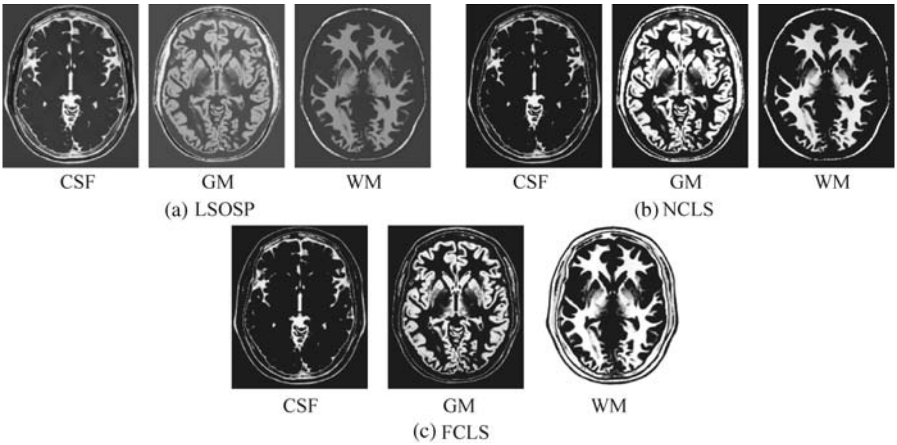
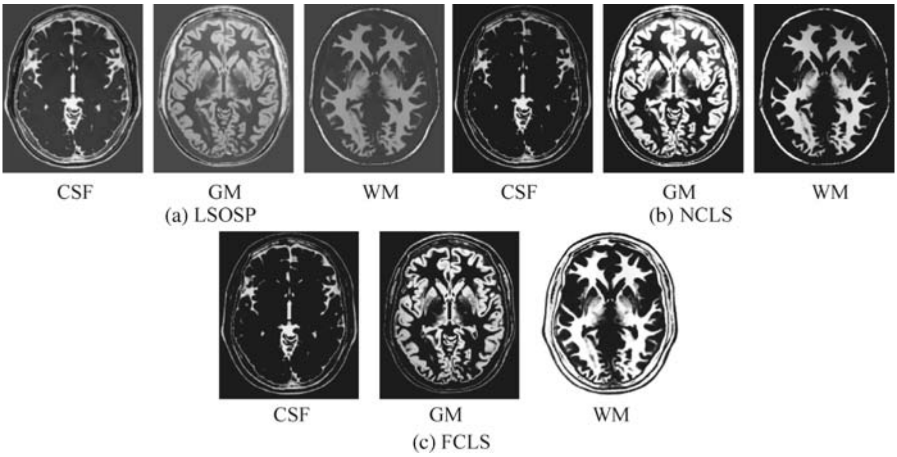


Figure 32.6 Classification of CSF, GM, and WM by their unmixed results in the original synthetic MR images with 10% noise INU fields produced by (a) LSOSP, (b) NCLS, and (c) FCLS.

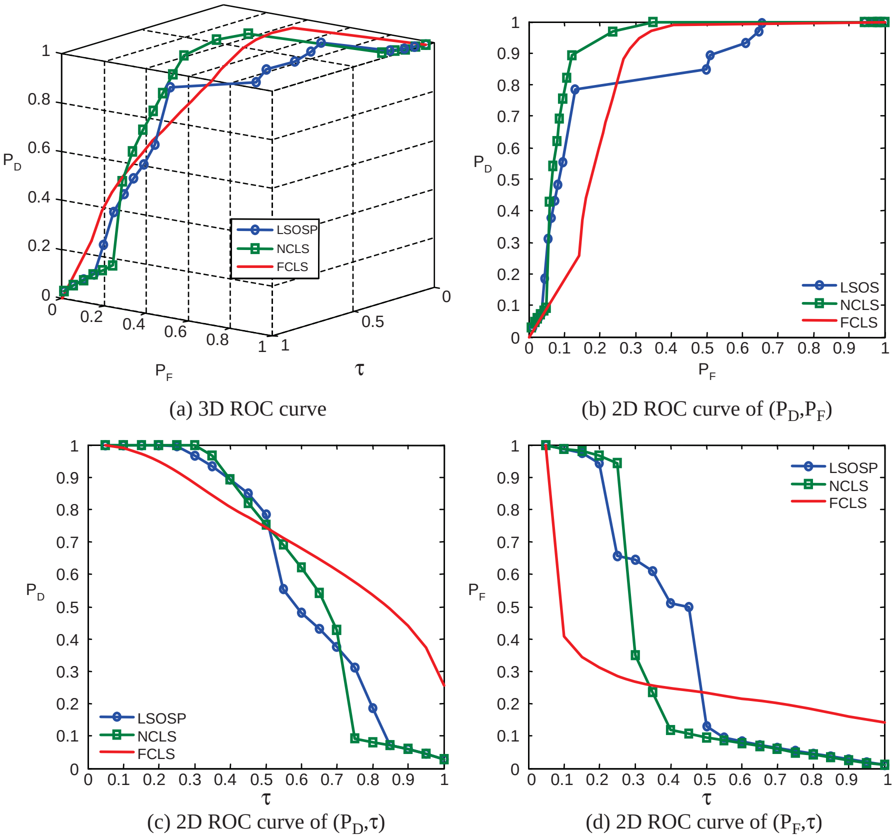


CSF GM WM
(a) LSOSP

CSF GM WM
(b) NCLS

CSF GM WM
(c) FCLS

Figure 32.7 Classification of CSF, GM, and WM by their unmixed results in the original synthetic MR images with 20% noise INU fields produced by (a) LSOSP, (b) NCLS, and (c) FCLS.



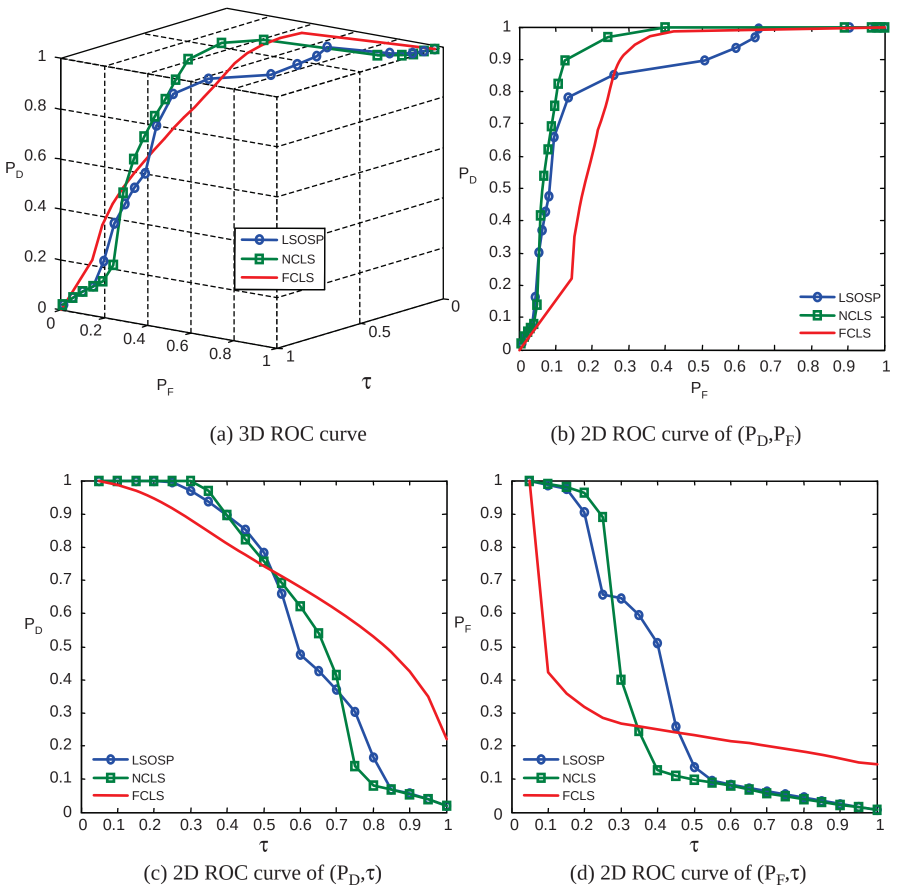
(a) 3D ROC curve

(b) 2D ROC curve of (P_D,P_F)

(c) 2D ROC curve of (P_D,τ)

(d) 2D ROC curve of (P_F,τ)

Figure 32.8 3D ROC analysis of LSOSP, NCLS, and FCLS methods operating original MR images with 0% noise INU fields.



(a) 3D ROC curve

(b) 2D ROC curve of (P_D,P_F)

(c) 2D ROC curve of (P_D,τ)

(d) 2D ROC curve of (P_F,τ)

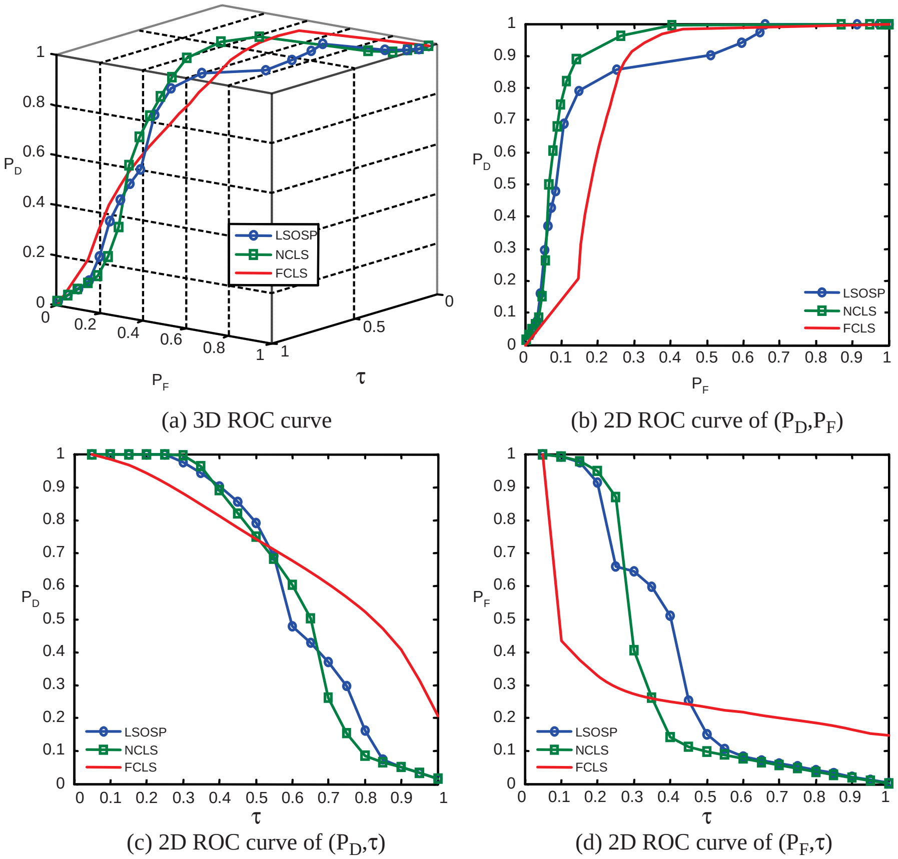
Figure 32.9 3D ROC analysis of LSOSP, NCLS, and FCLS methods operating original MR images with 5% noise INU fields.

matrix $\mathbf{M} = [\mathbf{CSF\ GM\ WM\ skull\ B}]$ is used to unmix CSF, GM, and WM. The noise used was specified by 0%, 5%, 10%, and 20% INU fields.

Following the same treatment conducted for Experiment 32.1, Figures 32.13–32.16 show classification of CSF, GM, and WM in terms of PVE by their unmixed results produced by LSOSP, NCLS, and FCLS using expanded MR images composed of the original three MR images, three autocorrelated, and three cross-correlated bands generated by the BEP with four different noise INU fields of 0%, 5%, 10%, and 20%.

Comparing the results in Figures 32.13–32.16 with those in Figures 32.4–32.7, it was apparent that the BEP did improve LSMA performance in unmixing. This was mainly due to the fact that LSMA techniques were intrapixel multispectral processing techniques that took advantage of extra spectral information provided by autocorrelated and cross-correlated bands via BEP. However, it was not the case shown in Ouyang et al. (2006) where the SVM gained only little benefit from BEP because it was the interpixel-based multispectral classification technique, which was designed to take care of intervoxel spatial correlation not intraspectral correlation.

In order to further conduct a quantitative analysis on performance, the 3D ROC analysis was used for evaluation. These quantification methods show different merits in measuring



(a) 3D ROC curve

(b) 2D ROC curve of (P_D,P_F)

(c) 2D ROC curve of (P_D,τ)

(d) 2D ROC curve of (P_F,τ)

Figure 32.10 3D ROC analysis of LSOSP, NCLS, and FCLS methods operating original MR images with 10% noise INU fields.

classification via unmixed PVE results. The 3D ROC analysis provides a tool to evaluate the relative performance of detection probability to false alarm probability. Figures 32.17–32.20 show the 3D ROC curves along with their three corresponding 2D ROC curves for the results in Figures 32.13–32.16, where FCLS seems to be the best in most cases.

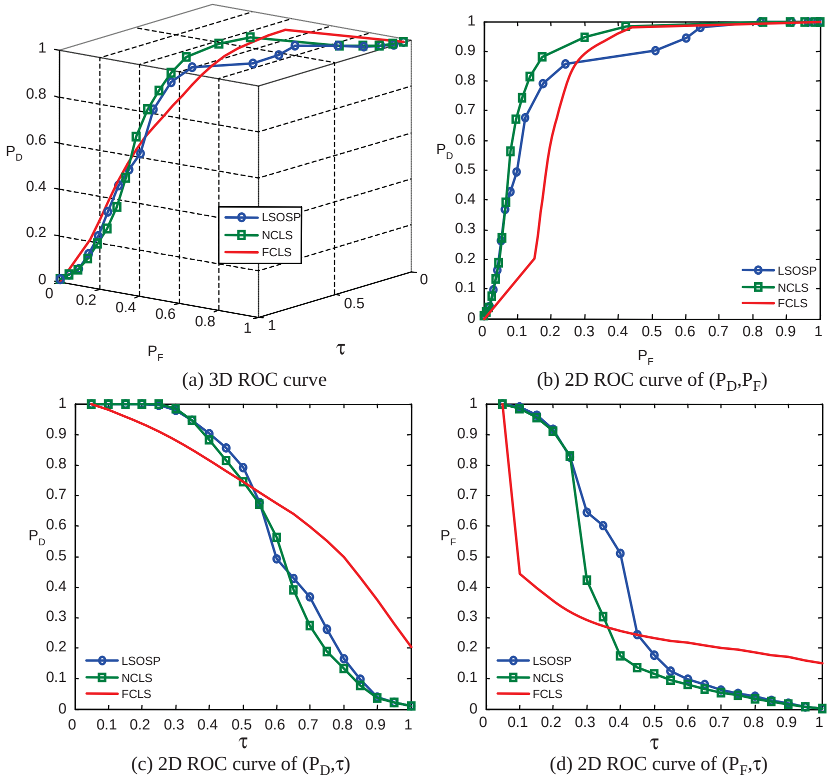
For a further comparison, Table 32.1 also gives A_z of 3D ROC curves for mean classification rates of CSF, GM, and WM produced by LSOSP, NCLS, and FCLS using three bands (original bands) in Figures 32.8–32.11 and nine bands (original plus six BEP-expanded bands) in Figures 32.17–32.20.

From Table 32.1, FCLS remained the best classifier in terms of highest A_z of (P_D,P_F) in all scenarios except the case of 20% noise with a very small fallout. It should be noted that both the curve of (P_D,τ) and the curve of (P_F,τ) generally behaved oppositely. That is, the better the detector, the higher the A_z of (P_D,τ) and the smaller the A_z of (P_F,τ). So, a good detector should have a high A_z of (P_D,P_F) with an appropriate compromise between the A_z of (P_D,τ) and the smaller A_z of (P_F,τ). By taking this into consideration, it seems that FCLS using six BEP-expanded MR images always yielded the best performance.

While the visual inspection of Figures 32.4–32.7 against 32.13–32.16 and Figures 32.8–32.11 against 32.17–32.20 provides qualitative evaluation of relative performance of LSMA with/

Table 32.1 A_z of 3D ROC curves for mean classification rates of CSF, GM, and WM for three-tissue (GM, WM, and CSF) classification of LSOSP, NCLS, and FCLS using three bands (original bands), six bands (original plus three BEP-expanded bands), and nine bands (original plus six BEP-expanded bands).

INU field	Number of bands used	LSOSP ($A_z(P_D,P_F)$, $A_z(P_D,\tau)$, $A_z(P_F,\tau)$)	NCLS ($A_z(P_D,P_F)$, $A_z(P_D,\tau)$, $A_z(P_F,\tau)$)	FCLS ($A_z(P_D,P_F)$, $A_z(P_D,\tau)$, $A_z(P_F,\tau)$)
0% noise	3	(0.6999,0.6589,0.4370)	(0.9104,0.6096,0.3048)	(0.9182,0.6195,0.0929)
	9	(0.89856,0.77758,0.5265)	(0.95263,0.74603,0.3434)	(0.92764,0.70764,0.13637)
5% noise	3	(0.7128,0.6548,0.3996)	(0.9096,0.6181,0.3024)	(0.9205,0.6222,0.0921)
	9	(0.72801,0.86923,0.71575)	(0.85198,0.60066,0.30038)	(0.89822,0.51055,0.071433)
10% noise	3	(0.7815,0.6284,0.3422)	(0.9036,0.6217,0.3062)	(0.9219,0.6208,0.0915)
	9	(0.81937,0.85872,0.708)	(0.86388,0.57327,0.28478)	(0.8983,0.50266,0.069633)
20% noise	3	(0.8550,0.6847,0.3614)	(0.8930,0.6249,0.2966)	(0.9240,0.6167,0.0905)
	9	(0.88367,0.78043,0.53958)	(0.95052,0.74283,0.36173)	(0.94355,0.57861,0.14488)



(a) 3D ROC curve

(b) 2D ROC curve of (P_D,P_F)

(c) 2D ROC curve of (P_D,τ)

(d) 2D ROC curve of (P_F,τ)

Figure 32.11 3D ROC analysis of LSOSP, NCLS, and FCLS methods operating the original MR images with 20% noise INU fields.

Table 32.2 Quantification least-squares error (QLSE) of abundance estimated of CSF, GM, and WM for three-tissue (GM, WM, and CSF) classification of LSOSP, NCLS, and FCLS using three bands (original bands) and nine bands (original plus six BEP-expanded bands)

Noise INU field (%)	Number of bands used	LSOSP	NCLS	FCLS
0	3	0.124253	0.0989178	0.139521
	9	0.237682	0.1060890	0.0234141
5	3	0.120411	0.0987043	0.140209
	9	0.230962	0.1000280	0.0237175
10	3	0.120253	0.0993571	0.141210
	9	0.246652	0.1034280	0.0252616
20	3	0.121424	0.1001290	0.144188
	9	0.299590	0.0986234	0.0320658

without BEP, another objective measure, called quantification least-squares error (QLSE) defined by (32.17), calculated the error of abundances between the estimated abundance fractions as PVE and their corresponding ground truth values. Table 32.2 gives QLSE of three-tissue (GM,WM, and CSF) classification produced by LSOSP, NCLS, and FCLS using three bands (original bands) and nine bands (original plus six BEP-expanded bands).

According to Table 32.2, the QLSE performance of LSMA techniques using BEP was worse compared to their counterparts without using BEP when no or only partial abundance constraints were imposed. However, the QLSE was significantly improved when abundance constraints were fully imposed, as shown in the last column of FCLS in Table 32.2. In addition, the upper rows produced by Experiment 32.4.1.1 in Table 32.1 gives QLSE obtained from Figures 32.3–32.6 where it is interesting to note that FCLS yielded the largest error compared to LSOSP and NCLS. However, the results were reversed as shown in the lower rows produced by Experiment 32.2 in Table 32.2 when nine expanded MR images were used including the original three MR images

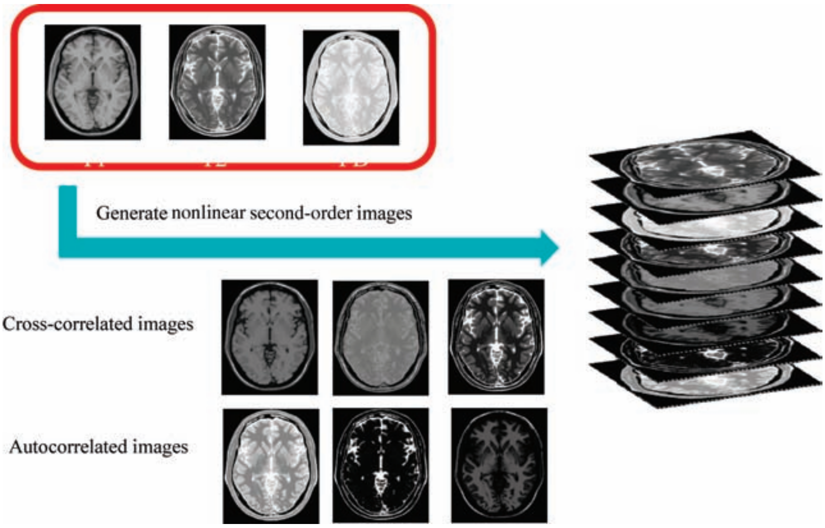


Figure 32.12 An example of nine-band image cube obtained from three origin MR images, three cross-correlated, and three autocorrelated band images.

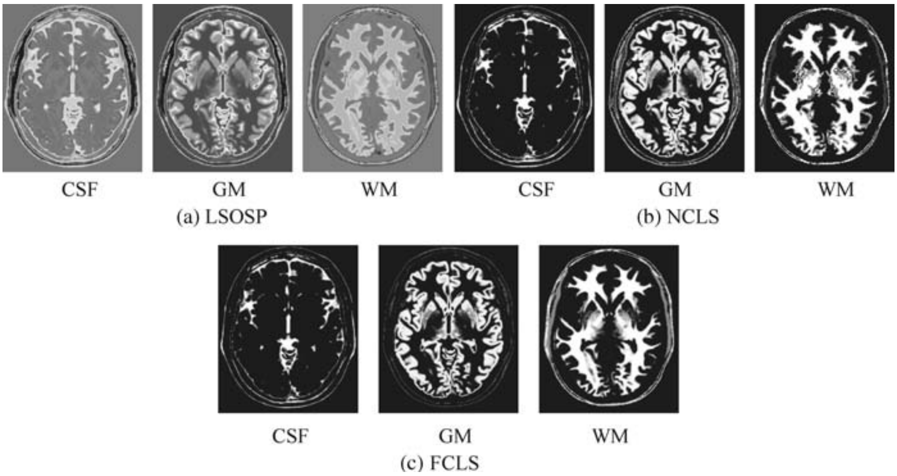


Figure 32.13 Classification of CSF, GM, and WM by their unmixed results using BEP-expanded nine bands (T1, T2, PD, three cross-correlation BEP, and three auto-correlation BEP) with 0% noise INU field produced by (a) LSOSP, (b) NCLS, and (c) FCLS with five substances (CSF, GM, WM, skull, and background).

combined with six BEP-expanded bands (autocorrelated and cross-correlated). Nevertheless, in both experiments, the performance of NCLS was right in between.

Prior to performing kernel-based LSMA, the prior knowledge of tissue signatures (**M**) was obtained from the areas shown in Figure 32.3 based on provided ground truth in Figure 32.2 (b) where only four tissues of interests, CSF, GM, WM, and skull were obtained by averaging the training samples of these four signatures in Figure 32.3 and the fifth signature was obtained by averaging the training samples of other signatures in Figure 32.3. Based on the study in Wong (2008), this five-signature matrix (**M**) usually provided best classification results. Once the signature matrix (**M**) was obtained, both the original 0% and 20% INU

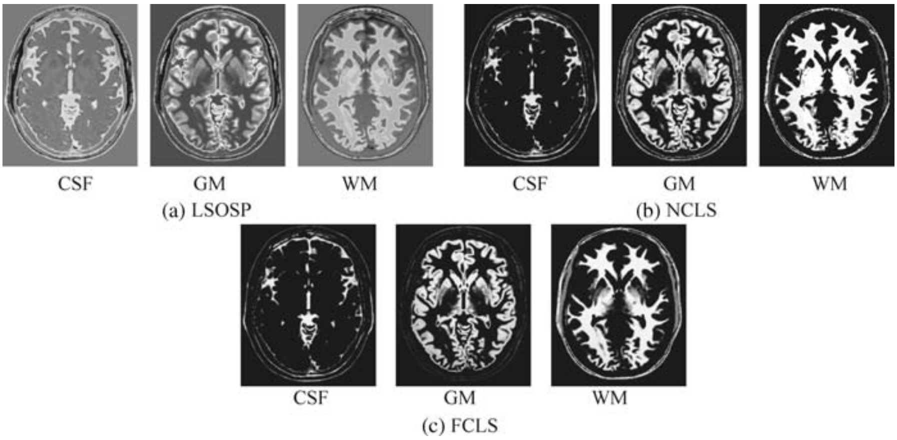


Figure 32.14 Classification of CSF, GM, and WM by their unmixed results using BEP-expanded nine bands (T1, T2, PD, cross-correlation BEP, and autocorrelation BEP) with 5% noise INU field produced by (a) LSOSP, (b) NCLS, and (c) FCLS with five substances (CSF, GM, WM, skull, and background).

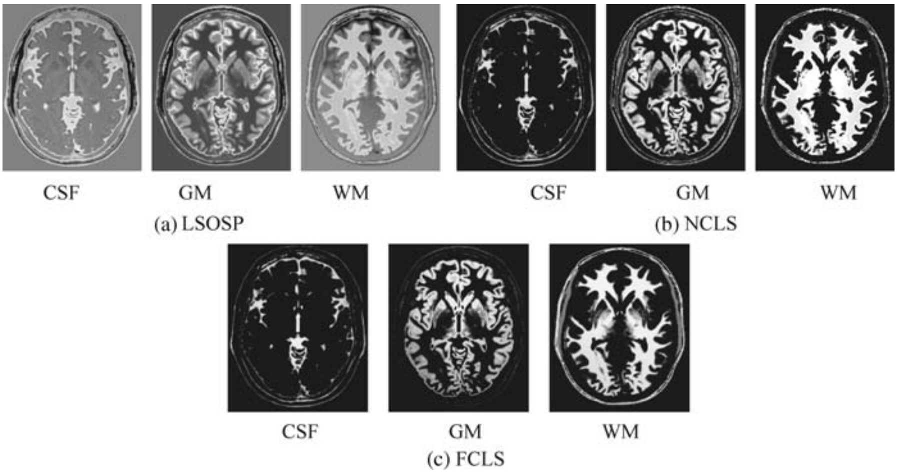

CSF GM WM
(a) LSOSP
CSF GM WM
(b) NCLS
CSF GM WM
(c) FCLS

Figure 32.15 Classification of CSF, GM, and WM by their unmixed results using BEP-expanded nine bands (T1, T2, PD, cross-correlation BEP, and autocorrelation BEP) with 10% noise INU field produced by (a) LSOSP, (b) NCLS, and (c) FCLS with five substances (CSF, GM, WM, skull, and background).

synthetic brain MR images were classified using *kernel-based LSMA* techniques, K-LSOSP, K-NCLS, and K-FCLS presented in Section 32.4. In addition, following a similar treatment to that carried out in Section 31.5 experiments were also conducted by combining the BEP-preprocessed MR images with LSOSP, NCLS, and FCLS and their kernel counterparts K-BEP-LSOSP, K-BEP-NCLS, and K-BEP-FCLS to see if the BEP could improve the performance. As noted earlier, only the cross-correlated band images were used since they provided sufficient spectral information for LSMA according to Wong (2008). Besides, the radial basis

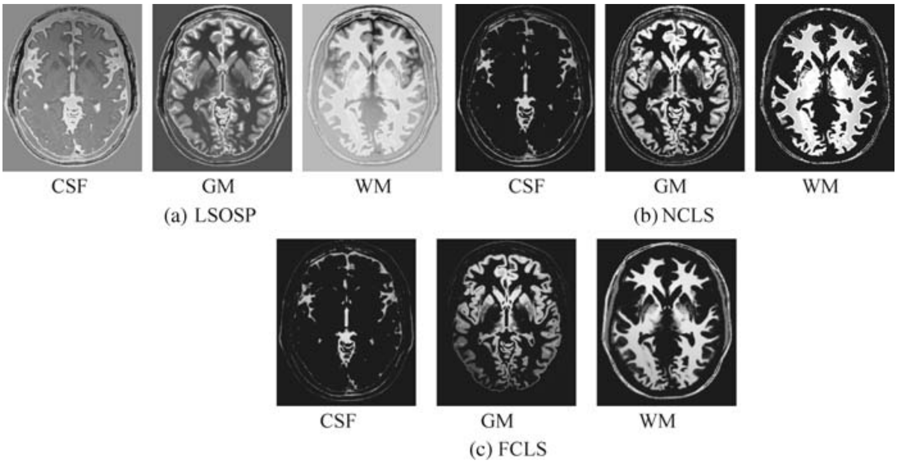
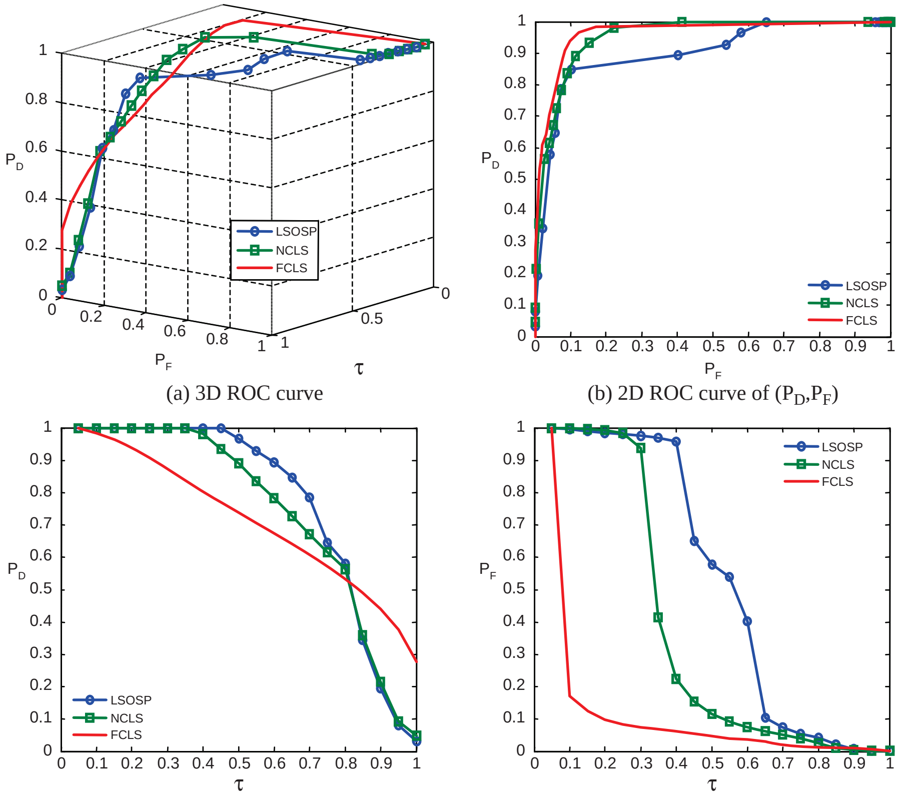

CSF GM WM
(a) LSOSP
CSF GM WM
(b) NCLS
CSF GM WM
(c) FCLS

Figure 32.16 Classification of CSF, GM, and WM by their unmixed results using BEP-expanded nine bands (T1, T2, PD, cross-correlation BEP, and autocorrelation BEP) with 20% noise INU field produced by (a) LSOSP, (b) NCLS, and (c) FCLS with five substances (CSF, GM, WM, skull, and background).



(a) 3D ROC curve

(b) 2D ROC curve of (P_D,P_F)

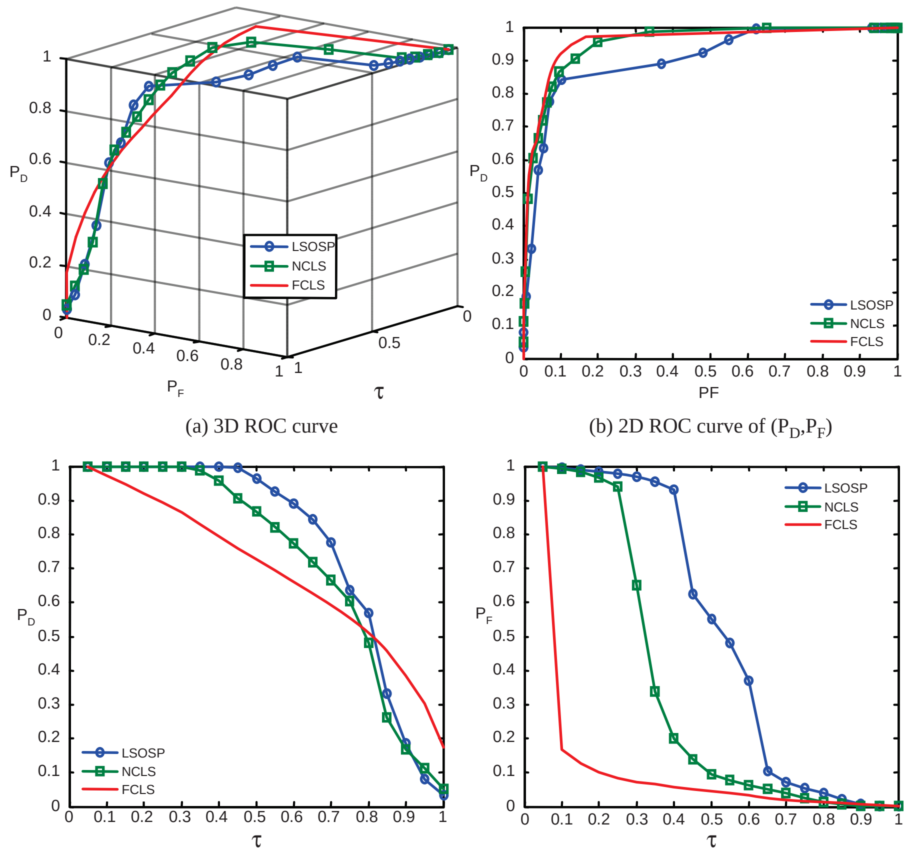
(c) 2D ROC curve of (P_D,τ)

(d) 2D ROC curve of (P_F,τ)

Figure 32.17 3D ROC analysis of LSOSP, NCLS, and FCLS methods operating nine expanded MR images with 0% noise INU fields.

function (RBF) has been widely used in kernel-based methods. Hence, it was used in our test kernel-based LSMA in which case only the parameter σ needed to be appropriately determined. Since the range of the parameter σ varies with different kernel-based LSMA methods, in order to make comparable analysis, each MR image band was normalized to the range of [0,1] so that the selection of the parameter σ could be made between 0 and 1. As a result, the value of the σ was selected empirically to achieve the best possible visual classification in which case for each kernel-based LSMA method, $\sigma = 0.1$ for K-LSOSP, $\sigma = 0.1$ for K-NCLS, $\sigma = 0.05$ for K-FCLS, $\sigma = 0.1$ for K-BEP-LSOSP, $\sigma = 0.1$ for K-BEP-NCLS, and $\sigma = 0.05$ for K-BEP-FCLS. Figure 32.21 shows respectively their classification results obtained by using LSMA-unmixed abundance fractions as their PVE for the cases of 20% INU field-corrupted MR images classification. Since CSF, GM, and WM were the main tissue of interest, other classification results were not included in the figures. As mentioned earlier, same conclusions could also be drawn for both 0% INU and 20% INU noise corruption. For this reason, only the 20% INU noise-corrupted MR brain image PVE results are shown in the following.

Some noteworthy observations can be made on Figure 32.21 by visual inspection. First, kernel-based approaches greatly improved their counterparts without using kernels. For example, it was easily observed by comparing Figure 32.21(a) with Figure 32.21(b) where the LSOSP approach was not able to distinguish CSF, GM, and WM tissues, but K-LSOSP was able to clearly classify



(a) 3D ROC curve

(b) 2D ROC curve of (P_D,P_F)

(c) 2D ROC curve of (P_D,τ)

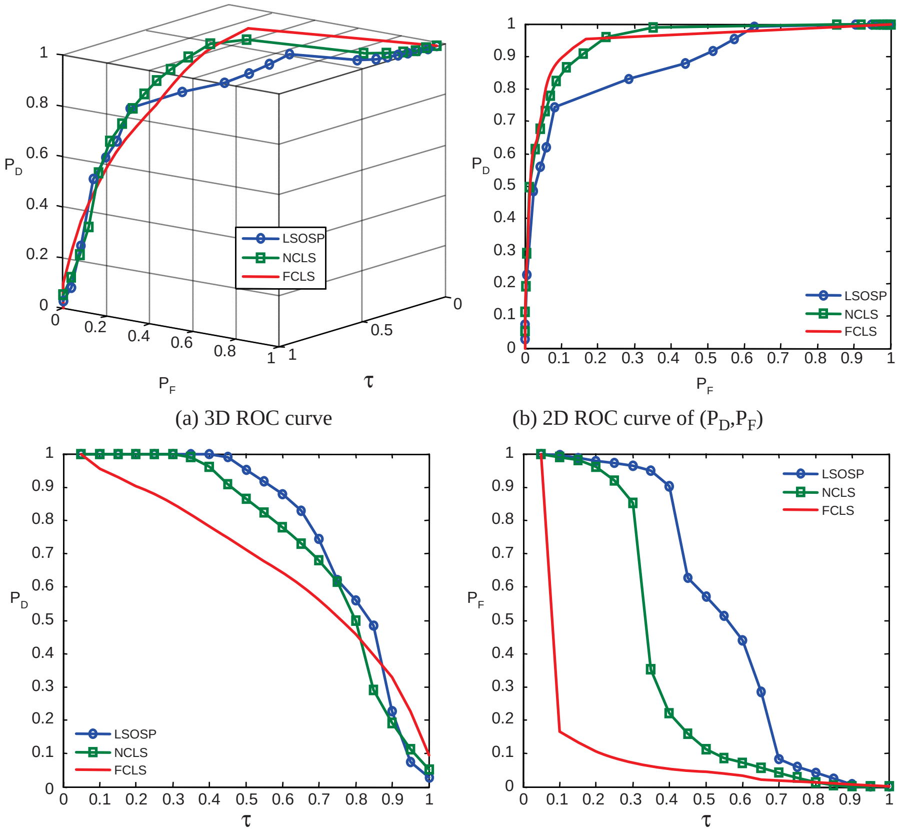
(d) 2D ROC curve of (P_F,τ)

Figure 32.18 3D ROC analysis of LSOSP, NCLS, and FCLS methods operating nine expanded MR images with 5% noise INU fields.

these tissues. Moreover, the K-LSOSP PVE results shown in Figure 32.21(a) produced the least false positive classification. In particular, the CSF and GM PVE results showed little false positive classification.

On the other hand, K-NCLS and K-FCLS produced better PVE results by classification for the three main brain tissues shown in Figure 32.21(b) and (c) than K-LSOSP. However, they also suffered from higher false positive detection. For example, muscle tissue was misclassified into WM as presented in Figure 32.21(b) and (c), and skin/connectives were falsely classified into GM. In the case of K-BEP-LSMA results, K-BEP-LSOSP and K-BEP-NCLS methods provided better results than K-BEP-FCLS. Moreover, the INU noise corruption effect indeed showed significant impact on the methods in Figure 32.21(a) and (b). Furthermore, a great improvement could be observed from most of the kernel-based LSMA PVE results as presented in Figure 32.21(d)–(i). Nevertheless, the INU corruption effect was still visible for K-LSOSP PVE results in Figure 32.21(d), which indicated that the K-LSOSP was still sensitive to INU corruption.

Since Figure 32.2(b) provides the ground truth of brain tissues used to simulate the MR images, this allows us to conduct a quantitative analysis among various proposed LSMA-based methods,



(a) 3D ROC curve

(b) 2D ROC curve of (P_D,P_F)

(c) 2D ROC curve of (P_D,τ)

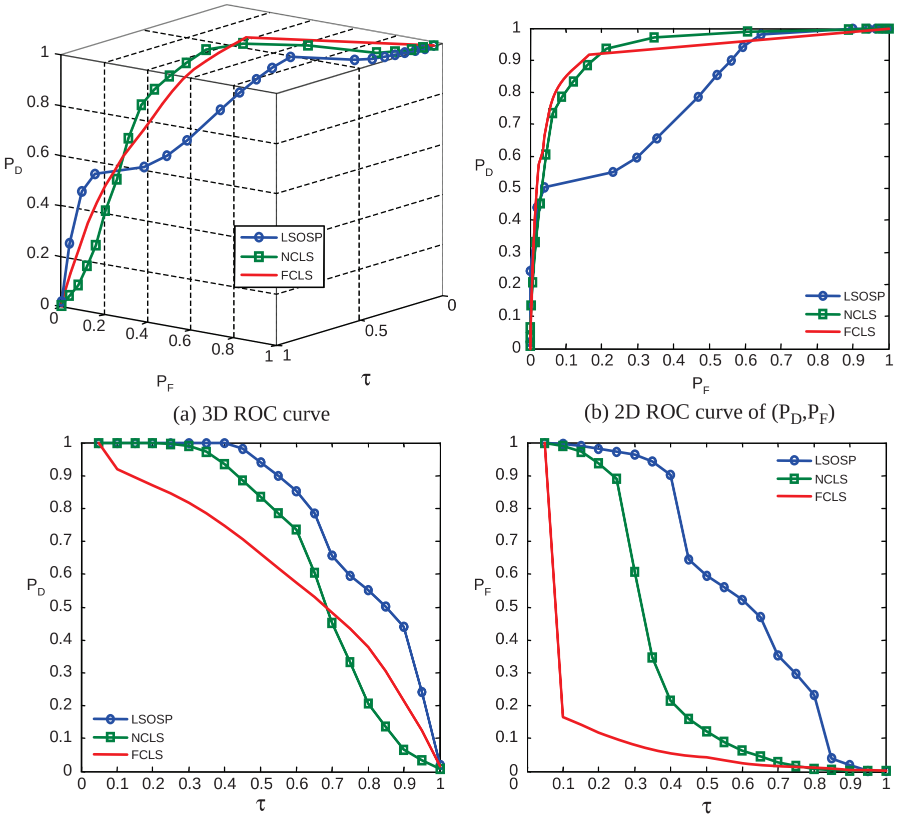
(d) 2D ROC curve of (P_F,τ)

Figure 32.19 3D ROC analysis of LSOSP, NCLS, and FCLS methods operating nine expanded MR images with 10% noise INU fields.

BEP-LSOSP, K-LSOSP, K-BEP-LSOSP, BEP-NCLS, K-NCLS, K-BEP-NCLS, BEP-FCLS, K-FCLS, and K-BEP-FCLS. Figure 32.22 shows 3D ROC curves (P_D versus P_F versus τ) along with three 2D ROC curves (P_D versus P_F, P_D versus τ, and P_F versus τ) of these methods plotted for the cases of 20% INU where (a–d), (e–h), and (i–l) are plots generated by LSOSP, NCLS, and FCLS methods, respectively. The ROC analyses of the 0% and 20% INU noise-corrupted brain MR images are again identical. Hence, only the 20% INU results are presented.

In addition to the ROC curves, the area under curve (A_z) value was calculated for each of 2D ROC curves. It is worth noting that the best performance of A_z for the overall detection (P_D versus P_F) and true positive detection (P_D versus τ) is 1, while the highest false positive performance (P_F versus τ) is 0. With this in mind, the resulting A_z values were calculated for each method and tabulated in Table 32.3 where the bold-faced numbers represent the best performance values under each classification criterion (overall, true positive, and false positive detection). In addition, the 0% INU noise corrupted is included in this table because the benefits of kernel-based techniques can be easily observed.

As shown in Table 32.3, all the kernel-based techniques produced better overall PVE than the non-kernel-based techniques. In particular, the overall PVE based on A_z under the 2D ROC curve of P_D versus P_F suggested that K-LSMA increased correct classification rates while also decreasing



(a) 3D ROC curve

(b) 2D ROC curve of (P_D,P_F)

(c) 2D ROC curve of (PD,τ)

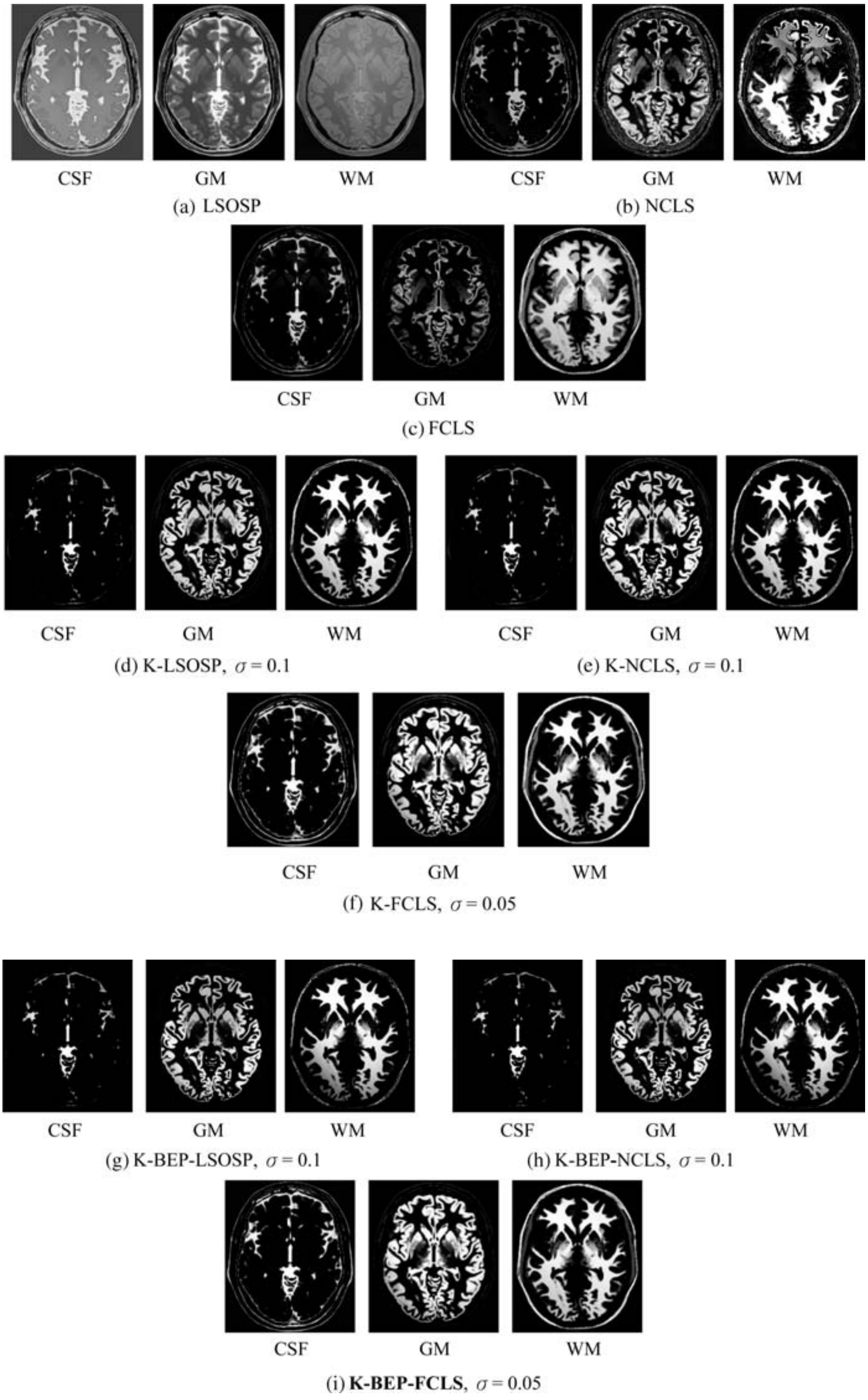
(d) 2D ROC curve of (PF,τ)

Figure 32.20 3D ROC analysis of LSOSP, NCLS, and FCLS methods operating nine expanded MR images with 20% noise INU fields.

false positive detection rates. Also, based on the values of P_D versus τ and P_F versus τ, the K-LSMA performed better than the non-kernel-based LSMA in terms of higher true positive rate and lower false positive rate.

If the true positive rate were the main criteria of concern, K-NCLS and K-FCLS techniques would had demonstrated superior performances for 0% and 20% INU-corrupted MR images experiments. Interestingly, LSMA combining BEP with kernels improved true positive performance, but it also introduced more distortions to the MR images and thus, generally speaking, they did not outperform K-NCLS and K-FCLS. Furthermore, kernel-based techniques did not find a substantial amount of improvement in the case of 0% INU-distorted MR image experiment. But, they did significantly improve the performance for 20% INU MR PVE. This strongly suggested that kernel-based techniques could suppress noise effects when MR images are heavily corrupted by unknown noise.

In addition, according to Table 32.3, K-LSMA could improve performance by reducing false positive rate. For example, K-LSOSP yielded the lowest A_z values for both cases of 0% and 20% INU-distorted MR images, while BEP-LSOSP showed the worst false positive performance. Furthermore, The K-BEP-NCLS and K-BEP-FCLS outperformed K-NCLS and K-FCLS. However, K-BEP-LSOSP did not show any advantage over K-LSOSP. Similar to true positive performance, little improvement of K-LSMA was observed over the non-kernel-based LSMA for the 0% INU



(a) LSOSP

(b) NCLS

(c) FCLS

(d) K-LSOSP, $\sigma = 0.1$

(e) K-NCLS, $\sigma = 0.1$

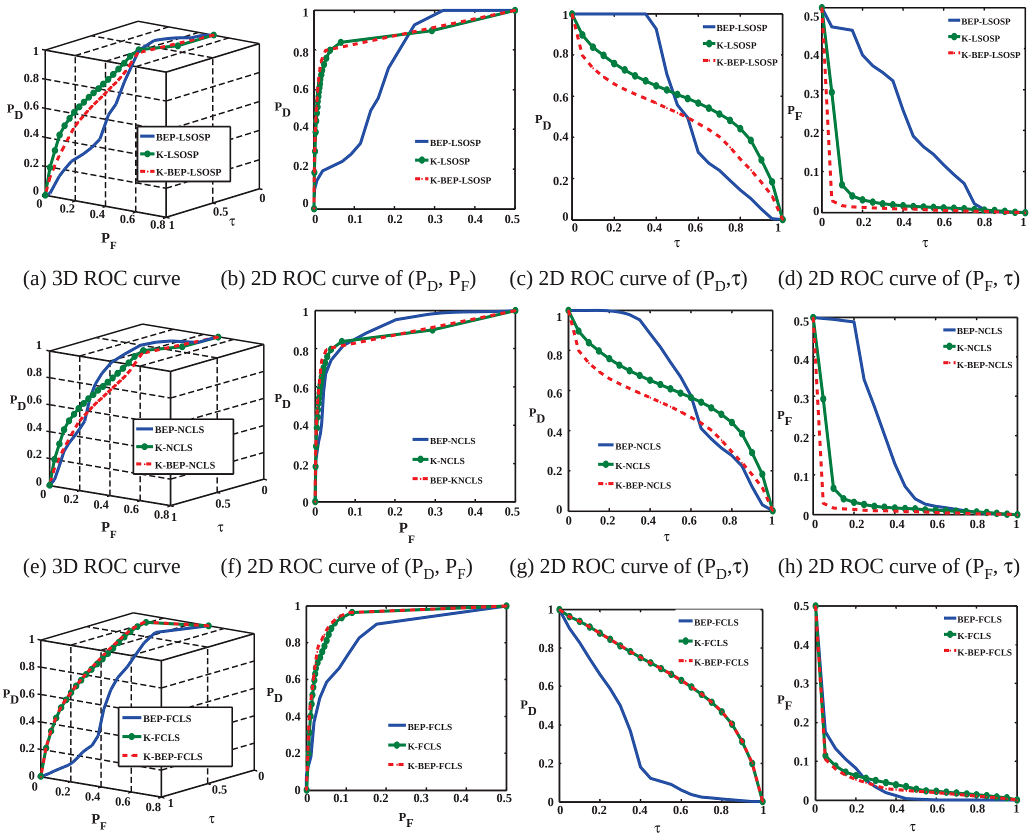
(f) K-FCLS, $\sigma = 0.05$

(g) K-BEP-LSOSP, $\sigma = 0.1$

(h) K-BEP-NCLS, $\sigma = 0.1$

(i) **K-BEP-FCLS**, $\sigma = 0.05$

Figure 32.21 Classification results of three main brain tissues with 20% INU field produced by (a) BEP-LSOSP, (b) BEP-NCLS, (c) BEP-FCLS, (d) K-LSOSP, (e) K-NCLS, (f) K-FCLS, (g) K-BEP-LSOSP, (h) K-BEP-NCLS, and (i) K-BEP-FCLS with five substances (CSF, GM, WM, skull, and background).



(a) 3D ROC curve (b) 2D ROC curve of (P_D, P_F) (c) 2D ROC curve of (P_D,τ) (d) 2D ROC curve of (P_F, τ)

(e) 3D ROC curve (f) 2D ROC curve of (P_D, P_F) (g) 2D ROC curve of (P_D,τ) (h) 2D ROC curve of (P_F, τ)

(i) 3D ROC curve (j) 2D ROC curve of (P_D, P_F) (k) 2D ROC curve of (P_D,τ) (l) 2D ROC curve of (P_F, τ)

Figure 32.22 3D ROC, P_D versus P_F, P_D versus τ, and P_F versus τ curves for 20% INU noise-corrupted MR images with (a)–(d) generated by BEP-LSOSP, K-LSOSP, and K-BEP-LSOSP, (e)–(h) generated by BEP-NCLS, K-NCLS, and K-BEP-NCLS, and (i)–(l) generated by BEP-FCLS, K-FCLS, and K-BEP-FCLS.

MR image experiment. On the contrary, false positive rate was greatly reduced for the 20% INU-distorted MR images, which suggested that K-LSMA improved non-kernel-based LSMA in heavily noise-corrupted MR image.

Finally, as for overall performance, the values of P_D versus P_F in Table 32.3 suggest that K-BEP-FCLS produced the best PVE. In general, K-LSMA outperformed non-kernel-based LSMA in terms of overall performance. In addition, K-BEP-LSOSP and K-BEP-FCLS showed better PVE results than their kernel-based counterparts. However, it was not true for K-BEP-NCLS, which did not perform better than K-NCLS. Also, K-LSMA showed little advantages in the 0% INU-distorted MR image experiments, but it did improve performance significantly for 20% INU-distorted MR image experiments.

In summary, both kernel-based LSMA (i.e., K-LSOSP, K-NCLS, and K-FCLS) and kernel-BEP-based LSMA (i.e., K-BEP-LSOSP, K-BEP-NCLS, and K-BEP-FCLS) significantly outperformed non-kernel-based LSMA (i.e., LSOSP, NCLS, and FCLS). Moreover, according to the above 3D analysis, the kernel-based LSMA showed great improvement in the true positive rate, false positive rate, and overall performance over non-kernel-based LSMA. If the kernel-based LSMA were further extended by combining the BEP, only slight improvement was observed.

Table 32.3 Area under curve (A_Z) calculated under P_D versus P_F, P_D versus τ, and P_F versus τ curves of BEP-LSMA, K-LSMA, and K-BEP-LSMA

P_D versus P_F

	BEP-LSOSP	BPE-NCLS	BEP-FCLS	K-LSOSP	K-NCLS	K-FCLS	K-BEP-LSOSP	K-BEP-NCLS	K-BEP-FCLS
0% noise INU field	0.757447	0.91195	0.937481	0.805387	0.923316	0.953130	0.869127	0.869127	**0.95887**
20% noise INU field	0.733262	0.856128	0.810244	0.813058	0.923988	0.942708	0.865991	0.865991	**0.948342**

P_D versus τ

	BEP-LSOSP	BPE-NCLS	BEP-FCLS	K-LSOSP	K-NCLS	K-FCLS	K-BEP-LSOSP	K-BEP-NCLS	K-BEP-FCLS
0% noise INU field	0.616633	0.543251	0.632599	0.453731	**0.712828**	0.679058	0.663633	0.663633	0.669978
20% noise INU field	0.584726	0.557835	0.483296	0.383323	0.652459	**0.660144**	0.655297	0.655297	0.634873

P_F versus τ

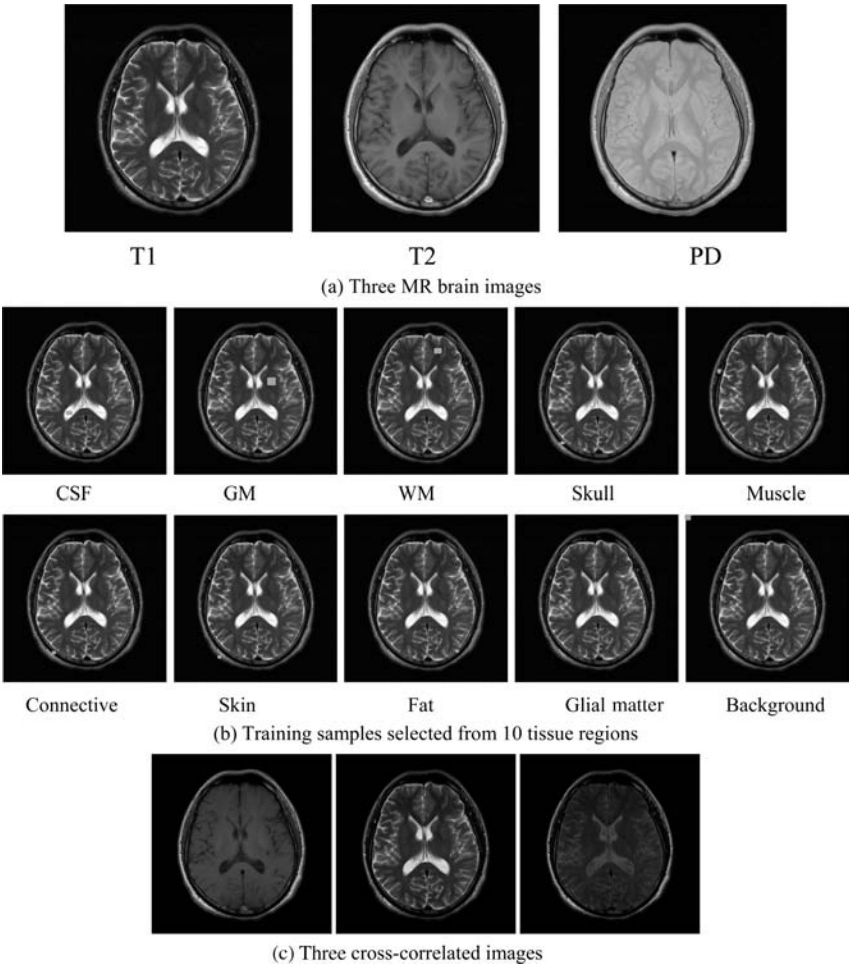
	BEP-LSOSP	BPE-NCLS	BEP-FCLS	K-LSOSP	K-NCLS	K-FCLS	K-BEP-LSOSP	K-BEP-NCLS	K-BEP-FCLS
0% noise INU field	0.386904	0.265387	0.0968043	**0.0344685**	0.242546	0.0905744	0.152254	0.152254	0.07266
20% noise INU field	0.388415	0.291541	0.109294	**0.0314984**	0.228361	0.0877941	0.187954	0.187954	0.0647801

However, without using kernels, the BEP did indeed help LSMA improve performance significantly as reported in Wong (2008). Finally, all the experiments conducted in this section demonstrated that the greatest advantage of the K-LSMA is the ability of using kernels in improving performance when the MR image is corrupted by high INU noise.

This benefit suggested that the KLSMA can be a promising technique in applications of MR image analysis.

32.6 Real MR Brain Image Experiments

The real MR images used for experiments were obtained in the TaiChung Veterans General Hospital (TCVGH) to further validate the utility of the proposed LSMA-based intrapixel techniques in real images in calculating partial volumes of brain tissues. The MR images shown in Figure 32.23 (a) were obtained from one normal volunteer by a whole body 1.5-T MR system (Sonata, Siemens, Erlangen, Germany). The routine brain MR protocol consisted of axial spin echo T1 weighted



(a) Three MR brain images

(b) Training samples selected from 10 tissue regions

(c) Three cross-correlated images

Figure 32.23 Three real MR images in (a) along with training samples in (b).

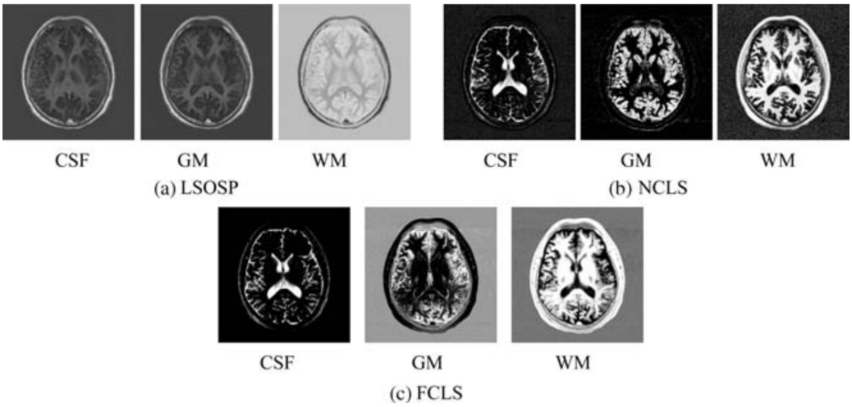


Figure 32.24 Classification results of the original real MR images produced by (a) LSOSP, (b) NCLS, and (c) FCLS.

images (TR/TE = 400/9 ms), proton density image (TR/TE = 4000/10 ms), and T2 weighted images (TR/TE = 4000/91 ms). Other imaging parameters included for this experiment were slice thickness = 6 mm, matrix = 256 × 256, field of view (FOV) = 24 cm, and number of excitations (NEX) = 2. These images are shown in Figure 32.23(a). Training sample sets for 10 classes were selected according to their anatomical structures shown in Figure 32.23(b).

By visual inspection of Figures 32.23–32.26, the PVE results produced by all the techniques separated the three major substances clearly in general. However, there were also some supposedly WM substances showing in GM-generated images by all the techniques. Moreover, similar to the synthetic brain image experiment, OSP and LSOSP produced similar estimated abundances and so did NCLS and FCLS. Another observation is that the skull showed up in the WM images generated by all the techniques. This suggested that using a skull stripping as a preprocessing to remove the skull prior to classification may actually improve PVE results.

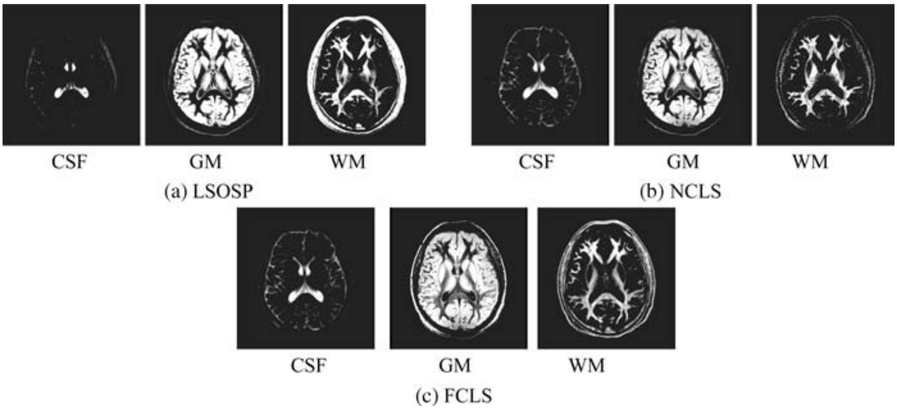


Figure 32.25 Classification results of the three main brain tissues with BEP-expanded six bands (T1, T2, PD, and cross-correlation BEP) produced by (a) OSP, (b) LSOSP, (c) NCLS, and (d) FCLS techniques with five substances (CSF, GM, WM, skull, and background).

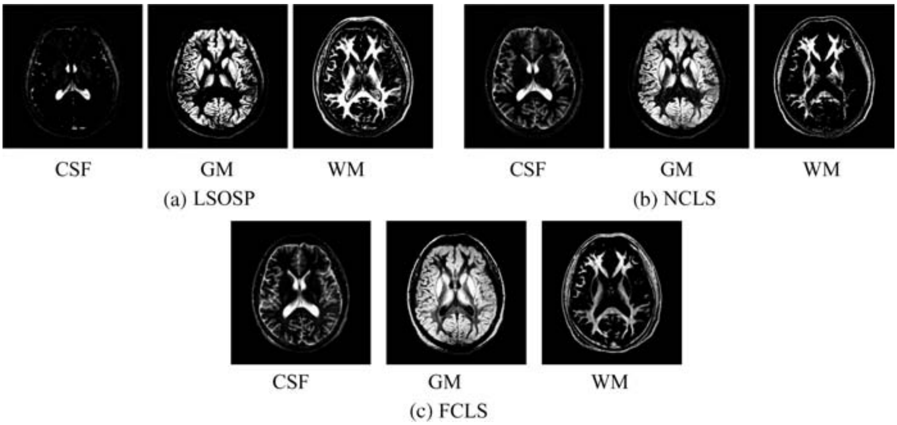


Figure 32.26 Classification results of three main brain tissues with BEP expanded nine bands (T1, T2, PD, cross-correlation BEP, and auto-correlation BEP) produced by (a) LSOSP, (b) NCLS, and (c) FCLS with five substances (CSF, GM, WM, skull, and background).

With this image dataset including three original images in Figure 32.23(a) and three additional BEP-generated images in Figure 32.23(c), the same experiments conducted for the synthetic images in Section 32.5 were repeated where the training samples from GM, WM, CSF, and skull were selected by an experienced radiologist, as shown in Figure 32.27.

In addition to the four tissues in Figure 32.27, a fifth tissue signature used as a background signature was obtained by taking the average over other unwanted tissue signatures plus the actual background of MR images to form a five-signature matrix **M** to be used for LSMA to unmix the three main brain tissues of interest, CSF, GM, and WM as their PVEs. Figures 32.28(a)–(c), 32.28 (d)–(f), and 32.28(g)–(i) show the PVE results of CSF, GM, and WM using three non-kernel-based LSMA methods (LSOSP, NCLS, and FCLS), three K-LSMA methods (K-LSOSP, K-NCLS, and K-FCLS), and three K-BEP-LSMA methods (K-BEP-LSOSP, K-BEP-NCLS, and K-BEP-FCLS), respectively.

The parameter σ used in the RBF kernels by K-LSMA was emperically chosen by $\sigma = 0.1$ for K-LSOSP, K-BEP-LSOSP, K-NCLS, and K-BEP-NCLS and $\sigma = 0.05$ for K-FCLS and K-BEP-FCLS. According to the results shown in Figure 32.28, NCLS demonstrated the best classification in terms of their unmixed PVEs among all LSMA methods. In particular, CSF and GM were classified correctly with little false positive classification as shown in Figure 32.28(b), (e), and (h) when NCLS, K-NCLS, and K-BEP-NCLS were used. On the other hand, the worst

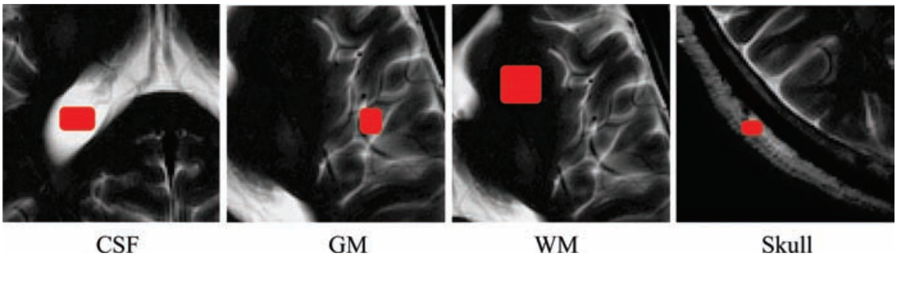


Figure 32.27 Tissues training sample regions for CSF, GM, WM, and skull for the real brain MR images.

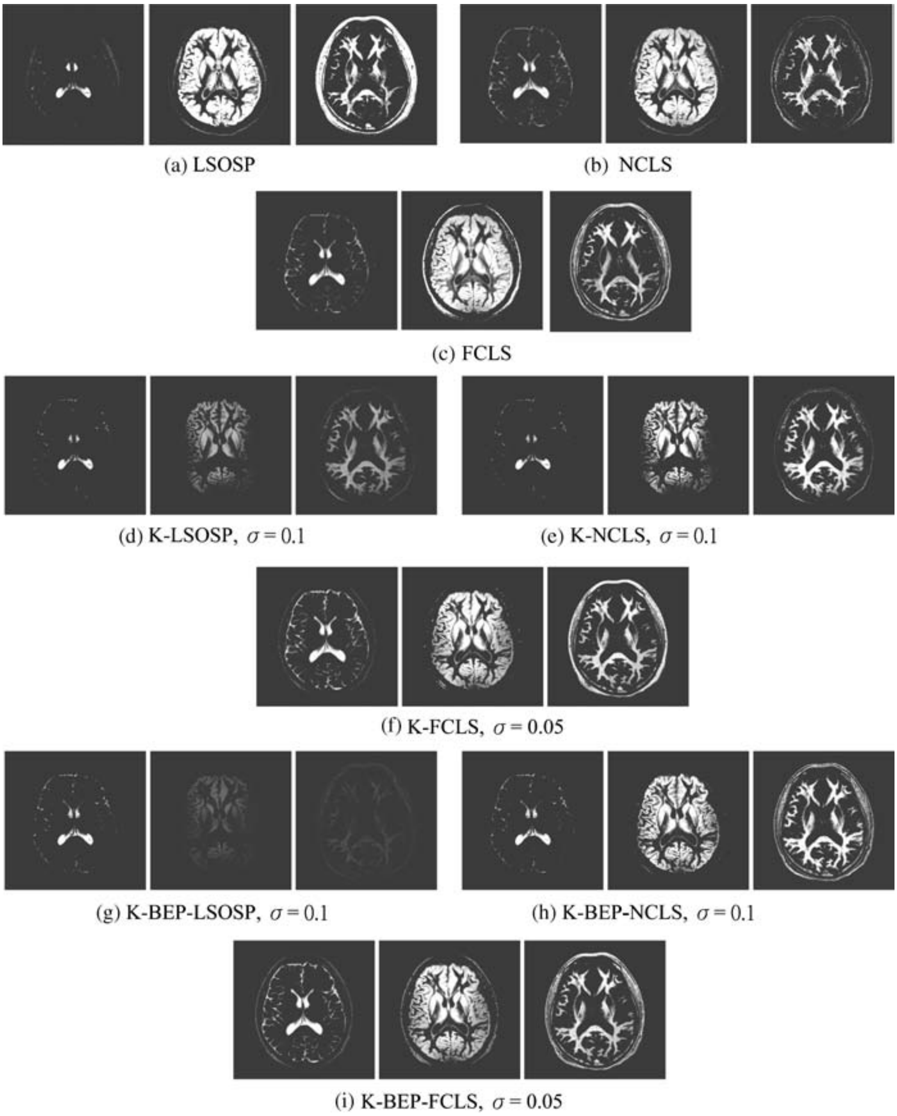

(a) LSOSP
(b) NCLS
(c) FCLS
(d) K-LSOSP, $\sigma = 0.1$
(e) K-NCLS, $\sigma = 0.1$
(f) K-FCLS, $\sigma = 0.05$
(g) K-BEP-LSOSP, $\sigma = 0.1$
(h) K-BEP-NCLS, $\sigma = 0.1$
(i) K-BEP-FCLS, $\sigma = 0.05$

Figure 32.28 Classification results of three main brain tissues of real brain MR image produced by (a) LSOSP, (b) NCLS, (c) FCLS, (d) K-LSOSP, (e) K-NCLS, (f) K-FCLS, (g) K-BEP-LSOSP, (h) K-BEP-NCLS, and (i) K-BEP-FCLS with five substances (CSF, GM, WM, skull, and background).

performance was LSOSP-based methods. As shown in Figures 32.28(a), (d), and (g), WM was classified with significant false positives from skull and muscle when LSOSP, K-LSOSP, and K-BEP-LSOSP were used. Generally speaking, K-LSMA improved the false positive classification rate. In particular, the false classification of the skull and muscle tissues into WM for LSOSP shown in Figure 32.28(a) was significantly reduced as shown in Figure 32.28(d). Similar to synthetic image experiments, K-BEP-LSMA had very little improvement on the false positive classification.

32.7 Conclusions

Spectral processing of multispectral MR images for partial volume estimation (PVE) is a new approach in MR image analysis where tissue substances are characterized by spectral information provided by MR image pulse sequences rather than spatial domain-based processing, which relies on spatial information provided by spatial correlation among sample pixels. As a result, spectral processing performs soft decisions on every pixel vector based on estimated abundance fractions of tissue substances to produce their partial volumes as opposed to spatial domain-based processing that makes hard decisions on every pixel vector based on class-labeling assignment. In order to materialize the concept of using spectral processing LSMA was first introduced by Wang et al. in a series of papers (Wang et al., 2000, 2001, 2003) in MR image classification where only unconstrained LSMA methods were investigated. This chapter extends their methods to two constrained LSMA methods and explores their utility in MR tissue characterization. Experimental results demonstrate that constrained LSMA generally performs better than unconstrained LSMA methods in terms of using unmixed results as PVE, specifically, tissue quantification, a task that cannot be accomplished by spatial domain-based processing techniques. However, it is the tissue quantification that can be used to calculate partial volumes of tissues, which are crucial in diagnosis of many diseases in progressive stages such as Alzheimer's diseases.

LSMA via nonlinear expansions has great potential in partial PVE since it is primarily designed to explore and take advantage of spectral properties provided by image sequences used for data collection. At present, an MR image slice can be processed as a 3D image cube formed by T1, T2, and PD. However, it is our belief that in future MR instruments may advance their technology to hyperspectral imagers like many others such as one in ophthalmology (Johnson et al., 2007) where more than three sequences will be used to characterize more soft tissue types, for example, tumors, anomalies. In this case, the information provided by these sequences is more crucial and vital than spatial information. Specifically, abnormal tissues usually neither have a large sample pool to provide reliable statistics nor have sufficient spatial information to allow spatial domain-based techniques to work effectively. Under such circumstances, MR image analysis must rely on the spectral information that can be obtained by stacking MR images as a multispectral or hyperspectral image by including extra band images generated by other image pulse sequences so that MR images can be processed as spectral band images with the total number greater than three. As a consequence, many currently available 3D techniques, such as 3D adaptive FCM segmentation (Ahmed et al., 2002; Liew and Yan, 2003; Siyal and Yu, 2005) and MRF (Zhang et al., 2001; Scherrer et al., 2009), may not be applicable to such high dimensions due to complexity of designing neighboring systems to capture spatial properties in a data space with dimensions higher than three. This is exactly where our proposed LSMA shows promise in future development of MRI and has advantages over the existing EM-MRF based or FCM-MRF-based techniques in PVE.

33

Conclusions

Writing a comprehensive book on hyperspectral data processing is very challenging since new applications and developments emerge any time as time goes along. Accordingly, keeping track of this area becomes realistically impossible or formidable if it can be done. During the course of preparing this book, the table of contents has been constantly revised to meet this high demand. Eventually, an attempt to cover all topics in this area is not only impossible but also beyond reality. To this end, this book is aimed to narrowing its scope and coverage to the research which has been carried out in the Remote Sensing Signal and Image Processing Laboratory (RSSIPL) at the University of Maryland, Baltimore County (UMBC), after my first book was published in 2003 (Chang, 2003a). Even in this case the materials to be included in this book also grow exponentially and exceed the original plan set for this book. To resolve this dilemma, several new interesting topics which are still ongoing and have potential in further development will not be discussed in this book; instead, they will be the main themes in my third book, entitled *Real Time Hyperspectral Image Processing* due to its publication by Springer-Verlag in 2013. Nevertheless, a brief review of these new topics will provide readers with a quick glimpse and also offer a peek at this cutting-edge research in this area. A second part of this final chapter is included to serve this purpose. The first part of this chapter summarizes what are presented in this book and provides further insights into topics which have potentials in future developments, but are not yet to explore at the time of writing this book.

33.1 Design Principles for Nonliteral Hyperspectral Imaging Techniques

Two principles, pigeon-hole principle and principle of orthogonality, have been backbones for designing and developing nonliteral hyperspectral imaging techniques presented in this book and Chang (2003a). While the pigeon-hole principle is a fundamental rule in discrete mathematics (Epp, 1995), the orthogonality principle is the most important criterion in designing mean squared error (MSE) estimation algorithms (Poor, 1994). If these two principles are intellectually interpreted to design algorithms, many new ideas and approaches will then follow naturally as described in the following.

33.1.1 Pigeon-Hole Principle

The pigeon-hole principle says that if p pigeons flying into L pigeon-holes (nests) with $L < p$, then there exists at least one pigeon-hole that must accommodate at least two or more pigeons. So, if we

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

interpret pigeons and pigeon-holes as material substances to be recognized and spectral bands, respectively, then a spectral band which is essentially a pigeon-hole can be used to accommodate a target substance, which is considered as a pigeon. In the following section, we described how this principle is used to interpret design rationales used for hyperspectral data processing.

33.1.1.1 Multispectral Imagery Versus Hyperspectral Imagery

First, the pigeon-hole principle can be used to resolve a long-standing controversial issue, which has been bothering researchers in remote sensing. From a linear spectral mixture analysis (LSMA) point of view, a linear mixing model can be used to describe a linear system of L algebraic equations to solve p unknowns. More specifically, each of these L linear equations is specified by one particular spectral band where the p unknowns are specified by abundance fractions of p image endmembers that are used to form the desired linear mixing model. The fact that LSMA utilizes an L-equation p-unknown system to unmix data sample vectors in terms of their unknown abundance fractions can be interpreted as the pigeon-hole principle using L pigeon-holes to accommodate p pigeons. So, when $L < p$ is true, there exists at least one pigeon-hole that must accommodate at least two or more pigeons which implies that at least one spectral band must be used to accommodate at least two or more image endmembers in which case it is a multispectral imaging problem. On the other hand, if $L \geq p$, there have more pigeon-holes than pigeons. In this case, we can use one spectral band to accommodate no more than one image endmembers, which is a hyperspectral imaging problem. Such interpretation also occurs in independent component analysis (ICA) in Hyvarinen et al. (2001), which can be interpreted exactly in the same way by the pigeon-hole principle. It makes use of a linear mixing model formed by L linear equations and p statistically independent random signal sources whose are unmixed by unknown weights via blind source separation. Accordingly, when $L < p$, ICA is called over-complete ICA as opposed to under-complete ICA if $L \geq p$. If the same language used by ICA is also applied to interpret LSMA, multispectral imaging and hyperspectral imaging can be further interpreted as over-complete LSMA and under-complete LSMA, respectively. Details of these discussions are provided in Chapter 31.

33.1.1.2 Virtual Dimensionality

Virtual dimensionality (VD) presented in Chapter 5 is another example that uses the pigeon-hole principle to define how many spectrally distinct signatures are present in hyperspectral data. However, as discussed in Chapter 5, VD is not a one-size-fit-all-applications criterion and cannot be fixed at a constant for all applications. In other words, VD must vary with targets of interests. To further address this, VD can be extended to a more general concept called target-specified VD, which will be studied in Chang (2013).

33.1.1.2.1 First-Order Spectral Statistics-Based Approaches

The key idea behind VD is the use of the pigeon-hole principle that was not explicitly explored in Harsanyi, Farrand, and Chang (HFC) method (1994). It assumes that each spectral band is considered as a pigeon-hole and can be used to accommodate only one target signal source as a pigeon. It further assumes that noise is a zero-mean random process, and target signal sources are deterministic and only contribute their energies to the first-order spectral statistics, that is, the sample spectral mean. With these assumptions, the sample spectral correlation matrix and sample spectral covariance matrix are used to calculate the differences between their corresponding eigenvalues. If the difference is greater than zero, a sample spectral correlation matrix-calculated eigenvalue will be greater than its corresponding sample spectral covariance matrix-calculated eigenvalue in which case it concludes that there must be a target signal source contributed to the sample mean of this particular spectral component. To materialize the above conjecture it formulates the difference between sample spectral correlation matrix-calculated eigenvalue and sample spectral covariance matrix-calculated eigenvalue for each of L spectral dimension as a binary composite hypothesis testing problem to determine if there is a target signal source present in the considered spectral dimension. By taking advantage of the Neyman–Pearson detection theory, a Neyman–Pearosn detector can be designed for such a test where a test fails if the alternative hypothesis is true to indicate that there is a target signal source present in the particular spectral dimension. Its effectiveness is completely determined by a given false alarm probability. The beauty of the HFC method is that its estimated value for VD varies with false probabilities, which can be adjusted according to different applications.

A general issue arising in the use of VD is that users always believe in that VD and the HFC methods are tied together as a pair and think that VD can only be found by the HFC method. As a result of such a misinterpretation of VD, some controversial issues occur (Bajorski, 2009). One major complaint about VD is that when the HFC method does not work as expected, it blames that VD is not appropriately defined. In the following section, we address this issue and clarify some controversial disputes that mislead users to misinterpreting VD.

In analogy with intrinsic dimensionality (ID) (Fukunaga, 1990), which defines the minimum number of parameters to represent high-dimensional data, VD is also a concept that defines an effective spectral dimensionality which is the number of *spectrally distinct* signatures required to characterize and represent a hyperspectral image. As a matter of fact, the effectiveness of VD is actually determined by its utility in various applications depending on different types of *spectral signatures* of major interest as well as the techniques used to find these signatures, both of which are the keys to its success. The HFC method is developed as *one* technique to determine VD in a similar manner that the principal component analysis (PCA) is used as *one* technique to determine ID. So, when the HFC method does not work effectively, it should not be concluded that VD is not correctly defined. It simply implies that the HFC method is not applicable to the considered applications. Similarly, the same conclusion cannot also be drawn for ID when PCA fails in finding a correct number of parameters. For example, the number of endmembers, n_{E}, for an endmember extraction algorithm to generate is generally not the same as the number of image endmembers required for LSMA to perform spectral unmixing n_{LSMA}. Also neither of n_{E} and n_{LSMA} should be the same as the number of anomalies n_{A} even though all these signal sources are considered as spectrally distinct signatures. This indicates that the value of VD should vary with applications which ultimately determine the techniques to be used to estimate VD. Furthermore, if we assume that each spectral band can be used to accommodate one particular material substance via the pigeon-hole principle, then the number of bands to selected, n_{BS}, and the number of dimensions required to be retained after dimensionality reduction (DR), n_{DR}, can also be determined by VD as well, but their values should also be different since DR and band selection (BS) are completely different processes in terms of how to preserve the desired information. Accordingly, the HFC method should not be considered as a one-size-fit-all-applications technique. This issue is addressed with great details in Chapter 5 where two types of approaches based on criteria driven by data characterization and data representation are developed to determine VD, and the HFC method is considered as a data characterization-driven technique.

The primary reason that the HFC method may not be effective on some occasions is due to the fact that the HFC method does not differentiate different types of spectrally distinct signatures because it is an application-independent technique. For example, it does not distinguish an endmember from an anomaly where the former is a pure signature while the latter may be a subpixel or mixed pixel signature. To resolve this issue, the HFC method must take into account the targets of interest for which it considers as spectrally distinct signatures. To do so, we extend the HFC method by replacing the eigenvalues considered in the binary hypothesis testing problem with target signatures generated by a specific algorithm which is, in turn, determined by a particular application. Since an algorithm to be used to produce targets of interest can be designed based on spectral statistics of different orders, the HFC method can be further extended from a data characterization-driven technique using the sample spectral mean to a data characterization-driven technique using statistics of orders higher than the first order. Two types of algorithms can be used for this purpose. First is the automatic target generation process (ATGP), which is based on second-order statistics and developed by Ren and Chang (2003), and the second is high-order statistics (HOS) algorithm developed by Ren et al. (2006). When the HFC method uses ATGP-generated targets and HOS-generated targets as spectrally distinct signatures for VD estimation, it will be referred to as second-order HFC method and HOS HFC method, respectively, while the original HFC method will be considered as the first-order HFC method.

33.1.1.2.2 Second-Order Spectral Statistics-Based HFC Methods

To develop second-order statistics HFC method, we revisit the idea behind the HFC method and investigate its difference from PCA which is the well-known second-order statistics-based transform. Despite the fact that both use sample spectral correlation and sample spectral covariance matrices to derive criteria, they also differ from each other greatly in many aspects. When PCA is used for VD estimation, two ways are generally used (also see Section 5.3.1 in Chapter 5). One is to plot the distribution of eigenvalues $\{\lambda_j\}_{j=1}^{L}$ in descending order and find a sudden drop which indicates where VD is. The problem with this approach is how much gap between two consecutive eigenvalues is considered as a sudden drop. As an alternative, it calculates the ratio of the accumulative sums of the eigenvalues in descending order to the total sum of eigenvalues, $\mathrm{R}_\lambda(p) = \left(\sum_{j=1}^{p}\lambda_j\right)/\left(\sum_{j=1}^{L}\lambda_j\right)$ and determines VD by setting a prescribed percentage $a\%$., that is,

$$\mathrm{VD}^{\mathrm{PCA}}(a\%) = \arg\left\{\min_p[\mathrm{R}_\lambda(p) \geq a/100]\right\} \tag{33.1}$$

In this case, selecting an appropriate $a\%$ in (33.1) as a cutting-threshold is also challenging. In a completely different approach, the HFC method implements a binary hypothesis test for each spectral dimension to see if a Neyman–Pearson detector fails for this particular spectral dimension. In this case, the HFC method tests if each of spectral dimensions can be used to accommodate one signal source as opposed to PCA using eigenvalues to determine the total number of signal sources instead of a signal source in an individual spectral dimension. How effective such a detector is then determined by the false alarm probability, $\mathrm{P_F}$. Another difference is that PCA uses the second-order statistics such as variance to determine VD while the HFC method uses the first-order statistics which is the sample spectral mean vector to determine VD. In hyperspectral data exploitation, the signatures of interest are generally *spectrally* distinct and have several unique features. For example, they usually do not have a large pool of sample vectors present in the data. So, they are insignificant targets in the sense of signal energy characterized by eigenvalues. As a result, the variance resulting from these sample vectors described as second-order of spectral statistics is relatively small and is nearly negligible. So using second-order spectral statistics such as variance may not be effective. On the other hand, these spectrally distinct signatures are generally contributed as a first-order of spectral statistics which are more significant and crucial than the variance. In this case, the HFC method can be very effective in determining whether or not a spectral dimension contains a signature. This also explains why the HFC method works effectively when the sample pools of target signatures are small in which case PCA fails because the variances of such target signatures are very small compared to that of background signatures. Finally, both PCA and the

HFC method do not explicitly provide a means of finding signatures of interest. Nevertheless, we can always use these eigenvectors that corresponds to selected eigenvalues as desired basic elements, that is, feature vectors to signatures of interest. In other words, PCA selects the first largest $\mathrm{VD}^{\mathrm{PCA}}(a\%)=p$ eigenvalues to identify the signatures of interest. Similarly, the HFC method finds the $\mathrm{VD}_{\mathrm{HFC}}^{\mathrm{NP}}(\mathrm{P_F})=p$ spectral dimensions determined by the Nyeman–Pearson detector which are not necessarily the same as the first p eigenvectors specified by PCA. In this case, the signatures of interest found by the HFC method as feature vectors will not be the same eigenvectors found by PCA. Consequently, a direct use of PCA-generated eigenvectors to derive the second-order HFC method for VD estimation seems not to be applicable. Recently, several efforts on developing second-order spectral statistics-based approaches have been reported. One is referred to maximum orthogonal complement algorithm (MOCA) developed by Kuybeda et al. (2007). Two others are maximum orthogonal subspace projection (MOSP) developed by Xiong and Chang (2010) and automatic target generation process/Mahalanobis distance (ATGP/MD) developed by Jiao (2010). Interestingly, all these approaches are actually derived from ATGP with its details discussed in Sections 8.5.1 and 30.2.

The idea of ATGP is to first find a data sample vector, $\mathbf{t}_0^{\mathrm{ATGP}}$, with the maximal vector length that will be considered as a spectral signal source. To make sure that a spectral dimension which has been used to accommodate the data sample vector $\mathbf{t}_0^{\mathrm{ATGP}}$ will not be used again to accommodate another spectral signal source, an orthogonal subspace projection (OSP) is performed to find a data space $\langle\mathbf{t}_0^{\mathrm{ATGP}}\rangle^{\perp}$ that is orthogonal to the space $\langle\mathbf{t}_0^{\mathrm{ATGP}}\rangle$ linearly spanned by $\mathbf{t}_0^{\mathrm{ATGP}}$. ATGP then finds a second data sample vector with the maximal vector length in the space $\langle\mathbf{t}_0^{\mathrm{ATGP}}\rangle^{\perp}$, denoted by $\mathbf{t}_1^{\mathrm{ATGP}}$ as a second spectral signal source. In this case, it requires another spectral dimension as a second dimension to accommodate this second spectral signal source $\mathbf{t}_1^{\mathrm{ATGP}}$. To find a third spectral signal source, ATGP finds a data sample vector with the maximal vector length, $\mathbf{t}_2^{\mathrm{ATGP}}$ as a third spectral signal source in the space $\langle\mathbf{t}_0^{\mathrm{ATGP}},\mathbf{t}_1^{\mathrm{ATGP}}\rangle^{\perp}$ that is orthogonal to the space linearly spanned by $\mathbf{t}_0^{\mathrm{ATGP}}$ and $\mathbf{t}_1^{\mathrm{ATGP}}$. The same process is repeated over and over again to find $\mathbf{t}_3^{\mathrm{ATGP}}$, $\mathbf{t}_4^{\mathrm{ATGP}}$, etc. The key issue is when ATGP should be terminated. In the original ATGP, the algorithm is terminated by a prescribed error threshold ε. However, in many applications such error threshold selection is somewhat subjective and difficult to determine. An alternative to selection of error threshold is to determine the number of spectral signal sources required for ATGP to extract. This leads to an interesting conjecture: "Can a stopping rule of ATGP be used to estimate VD?" The three abovementioned approaches, MOCA, MOSP, and ATGP/MD, are particularly designed to come up different appropriate stopping rules to terminate ATGP, which turns out to the same issue in finding an effective criterion to estimate VD. In other words, once ATGP is terminated at the mth target, $\mathbf{t}_m^{\mathrm{ATGP}}$, the value of VD is then estimated as m, that is, $n_{\mathrm{VD}}=m$.

More specifically, for each $1\leq l\leq L$ we define

$$\mathbf{t}_l^{\mathrm{ATGP}}=\arg\{\max_{\mathbf{r}}||\mathbf{P}_{\mathbf{S}_l}^{\perp}\mathbf{r}||\} \tag{33.2}$$

$$\eta_l=||\mathbf{t}_l^{*}||^2 \tag{33.3}$$

where $\mathbf{S}_l=\{\mathbf{t}_0^{\mathrm{ATGP}},\mathbf{t}_1^{\mathrm{ATGP}},\ldots,\mathbf{t}_{l-1}^{\mathrm{ATGP}}\}$. It should be noted that $\{\mathbf{S}_l\}$ is monotonically increasing at l in the sense that $\mathbf{S}_1\subset\mathbf{S}_2\subset\cdots\subset\mathbf{S}_l$ and $\{\eta_l\}$ is monotonically decreasing at l in the sense that $\eta_1\geq\eta_2\geq\cdots\geq\eta_l$. Now the stopping rule of ATGP can be cast as a binary composite hypothesis testing problem given by

$$\begin{array}{ll} H_0\;:\;\eta_l\approx p(\eta_l|H_0)=p_0(\eta_l) & \\ \text{versus} & \text{for } l=1,2,\ldots,L \\ H_1\;:\;\eta_l\approx p(\eta_l|H_1)=p_1(\eta_l) & \end{array} \tag{33.4}$$

where the alternative hypothesis H_1 and the null hypothesis H_0 represent target signal subspace and background subspace, respectively. To make (33.4) work, we need to find probability distributions under both hypotheses. Because the orthogonal complement subspace projections of data sample vectors $\mathbf{P}_{\mathbf{S}_l}^{\perp}\mathbf{r}_i$ under H_0 are supposed to be noise sample vectors, it is reasonable for MOCA to assume that the vector $\mathbf{P}_{\mathbf{S}_l}^{\perp}\mathbf{r}_i$ under H_0 behaves as independent identically Gaussian random variables. Moreover, η_l is the maximal residuals of orthogonal projection (OP) obtained in $\langle \mathbf{S}_l \rangle^{\perp}$ under H_0. By virtue of extreme value theory (Leadbetter, 1987), η_l can be modeled as a Gumbel distribution, that is, $F_{v_l}(\eta_l)$ is the cumulative distribution function (cdf) of v_l given by

$$F_{v_l}(x) \approx \exp\left\{ -e^{-(2\log N)^{1/2}\left[\frac{x-\sigma^2(L-l)}{\sigma^2\sqrt{2(L-l)}}-(2\log N)^{1/2}+\frac{1}{2}(2\log N)^{-1/2}(\log\log N+\log 4\pi)\right]} \right\} \tag{33.5}$$

By virtue of (33.4) and (33.5), we can derive two detectors, one is ATGP-Bayes detector, $\delta_{\text{ATGP}}^{\text{Bayes}}$, which assumes the uniform cost and equal likely prior probabilities in (2.1) and ATGP-Neyman Pearson detector, $\delta_{\text{ATGP}}^{\text{NP}}$ which assumes no prior information as follows:

$$\delta_{\text{ATGP}}^{\text{Bayes}}(\eta_l) = \begin{cases} 1; & \Lambda(\eta_l) \geq \tau \\ 0; & \Lambda(\eta_l) < \tau \end{cases} \tag{33.6}$$

and

$$\delta_{\text{ATGP}}^{\text{NP}}(\eta_l) = \begin{cases} 1; & > \tau \\ \kappa; & \Lambda(\eta_l) = \tau \\ 0; & < \tau \end{cases} \tag{33.7}$$

where κ is the probability that H_1 is true when $\Lambda(\eta_l)$ is equal to the threshold τ and the false alarm probabilities of $\delta_{\text{ATGP}}^{\text{Bayes}}$ for (33.6) and $\delta_{\text{ATGP}}^{\text{NP}}$ for (33.7) are given by

$$\mathrm{P_F}\left(\delta_{\text{ATGP}}^{\text{Bayes}}\right) = \int_{\Lambda(\eta_l)\geq\tau} p_0(\eta_l)d\eta_l \text{ and } \mathrm{P_F}\left(\delta_{\text{ATGP}}^{\text{NP}}\right) = \int_{\Lambda(\eta_l)\geq\tau} p_0(\eta_l)d\eta_l \tag{33.8}$$

where $p_0(\eta_l) = p(\eta_l|H_0) = p_{v_l}(\eta_l)$ resulting from the use of the linear mixture model with i.i.d. noise. For a given false alarm probability $\mathrm{P_F}$, $\mathrm{VD}_{\text{ATGP}}^{\text{Bayes}}(\mathrm{P_F})$ and $\mathrm{VD}_{\text{ATGP}}^{\text{NP}}(\mathrm{P_F})$ can be defined as ATGP-HFC methods, which are based on the first two orders of spectral statistics, as follows:

$$\mathrm{VD}_{\text{ATGP}}^{\text{Bayes}}(\mathrm{P_F}) = \arg\left\{\min_{1\leq l\leq L}\left\{\delta_{\text{ATGP}}^{\text{Bayes}}(\mathrm{P_F}\,) \ = 0\right\}\right\} \tag{33.9}$$

$$\mathrm{VD}_{\text{ATGP}}^{\text{NP}}(\mathrm{P_F}) = \arg\left\{\min_{1\leq l\leq L}\left[\delta_{\text{ATGP}}^{\text{NP}}(\mathrm{P_F})\right]\right\} \tag{33.10}$$

where the bracket $\left[\delta_{\text{ATGP}}^{\text{NP}}(\mathrm{P_F})\right]$ in (33.10) is the largest integer smaller or equal to $\delta_{\text{ATGP}}^{\text{NP}}(\mathrm{P_F})$. For details we refer readers to the reference, Chang et al. (2011c).

33.1.1.2.3 High-Order Spectral Statistics-Based Approaches

Many hyperspectral targets of interest are generally insignificant and their occurrences usually have low probabilities with small populations; their contributions to second-order spectral statistics are usually very limited. Consequently, using the first two order spectral statistics-based techniques such as the first-order spectral statistics, sample spectral mean in the HFC method and the first two order spectral statistics, i.e., ATGP in the ATGP-HFC method to determine VD may sometimes ineffective. Therefore, this section further extends the HFC method to high-order spectral statistics (HOS) HFC method which can be used to estimate VD for applications in hyperspectral target analysis where target signal sources of interest are spectrally characterized by HOS (Chang and Xiong, 2010). This can be accomplished by replacing ATGP-generated targets used in (33.4) used for the ATGP-HFC method with targets generated by HOS-based algorithms. Specifically, let $\left\{\mathbf{t}_l^{\mathrm{HOS}}\right\}_{l=1}^{L}$ denote the HOS-generated targets which can be divided into two groups of signal sources, one representing signal sources $\mathbf{s}_l$ including target signals and background signatures, while others are noise $\mathbf{n}_l$. Then (33.4) becomes a binary hypothesis testing problem for HOS-statistics signal sources, that is, noise $\mathbf{n}_l$ under H_0 versus signal $\mathbf{s}_l$ under H_1, $1 \leq l \leq L$. One point should be noted here is that the signals $\mathbf{s}_l$ have different meanings when they are generated by different HOS methods. For example, when PCA is used to generate the feature vectors, which optimize the second-order spectral statistics, the background or target signals with large area are more effective to be detected. However, HOS methods such as skewness, kurtosis, fifth moment, and ICA are more effective to find the target signals with small populations, which are more significant to high-order spectral statistics.

The binary hypothesis test (33.4) can be re-expressed by

$$\begin{array}{ll} H_0 \; : \; z_l = 0 & \\ \text{versus} & \text{for } l = 1, 2, \ldots, L \\ H_1 \; : \; z_l > 0 & \end{array} \tag{33.11}$$

where z_l is the maximum of vectors residuals given by $z_l = \max_{1 \leq i \leq N} \left\|\mathbf{P}_{\mathbf{t}_l^{\mathrm{HOS}}}\mathbf{r}_i\right\|^2$ and $\mathbf{P}_{\mathbf{t}_l^{\mathrm{HOS}}}\mathbf{r}_i$ is the OP of each pixel vectors $\mathbf{r}_i$, $1 \leq i \leq N$, to the subspace spanned by each signal source vector $\mathbf{t}_l^{\mathrm{HOS}} 1 \leq l \leq L$, that is $\mathbf{s}_l$ and $\mathbf{n}_l$, where $\mathbf{P}_{\mathbf{t}_l^{\mathrm{HOS}}} = \mathbf{t}_l^{\mathrm{HOS}}\left(\left(\mathbf{t}_l^{\mathrm{HOS}}\right)^T \mathbf{t}_l^{\mathrm{HOS}}\right)^{-1}\left(\mathbf{t}_l^{\mathrm{HOS}}\right)^T$. As we know, the maximum of residuals of projection vectors on the signal subspace is a high value contributed by the signals in this direction under H_1, whereas the noise under H_0 is a low value which is governed by the maximal norm noise residual. Under the null hypothesis H_0 with the white Gaussian noise assumption, the cdf of z_l is Gumbel distribution (Leadbetter, 1987) given by

$$F_0(z) \approx \exp\left\{-e^{-(2\log N)^{1/2}\left[\frac{z-\sigma^2(L-l)}{\sigma^2\sqrt{2(L-l)}}-(2\log N)^{1/2}+\frac{1}{2}(2\log N)^{-1/2}(\log\log N+\log 4\pi)\right]}\right\} \tag{33.12}$$

In addition, under the alternative hypothesis H_1 with the uniformly distribution assumption as in Kuybeda et al. (2007), *a posteriori* probability distribution is given by

$$p(H_1 \mid z_l) = \frac{F_0(z_l)}{z_l p_0(z_l) + F_0(z_l)} \tag{33.13}$$

$$p(H_0 \mid z_l) = \frac{z_l p_0(z_l)}{z_l p_0(z_l) + F_0(z_l)} \tag{33.14}$$

where $p_0(z_l)$ is the pdf z_l under H_0.

Like ATGP-HFC methods, two detectors are used to determine VD by our HOS-HFC methods as follows:

$$\delta_{\mathrm{HOS}}^{\mathrm{Bayes}}(z_l) = \begin{cases} 1; & \text{if } \Lambda(z_l) > 1 \\ 0; & \text{if } \Lambda(z_l) \leq 1 \end{cases} \tag{33.15}$$

$$\delta_{\mathrm{HOS}}^{\mathrm{NP}}(\eta_l) = \begin{cases} 1; & > \tau \\ \kappa; & \Lambda(\eta_l) = \tau \\ 0; & < \tau \end{cases} \tag{33.16}$$

where $\Lambda(z_l) = p(H_1 \mid z_l)/p(H_0 \mid z_l)$. Using (33.15) and (33.16), VD can be determined by calculating

$$\mathrm{VD}_{\mathrm{HOS}}^{\mathrm{Bayes}} = \sum\nolimits_{l=1}^{L} \delta_{\mathrm{HOS}}^{\mathrm{Bayes}}(z_l) \tag{33.17}$$

and

$$\mathrm{VD}_{\mathrm{HOS}}^{\mathrm{NP}} = \sum\nolimits_{l=1}^{L} \left\lfloor \delta_{\mathrm{HOS}}^{\mathrm{NP}}(z_l) \right\rfloor \tag{33.18}$$

where the false alarm rates are given by

$$\mathrm{P_F}(\delta^{\mathrm{NP}}) = \int_{\Lambda(z_l) \geq \tau} p_0(z_l)\mathrm{d}z_l \text{ and } \mathrm{P_F}(\delta^{\mathrm{NP}}) = \int_{\Lambda(z_l) \geq \tau} p_0(z_l)\mathrm{d}z_l \tag{33.19}$$

The above HOS-HFC methods stem from the HFC method to determine VD using the target signals generated by various HOS criteria, such as variance used by PCA, skewness, kurtosis, fifth moment, and mutual information used by ICA. It takes advantage of assumptions of Gaussian noise distribution under H_0 and uniformly distribution under H_1.

33.1.2 Principle of Orthogonality

The pigeon-hole principle cannot work alone effectively by itself. When $L < p$ in which case there are at least two or more material substances which must be accommodated by a single spectral band. To mitigate this problem, spatial domain-based techniques using intersample spatial correlation are generally implemented. This may explain why most multispectral imaging techniques are developed based on spatial domain correlation. On the other hand, if $L \geq p$, it implies that there are more spectral bands than material substances to be recognized. As a result, a single spectral band can be used to accommodate no more than one material substance. Under such a circumstance, two scenarios can occur. One is absence of material subtances in a single spectral band in which case only noise assumed to be in the band. The other is the presence of one material substance in a single spectral band. In the latter case, there should be a means to be used to guarantee that once a single spectral band is used to accommodate one material substance, it cannot be used again to accommodate other material substances. The principle of orthogonality provides a solution to resolve this issue and it turns out to be a very effective means of accomplishing this goal.

For an illustrative purpose, we use the orthogonal subspace projection (OSP) detector developed in Chapter 12 as an example. It is primarily designed based on the principle of orthogonality and specified by $\delta^{\mathrm{OSPD}}(\mathbf{r}) = M_{\mathbf{d}} P_{\mathbf{U}}^{\perp}\mathbf{r} = \kappa \mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r}$ in (12.9). It involves two operators, a matched filter $\mathrm{M}_{\mathbf{d}}$ and an orthogonal subspace projector, $P_{\mathbf{U}}^{\perp}$ to perform spectral unmixing for LSMA. First, it takes advantage of a signal detection model (12.1) formulated by a binary hypothesis testing problem (2.1). It then separates a desired signature **d** from a set of undesired signatures, **U** to re-expressed (12.1) as (12.2) so that all the undesired signatures in **U** can be annihilated by $P_{\mathbf{U}}^{\perp}$ prior to detection of the desired signature **d**. This is then followed by a matched filer $\mathrm{M}_{\mathbf{d}}$ to extract the signature **d**. If the number of spectral bands, L, is greater than the number of signatures, p, used for LSMA, the pigeon-hole principle allows each of p signatures to be accommodated in one spectral band and also no more than one signature accommodated in a single spectral band. Furthermore, since the desired signature **d** is separated from the undesired signatures in **U** in different spectral bands, the principle of orthogonality allows the operator $P_{\mathbf{U}}^{\perp}$ to eliminate all the undesired signatures in **U**, while making sure that it does not have effect on the desired signature **d**. In the mean time, it also allows the matched filter $\mathrm{M}_{\mathbf{d}}$ to extract only the desired signature **d** since signatures other than **d** had been eliminated by $P_{\mathbf{U}}^{\perp}$ via the principle of orthogonality.

Another good illustrative example is ATGP discussed in Section 33.1.1.2.2 which has a wide range of applications such as unsupervised target detection, endmember extraction, anomaly detection, and VD estimation. It performs a succession of orthogonal subspace projections to generate a set of targets so that these targets statistically un-correlated in the least squares sense. However, this can only occur when the total number of spectral bands is greater than the number of targets to be generated. A joint use of principle of orthogonality and the pigeon-hole principle guarantee to make ATGP work effectively.

33.2 Endemember Extraction

Endmember extraction has received considerable interests in recent years. This is mainly because with very high spectral resolution provided by hundreds of contiguous spectral bands hyperspectral sensors are capable of extracting many unknown and subtle signal sources which cannot be generally resolved by multispectral sensors. One of such signal sources is endmembers which provides crucial and vital information in data exploitation. Particularly, endmembers can be used to specify spectral classes present in the data. According to the definition in Schowengerdt (1997), an endmember is an idealized, pure signature for a class. So, technically speaking, an endmember is a signature generally available in a spectral library or database, and it is not necessarily a data sample vector present in the data. With this clarification, a pure data sample vector is specified by an endmember, but may not be the other around. Unfortunately, in real applications pure data sample vectors probably never exist because they may be contaminated by many unexpected effects resulting from noise, clutters, unknown interferers, etc. Accordingly, when an endmember extraction algorithm is applied to real data, it intends to find those data sample vectors which are most likely pure and represent endmembers. In unsupervised LSMA (ULSMA) discussed in Chapter 17, such data sample vectors are referred to as virtual signatures (VSs) and can be found by an unsupervised algorithm to be used for spectral unmixing.

Due to the nature of purity in endmembers, convexity geometry has been a key notion to design algorithms to find endmembers. An earliest attempt was reported by Boardman (1994) who used OP to develop one of best known endmember extraction algorithms, pixel purity index (PPI). Since there is no prior knowledge that can be used to find endmembers, its idea uses a set of randomly generated unit vectors, to be called skewers, to point to as many different projection directions as possible. All data sample vector are then orthogonally projected on each of these skewers where the maximal or minimal projection on each skewer can be calculated. For each data sample vector, a PPI count can be produced by counting many skewers on which this particular data sample vector has either maximal or minimal projection. By adaptively thresholding PPI counts obtained for all data sample vectors, a set of candidates for endmembers can be generated and further used to extract a desired set of endmembers by human intervention. To alleviate the random nature caused by using skewers resulting from PPI and human intervention, Chang and Plaza (2006) developed an initialization-specified PPI where an endmember initialization algorithm, ATGP was used to generate a specific set of initial skewers and Chang et al. (2010) took a complete opposite direction by considering PPI as a random algorithm where the result produced by PPI using one set of random initial skewers is considered as a realization of such a random algorithm.

Another attempt deviated from using of OP was made by Ifarraguerri and Chang (1999) who appealed for the concept of positive convex cone to satisfy abundance non-negativity constraint (ANC) imposed on endmembers and then searched for a set of data sample vectors with maximal convex cone volume within the data space. The vertices of the found convex cone are the desired endmembers. Since then several similar ideas using the concept of non-negative matrix factorization (NMF) developed in Lee and Seung (1999) were further explored for endmember extraction, for example, the work by Pauca et al. (2006).

A third attempt was reported by Craig (1994) who proposed a minimal volume transform (MVT)-based approach to satisfy both ANC and abundance sum-to-one (ASC) imposed on endmembers to find a set of data sample vectors that forms a simplex with minimal volume which includes all the data sample vectors. The MVT criterion has played a key role in the development of NMF-based minimal volume constrained non-negative matrix factorization (MVC-NMF) developed by Miao and Qi (2007), and linear programming-based minimal volume enclosing simplex (MVES) was developed by Chan et al. (2009).

A fourth attempt was further made by Winter who developed an N-finder algorithm (N-FINDR) (Winter, 1999a, 1999b, 2004). It is very similar to MVT. But instead of minimizing simplex volume it maximizes the volumes of simplexes that are embedded in data sample space. In other words, Craig deflated simplexes that contain all data sample vectors until it reaches a simplex with minimal volume, whereas Winter took an opposite approach by inflating simplexes embedded in data space until it reaches a simplex with maximal volume. Up to now the concept of the Winter approach is probably the most widely used criterion in the literature to design an endmember extraction algorithm due to its close tie with spectral unmixing. Therefore, it is interesting to provide a little history for an evolution of the N-FINDR development over the past years.

Since the Winter N-FINDR was first proposed (Winter, 1999a, 1999b, 2004; Winter and Winter, 2000), many efforts have been devoted to improving this algorithm in the sense of computational efficiency. According to Winter's criterion finding a simplex of maximal volume should exhaust all possible simplexes. While such an exhaustive search is nearly impossible in practice, two approaches have been investigated in the past. One is to grow simplexes one vertex after another such as the simplex growing algorithm (SGA) developed Chang et al. (2006). The other is modifications of the Winter algorithm in terms of implementing N-FINDR sequentially similar to the one developed in Winter (2004) which implements two iterative loops with the outer loop iterated for data sample vectors and the inner loop iterated for endmember replacement. Many improved N-FINDR algorithms reported in the literature belong to the latter category which uses such a sequential Winter N-FINDR as a base to develop different variants (Plaza and Chang, 2005, 2006; Wu et al., 2008; Chowdhury and Alam, 2007; Zortea and Plaza, 2009; Dowler and Andrews, 2011; Du et al., 2008a, 2008b; Wang et al., 2009). One major approach is to modify the algorithmic structure used in the sequential Winter N-FINDR. An earliest attempt was made in Plaza and Chang (2005) and Plaza and Chang (2006) where an algorithm was specifically laid out to fill in missing details in Winter (1999a, 1999b) to implement a different version of sequential N-FINDR. It was then followed by various versions developed by Wu et al. (2008), Chowdhury and Alam (2007), Zortea and Plaza (2009), and Dowler and Andrews (2011). With an interesting twist of swapping two loops in the Winter N-FINDR (2004), that is, making the inner loop an outer loop and the outer loop an inner loop, Wu et al. (2008) derived another version of the sequential Winter N-FINDR algorithm, called SuCcessive N-FINDR (SC N-FINDR). Unfortunately, the work in Wu et al. (2008) was not referenced in Du et al. (2008a, 2008b) and Wang et al. (2009) where their developed algorithms either identical or very close to the SC N-FINDR.

While many N-FINDR developers have placed their focus on sequential implementation of N-FINDR (Winter, 1999a, 1999b, 2004; Winter and Winter, 2000; Chowdhury and Alam, 2007; Zortea and Plaza, 2009; Dowler and Andrews, 2011; Du et al., 2008a, 2008b; Wang et al., 2009), the issue of using initial conditions in N-FINDR has been overlooked. This is in fact a serious problem with reproducibility of final selection of endmembers since N-FINDR produces different final results if different sets of random initial conditions are used. Three ways have been suggested to resolve this issue. One is to use an endmember initialization algorithm (EIA) to generate an appropriate set of initial endmembers such as automatic target generation (ATGP) in Plaza and Chang (2005). Another is real-time N-FINDR processing developed in Wu et al. (2010) which takes the first p input data sample vectors as initial endmembers. A third one is random N-FINDR proposed in Chang et al. (2009), Wu (2009), and Chang et al. (2011), which considered N-FINDR as random algorithm to repeatedly implement N-FINDR with different sets of random initial endmembers until a stopping rule is met.

According to the algorithmic structure implemented in IN-FINDR developed in Section 7.2.3.2, the inner loop is indeed a two-loop sequential process with one loop iterating N data sample vectors for each a given fixed endmember position and the other loop iterating p endmember positions, which is the number of endmembers, while the outer loop simply iterates another new set of endmembers found by the inner loop in a previous iteration. So, in order to make such an implementation more clearer and attractive Xiong et al. (2011) revisited IN-FINDR and separated the inner loop implemented in the IN-FINDR from its outer loop to have it implemented as a stand-alone algorithm, referred to as SeQuential N-FINDR (SQ N-FINDR). By virtue of this new defined SQ N-FINDR, IN-FINDR can be re-implemented in a broader sense as a three-loop (outermost, outer, and inner loops) sequential process where the outermost loop is used to iterate a new initial set of p endmembers generated by the outer loop; the outer and inner loops are designed to iterate data sample vectors and endmember positions separately. When the inner loop iterates p endmember positions and the outer loop iterates N data sample vectors, it is called the SQ N-FINDR. When these two loops are reversed, the resulting two-loop process is called SC N-FINDR. As a result, in analogy with SC N-FINDR, the performance of SQ N-FINDR is also heavily determined by its used initial conditions. In this case, in order to mitigate this random issue, they can be jointly implemented with the outermost loop carried out in IN-FINDR, respectively. The resulting IN-FINDR algorithms are referred to as iterative SQ N-FINDR (ISQ N-FNDR) and iterative SC N-FINDR (ISC N-FINDR). With this interpretation/IN-FINDR developed in Section 7.2.3.2 is in fact the ISQ N-FINDR which is identical to the sequential N-FINDR in Wu et al. (2008) as well as the IN-FINDR referred in Chang et al. (2009), Wu (2009), and Xiong et al. (2011), all of which include a third loop to implement SQ N-FINDR or SC N-FINDR repeatedly by taking final set of endmembers generated in the previous run as a new set of initial conditions for next run. To the author's best knowledge, many algorithms designed to implement N-FINDR in the literature can be considered to be either identical or equivalent to SQ N-FINDR, SC N-FINDR, or IN-FINDR one way or another. Interestingly, it is believed that IN-FINDR remains the only one which has never been explored in the literature except those in Wu et al. (2008), Chang et al. (2009), Wu (2009), and Xiong et al. (2011) and is yet to be explored in the future. More details can be found in Chang (2013).

Several dilemmas resulting from implementing N-FINDR described above have led to the development of SGA which finds p endmembers by growing simplexes one vertex at a time where each new endmember is specified by a newly added vertex that yields the maximal volume of simplexes being considered (Chang et al., 2006). As we recall in Section 7.2.3.3, where a multiple-replacement version of IN-FINDR, referred to as s-IN-FINDR, is developed the number of endmembers needed to be replaced in each iteration is set to s. When $s = 1$, it is reduced to the single-replacement IN-FINDR. On the other hand, when $\mathrm{s} = p$, s-IN-FINDR becomes the original version of N-FINDR. So, this idea can be also carried over to generalize SGA to s-SGA where the number of initial endmembers needed to start with SGA is set to s. In this case, implementing s-SGA requires two-stage processes. The first-stage process is to find a set of s initial endmembers that maximizes volumes of s-simplexes to initialize SGA, and the second-stage process is to grow and find the maximal simplex volume starting from an $(s+1)$-simplex to a p-simplex. Using this generalization, when $s = 1$ and 2, s-SGA is reduced to 1-SGA and 2-SGA as described in Section 8.3 and Figure 8.1. As a consequence, the computational complexity of implementing s-SGA is the sum of computational costs of two-stage processes, that is, computation complexity of finding an optimal set of s initial endmembers plus computational complexity of finding growing simplexes with maximal volumes starting from the number of vertices, $s+1$ to p. As s is increased, the heavy computational load begins to shift from finding growing simplexes with maximal volumes to finding an optimal set of s initial endmembers. When it reaches p, that is, $s = p$, no process of growing simplexes is needed and the entire computational cost is completely determined by finding an optimal set of p initial endmembers. In this case, p-SGA becomes the original version of N-FINDR, which can be considered as a special case of s-SGA with s set to p.

The three design rationales, OP, convex cone, and simplex described above are main trends for designing and developing endmember extraction algorithms. Interestingly, these three are actually derived from three versions of how to implement LSMA discussed in Chang (2003a) which are abundance-unconstrained LSMA, least squares OSP (LSOSP), ANC-constrained constrained least squares (NCLS) (a partially abundance-constrained LSMA), and (ANC,ASC)-constrained fully constrained least squares (FCLS) (fully abundance-constrained LSMA). So, it is not a surprise to see that the least-squares error (LSE) used as a criterion for spectral unmixing performed by LSMA can also be used as a design criterion to find endmembers where three LSE-based endmember extraction algorithms corresponding to LSOSP, NCLS, and FCLS can also be developed, unsupervised LSOSP (ULSOSP), unsupervised non-negativity constrained least squares (UNCLS) by Chang and Heinz (2000), iterative error analysis (IEA) by Neville et al. (1999), and unsupervised fully constrained least squares (UFCLS) by Heinz and Chang (2001) as their three respective representatives. All these discussions can be found in Chapter 8. According to the studies in Chang et al. (2010) and Chapter 11, the use of maximal simplex volume as a criterion is among best criteria for extracting endmembers. This makes sense from a view point of linear convexity where two constraints, ASC and ANC must be satisfied, while the OP does not satisfy any one of these two constraints. Interestingly, the convex cone-based approach such as VCA actually implements the OP criterion with ANC. As a result, in general, VCA does not perform as good as N-FINDR. However, for a convex cone to also satisfy the ASC the convex cone analysis (CCA) developed by Ifarraguerri and Chang (1999) was designed for this purpose. This similarity can also be found in LSMA where the abundance-unconstrained LSOSP, ANC-constrained NCLS and (ASC,ANC)-constrained FCLS can be considered as respective counterparts of PPI, ANC-constrained convex cone-based VCA, and (ASC,ANC)-constrained simplex-based N-FINDR. More details on endmember extraction from this perspective will be explored in Chang (2013).

Since the above-mentioned approaches, PPI, CCA, and MVT/N-FINDR are developed for finding all the endmembers simultaneously at once, their computational complexity is generally very high and expensive, particularly, for CCA and MVT/N-FINDR. To mitigate and alleviate this computational problem, two approaches have been investigated. One is to make endmember extraction a two-iterative sequential process instead of a simultaneous process. Examples include IN-FINDR, SC N-FINDR, Miao and Qi (2007), Wu et al. (2008) and Chan et al. (2009) and most recently, SQ N-FINDR in Chang et al. (2011). A second approach is to grow endmembers one at a time until it reaches the desired number of endmembers. In this regard, OP-based ATGP (Ren and Chang, 2003), convex cone-based vertex component analysis (VCA) (Nascimento and Dias, 2005), and simplex-based SGA (Chang et al., 2006) are developed for this purpose where ATGP, VCA, and SGA represent three categories of abundance constraints, no abundance constraint corresponding to OP, ANC corresponding to convex cone and both ANC and ASC corresponding to simplex. Tables 33.1 and 33.2 with their corresponding counterparts of block diagrams depicted in Figures 33.1 and 33.2 summarize relationships among algorithms developed based on three design criteria, OP, convex cone, and simplex plus LSE as well as how to find endmembers simultaneously or one after another by growing endmembers sequentially.

As recalled in PART II (Chapters 7–11) in this book, endmember extraction algorithms (EEAs) are treated on the basis of algorithmic implementation, viz. SiMultaneous EEAs (SM-EEAs), SeQuential EEAs (SQ-EEAs), initialization-driven EEAs (ID-EEAs), and random EEAs (REAAs). Tables 33.1 and 33.2 and Figures 33.1 and 33.2 provide another categorization of endmember extraction a which will be discussed in great detail in Chang (2013) which is based on various abundance constraints imposed on algorithms according to the abundance constraints imposed on EEAs from abundance-unconstrained OP-based EEAs, partially abundance-constrained convex cone-based EEAs and fully abundance-constrained simplex-based EEAs.

In addition to criteria described above, OP, linear convexity, and LSE, there is a fourth criterion that can be used to find endmembers. It considers endmembers as statistically independent random signal sources where mutual information is the criterion to be used to identify endmembers. The

Table 33.1 Least squares (LS)-based endmember extraction algorithms

Constraints	How to find endmembers	Algorithms
No constraint	Growing endmembers	ULSOSP
ANC	Growing endmembers	UNCLS
ANC, ASC	Growing endmembers	IEA/UFCLS

Table 33.2 Categories of endmember extraction algorithms using convexity geometry as a criterion

Constraints	Criterion	How to find endmembers	Algorithms
No constraint	Orthogonal projection	All p endmembers	PPI
		Growing endmembers	ATGP
ANC	Convex cone/NMF	All p endmembers	CCA
		Growing endmembers	VCA
ANC, ASC	Simplex	All p endmembers	MVT/N-FINDR
		Growing endmembers	SGA

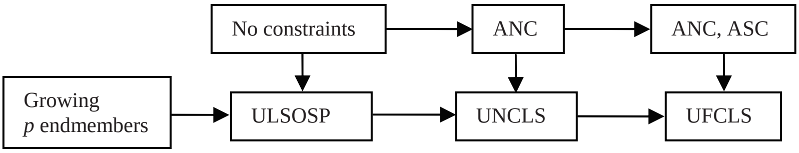


Figure 33.1 Relationships among LSE-based endmember extraction algorithms.

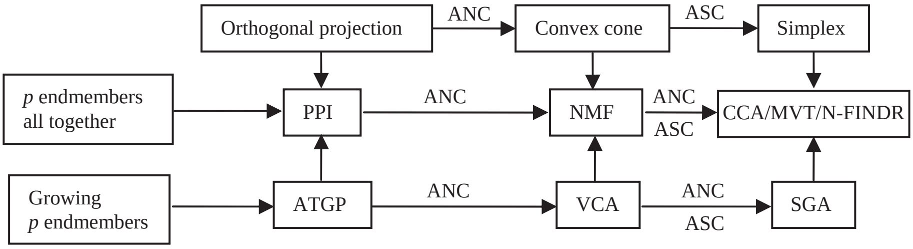


Figure 33.2 Relationships among convexity-based endmember extraction algorithms.

first work reported in the literature is the one developed by Wang and Chang (2006b), referred to as ICA-based SQ-EEA discussed in Section 8.6.5. However, the abundance vectors blindly separated by ICA are not necessarily positive signal sources. To address this need, Oja and Plumbley (2004) developed an approach to finding positive abundance vectors that satisfy the ANC. Nevertheless, it should be noted that such an approach does not satisfy the ASC. Otherwise, the separated sources will not be statistically independent.

As a final comment on endmember extraction, it has been misleading using various terminologies to represent endmember extraction. As a matter of fact, endmember extraction, endmember selection, and endmember determination are all completely different concepts. First, endmember extraction is a task to extract signatures which are supposed to be endmembers in the data without prior knowledge, while endmember selection is performed by singling out a set of potential endmember candidates which are supposed to be known a priori. In general, it is a follow-up task of endmember extraction. On the other hand, endmember determination is to determine whether a given signature is an endmember. It does not perform endmember extraction or endmember selection. Instead, it generally requires performing linear spectral unmixing (LSU) to determine if a given set of signatures used for unmixing are endmembers. It may be a main reason why endmember extraction is confused with linear spectral unmixing. In his 1999's paper (1999b), Winter proposed an autonomous spectral endmember determination where the knowledge of endmember as not provided. To resolve this issue, Winter develops endmember "finding" (Note: not endmember "extraction") via the so-called N-finder algorithm to first find potential endmember candidates using maximal simplex volume as a criterion and then further performs LSU determine whether the found potential endmember candidates are indeed endmembers. So, basically, Winter did not use LSU to find endmembers, but rather used it to determine endmembers. However, since a simplex also satisfies ASC and ANC, LSMA has been also used as a means of extraction endmembers such as iterative error analysis (Neville et al., 1999) and unsupervised fully constrained least squares (UFCLS) (Heinz and Chang, 2001). But it does not imply that endmember extraction must be implemented in conjunction with LSU or considered as a LSU technique. As a matter of fact, as shown in Chang et al. (2010) where five different criteria were investigated, using LSMA may be effective but may not be as good as maximal simplex volume for endmember extraction. On the other hand, as also shown in Chang et al. (2010), signatures used for LSU are not necessarily pure signatures as endmembers and can be mixed signatures. This provides further evidence that endmember extraction and LSMA are indeed separate techniques designed for different tasks. So, performing LSU is not a part of endmember extraction. However, when LSU is performed in an unsupervised manner when no prior knowledge of endmembers is available, endmember extraction can always be used as a preprocessing step to find potential endmembers which will be later determined by the follow-up LSU, which can be considered as a similar approach proposed by Winter's N-FINDR. Accordingly, LSU and endmember extraction can be benefited by each other.

33.3 Linear Spectral Mixture Analysis

Linear spectral mixture analysis (LSMA) is a theory developed for sub-sample and mixed sample analysis discussed in Chapter 2, whereas LSU is a technique to carry out LSMA to unmix data sample vectors via a linear mixing model into a number of basic constituent spectra assumed to make up the entire data sample vectors with appropriate abundance fractions of these constituent spectra. In spite of this distinction, these two terminologies have been used interchangeably in the literature. Depending on whether or not the signature knowledge is available, LSMA can be carried out by supervised LSMA (SLSMA) or ULSMA.

There are three crucial differences between LSMA and classification. One is that the spectral unmixing performed by LSMA produces abundance fractional maps considered as soft decisions as opposed to classification, which produces classification maps considered as hard decisions. Another difference is that LSMA only requires signature knowledge to form a linear mixture model and does not need training samples as required by classification. Even for ULSMA, the signature knowledge can be obtained by endmember extraction algorithms or unsupervised virtual signature finding algorithms (UVSFAs) developed in Chapter 17 where the only training samples are endmembers or virtual signatures themselves. So, the cross-validation used by classification via a set of training samples to evaluate performance is not applicable to LSMA. A third difference is background suppression. LSMA takes the advantage of OSP to eliminate effects of undesired signatures to improve background suppression so as to enhance unmixing ability of desired signals, a task that generally cannot be done by classification. Due to the lack of background knowledge such background suppression cannot be evaluated quantitatively but is better evaluated by visual assessment of each of abundance fractional maps. Nevertheless, the 3D ROC analysis developed in Chapter 3 can be further used as an evaluation tool to allow users to perform quantitative analysis and study of abundance fractional maps.

33.3.1 Supervised LSMA

Solving LSMA is a least squares estimation problem. It makes use of a linear mixing model specified by

$$\mathbf{r} = \mathbf{M}\boldsymbol{\alpha} + \mathbf{n} \tag{33.20}$$

where $\mathbf{M}$ is a signature matrix formed by a set of p known signatures, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ and $\mathbf{n}$ can be interpreted in various applications. When the $\mathbf{n}$ is considered as a model error, the solution to (33.20) is given by (12.24) as

$$\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r}) = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{r} \tag{33.21}$$

which is the least squares estimation of the abundance vector $\boldsymbol{\alpha}$ (Shimabukuro and Smith, 1991; Settle and Drake, 1993). On the other hand, if $\mathbf{n}$ is interpreted as a Gaussian noise in Section 12.3.3, the solution to (33.20) is the given by the Gaussian Maximum Likelihood (GML) estimator (Settle, 1996; Chang, 1998a) as

$$\hat{\alpha}^{\text{GML}}(\mathbf{r}) = \left(\mathbf{M}^T\mathbf{M}\right)^{-1}\mathbf{M}^T\mathbf{r} \tag{33.22}$$

which is the MSE estimate of abundance vector $\boldsymbol{\alpha}$ that is identical to (33.21).

As a completely different treatment, Harsanyi and Chang (1994) developed an OSP-based approach to LSMA form a signal detection point of view. Instead of considering (33.20) as an estimation problem by estimating abundance fractions of all endmembers $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ all together as (33.21) and (3.22) the OSP idea considers the problem of (33.20) as a multiple signal detection problem for p signal sources $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ present in the data where it makes use of signal-to-noise (SNR) as a criterion (also known as deflection criterion in detection theory) to measure signal detection performance. In this case, it only assumes that the noise is additive but not necessarily Gaussian. According to (33.20), the signal detection is carried out by considering $\mathbf{M}\boldsymbol{\alpha}$ as a single signal source which is actually a mixed signal source linearly combining the p signal sources, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$. Thus, it does not discriminate multiple signal sources detected in the data. To resolve this issue, OSP extends the standard single-signal detection approach by a two-stage process to solve (33.20) as a multiple-signal detection problem. More specifically, Let $\mathbf{d} = \mathbf{m}_p$ be the desired signal source to be detected. Under this circumstance, all the remaining signal sources, p signal sources $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{p-1}$, will be considered as interferers to $\mathbf{m}_p$ which must be annihilated prior to detection of $\mathbf{m}_p$ so as to increase as well as enhance detectability of $\mathbf{m}_p$. The key idea of OSP is to introduce an operator for this purpose to eliminate the influence of these $p-1$ undesired signal sources $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{p-1}$ prior to detection of the desired signal source $\mathbf{m}_p$. In other words, let $\mathbf{U} = \left[\mathbf{m}_1\mathbf{m}_2 \ldots \mathbf{m}_{p-1}\right]$ be an undesired signal source matrix. An operator defined by

$$P_{\mathbf{U}}^{\perp} = \mathbf{I} - \mathbf{U}\mathbf{U}^{\#} \tag{33.23}$$

maps all data sample vectors onto a space, $\langle \mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{p-1}\rangle^{\perp}$, orthogonal to the space linearly spanned by the $p-1$ signal sources $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{p-1}$. Taking advantage of operating $P_{\mathbf{U}}^{\perp}$ on the model in (33.20) results in

$$P_{\mathbf{U}}^{\perp}\mathbf{r} = P_{\mathbf{U}}^{\perp}\mathbf{d}\alpha_p + P_{\mathbf{U}}^{\perp}\mathbf{n} \tag{33.24}$$

where the undesired signatures in $\mathbf{U}$ have been annihilated and the original noise $\mathbf{n}$ has also been suppressed to $\tilde{\mathbf{n}} = P_{\mathbf{U}}^{\perp}\mathbf{n}$. The role of $P_{\mathbf{U}}^{\perp}$ plays in (33.24) is to remove all signal sources other than $\mathbf{d}$ from the mixed signal source $\mathbf{M}\boldsymbol{\alpha}$ so that only the desired signal source $\mathbf{d}$ will be present in $P_{\mathbf{U}}^{\perp}\mathbf{r}$. In this case, the desired signal source will also be suppressed by $P_{\mathbf{U}}^{\perp}$ as $P_{\mathbf{U}}^{\perp}\mathbf{d}$. As a consequence, (33.24) becomes a standard model commonly used in communications and signal processing to detect the single signal source $P_{\mathbf{U}}^{\perp}\mathbf{d}$. The optimal solution to (33.24) previously derived from (12.5) to (12.9) in Chapter 12 is a matched filter, $\mathrm{M}_{\mathbf{d}}$, given by

$$M_{\mathbf{d}}\left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right) = \kappa\mathbf{d}^T\left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right) = \kappa\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T\mathbf{r} \text{ for some constant } \kappa. \tag{33.25}$$

Two interpretations can be made on (33.25). One is to interpret OSP as a matched filter using the "$\mathbf{d}$" as the desired matching signal source to extract $\mathbf{d}$ from all data sample vectors $P_{\mathbf{U}}^{\perp}\mathbf{r}$ in the space $\langle \mathbf{U} \rangle^{\perp} = \langle \mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{p-1} \rangle^{\perp}$. Alternatively, OSP can also be interpreted as a matched filter which a matched signal source "$P_{\mathbf{U}}^{\perp}\mathbf{d}$" to extract the desired signal source from data sample vector $\mathbf{r}$ in the original data space. Using (33.25) the abundance fraction of the desired signal source can be detected as

$$\hat{\alpha}_p^{\mathrm{OSP}}(\mathbf{r}) = \kappa \mathbf{d}^T \left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right) = \kappa\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T \mathbf{r} \text{ for some constant } \kappa \tag{33.26a}$$

where the constant κ is generally set to $\kappa = 1$ for the purpose of detection. Accordingly, for the OSP to perform multiple signal detection, two-stage processes can be designed in sequence as follows.

The first-stage process separates a desired signal source $\mathbf{d}$ from other signal sources which form an undesired signal source matrix $\mathbf{U}$ so that an undesired signal source annihilator $P_{\mathbf{U}}^{\perp}$ is applied to all data sample vectors in the original data space to suppress the effects of the undesired signal sources on detection of the desired signal source $\mathbf{d}$. This is then followed by the second-state process, which implements a matched filter to extract the desired signal source $\mathbf{d}$. Such a two-stage process can be considered as an information-processed matched-filter which has been studied in great details in Chang (2007c). For example, since OSP assumes that the complete knowledge of signal sources, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ is known *a priori*, it can take advantage of $P_{\mathbf{U}}^{\perp}$ and $\mathbf{d}$ to carry out two-stage processes in Figure 33.3. However, in many applications such knowledge may not be available. In this case, the *a priori* knowledge of $P_{\mathbf{U}}^{\perp}$ and the desired signal source $\mathbf{d}$ must be replaced with the *a posteriori* knowledge $\mathbf{R}^{-1}$ where $\mathbf{R}$ is the auto-correlation matrix formed by all data sample vectors and the data sample vector currently being processed $\mathbf{r}$. As a result, $\hat{\alpha}_p^{\mathrm{OSP}}(\mathbf{r})$ becomes an anomaly detector, $\hat{\alpha}_p^{\mathrm{AD}}(\mathbf{r}) = \kappa\left(\mathbf{R}^{-1}\mathbf{r}\right)^T \mathbf{r} = \kappa \mathbf{r}^T \mathbf{R}^{-1}\mathbf{r}$ which is exactly so-called RX-detector developed by Reed and Yu (1990). On the other hand, the above two-stage process in Figure 33.3 can also be implemented as the one-shot operation process by linear constrained minimum variance (LCMV) filter developed by Chang (2002b) and Chang (2003a, Chapter 11) where the detection of $\mathbf{d}$ and annihilation of $\mathbf{U}$ are carried out simultaneously at the same time. Many more details can be also found in Chapter 12.

It has been shown in Settle (1996) and Chang (1998a) that the constant κ accounts for estimation accuracy and cannot be arbitrary if (33.26a) is used to estimate the abundance fraction of $\mathbf{d} = \mathbf{m}_p$ in which case κ must be set by $\kappa = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$. When OSP implements (33.26a) by setting $\kappa = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$

$$\hat{\alpha}_p^{\mathrm{LSOSP}}(\mathbf{r}) = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}\mathbf{d}^T\left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right) = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T\mathbf{r} \tag{33.26b}$$

it is called LSOSP (Tu et al., 1997), which produces exactly the same abundance fraction estimated by (33.21) and (33.22) for the endmember $\mathbf{m}_p$, $\hat{\alpha}_p^{\mathrm{LS}}(\mathbf{r}) = \hat{\alpha}_p^{\mathrm{GML}}(\mathbf{r})$ where $\hat{\alpha}_p^{\mathrm{LS}}(\mathbf{r})$ and $\hat{\alpha}_p^{\mathrm{GML}}(\mathbf{r})$ are the pth component of the estimated abundance vectors $\hat{\boldsymbol{\alpha}}^{\mathrm{LS}}(\mathbf{r}) = \left(\hat{\alpha}_1^{\mathrm{LS}}(\mathbf{r}), \hat{\alpha}_2^{\mathrm{LS}}(\mathbf{r}), \ldots, \hat{\alpha}_p^{\mathrm{LS}}(\mathbf{r})\right)^T$

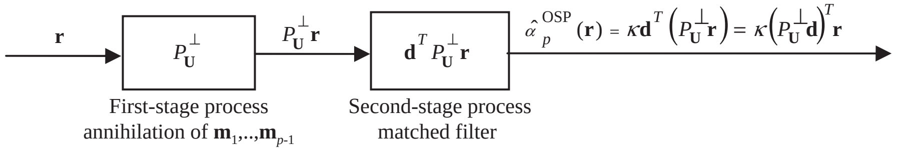

$\mathbf{r}$
$P_{\mathbf{U}}^{\perp}$
$P_{\mathbf{U}}^{\perp}\mathbf{r}$
$\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r}$
$\hat{\alpha}_p^{\mathrm{OSP}}(\mathbf{r}) = \kappa\mathbf{d}^T\left(P_{\mathbf{U}}^{\perp}\mathbf{r}\right) = \kappa\left(P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^T\mathbf{r}$
First-stage process
annihilation of $\mathbf{m}_1,..,\mathbf{m}_{p-1}$
Second-stage process
matched filter

Figure 33.3 A block diagram of $\hat{\alpha}_p^{\mathrm{OSP}}$.

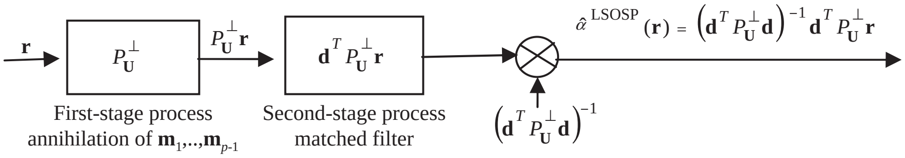

r
$P_{\mathbf{U}}^{\perp}$
$P_{\mathbf{U}}^{\perp}\mathbf{r}$
$\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r}$
$\hat{\alpha}^{\text{LSOSP}}(\mathbf{r}) = \left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{r}$
$\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$
First-stage process
annihilation of $\mathbf{m}_1,..,\mathbf{m}_{p\text{-}1}$
Second-stage process
matched filter

Figure 33.4 A block diagram of $\hat{\alpha}_p^{\text{LSOSP}}$.

and $\hat{\boldsymbol{\alpha}}^{\text{GML}}(\mathbf{r}) = \left(\hat{\alpha}_1^{\text{GML}}(\mathbf{r}), \hat{\alpha}_2^{\text{GML}}(\mathbf{r}), \ldots, \hat{\alpha}_p^{\text{GML}}(\mathbf{r})\right)^T$, respectively. The key difference between (33.26a) and, (33.21), and (33.22) is that the former only estimates the abundance fraction of one particular signal source $\mathbf{m}_p$ as opposed to the latter which estimates abundance fractions of all the p signal sources, $\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_p$ simultaneously. In other words, the LSOSP basically makes a signal detector $\hat{\alpha}_p^{\text{OSP}}(\mathbf{r})$ a signal estimator $\hat{\alpha}_p^{\text{LSOSP}}(\mathbf{r})$ in terms of converting $\hat{\alpha}_p^{\text{OSP}}(\mathbf{r})$-detected abundance fraction to the $\hat{\alpha}_p^{\text{LSOSP}}(\mathbf{r})$-estimated abundance fraction.

Furthermore, replacing the constant κ in Figure 33.3 with $\left(\mathbf{d}^T P_{\mathbf{U}}^{\perp}\mathbf{d}\right)^{-1}$ results in Figure 33.4 depicted as follows.

As noted earlier, if the *a priori* knowledge of $P_{\mathbf{U}}^{\perp}$ is not available, *a posteriori* knowledge $\mathbf{R}^{-1}$ with $\mathbf{R}$ being the autocorrelation matrix formed by all data sample vectors can be used to replace $P_{\mathbf{U}}^{\perp}$. In this case, the $\hat{\alpha}_p^{\text{LSOSP}}(\mathbf{r})$ in (33.26a) and Figure 33.4 becomes the $\hat{\alpha}_{\mathbf{d}}^{\text{CEM}}(\mathbf{r}) = \frac{\mathbf{d}^T\mathbf{R}^{-1}\mathbf{r}}{\mathbf{d}^T\mathbf{R}^{-1}\mathbf{d}}$ given by (2.33) where the desired signal source $\mathbf{d}$ is identified by a target signal source of interest, $\mathbf{t}$. Finally, it should be noted that the LSOSP, $\hat{\alpha}_p^{\text{LSOSP}}(\mathbf{r})$ bridges the gap between the OSP detector, $\hat{\alpha}_p^{\text{OSP}}(\mathbf{r})$ and LS estimator, $\hat{\alpha}_p^{\text{LS}}(\mathbf{r}) = \hat{\alpha}_p^{\text{GML}}(\mathbf{r})$. By virtue of LSOSP, the OSP can be further extended to abundance-constrained least squares methods, NCLS and FCLS methods discussed in Chapter 2 and Chang (2003a).

There are several approaches to extending SLSMA. It has been shown by Juang and Katagiri (1992) that the LSE $(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})^T(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})$ resulting (33.20) is not an appropriate criterion for classification. It is also known that Fisher linear discriminant analysis is very effective in classification due to its use of Fisher's ratio particularly designed for class discrimination. So, one approach is to replace the LSE criterion used by LSMA with classification criterion, Fisher's ratio. The resulting LSMA is called Fisher's LSMA (FLSMA) discussed in Chapter 13. It turns out that the LSE $(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})^T(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})$ used by LSMA becomes Fisher's ratio for FLSMA $(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})^T\mathbf{S}_W^{-1}(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})$ where $\mathbf{S}_W$ is the within-class scatter matrix defined in (13.1). Another is to generalize the LSE criterion $(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})^T(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})$ to a weighted LSE criterion, denoted by $(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})^T\mathbf{A}(\mathbf{r} - \mathbf{M}\boldsymbol{\alpha})$ where the weighting matrix $\mathbf{A}$ is a positive definite matrix. The resulting LSMA is called weighted abundance constrained LSMA (WACLSMA) discussed in Chapter 14. Consequently WACLSMA can be considered as a generalized LSMA which includes the standard LSMA and FLSMA as it special cases by setting $\mathbf{A} =$ identity matrix $\mathbf{I}$ and $\mathbf{S}_W^{-1}$, respectively.

33.3.2 Unsupervised LSMA

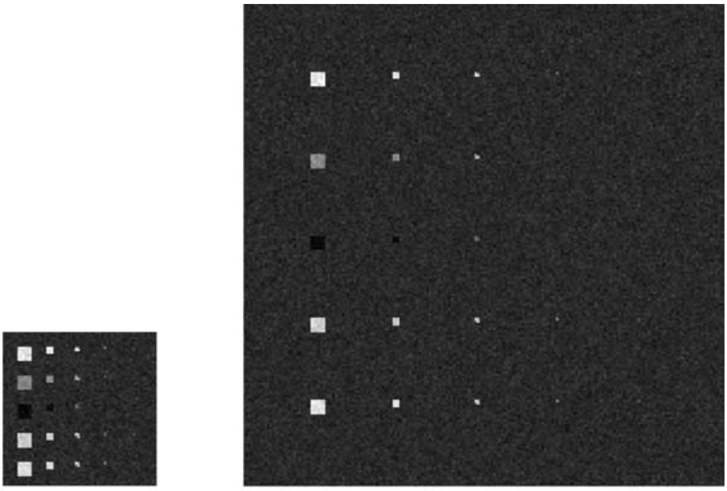
While SLSMA has been well studied in the literature, ULSMA has not received as much attention as it should have due to the following reasons. To perform LSMA effectively, the accurate signature knowledge is required. In SLSMA, such knowledge is provided by a set of constituent spectra, $\{\mathbf{m}_j\}_{j=1}^p$, referred to as endmembers in the literature, which are known provided by either prior knowledge or visual inspection. However, according to the definition in Schowengerdt (1997), an endmember must be an idealized, pure signature for a class. Unfortunately, as reported in many recent results (Chang et al., 2006; Wu et al., 2009) this is generally not true for real datasets where a true endmember may never exist. Nevertheless, this does not prevent users from using the term of endmembers. To address this issue, the desired endmembers required by LSMA must be obtained directly from the data to be processed. However, this easier said than done because finding an appropriate set of $\{\mathbf{m}_j\}_{j=1}^p$ for LSMA is very challenging and not a trivial matter with two issues involved. The first one is to determine how many signatures should be used by LSMA for spectral unmixing. The concept of VD developed in Chapter 5 is particularly designed to address this issue with extensive discussion in Section 33.1.1 where VD is defined as the number of spectrally distinct signatures, p in hyperspectral data instead of the number of endmembers. Once the value of the p is determined, the second and following issue is how to find these p signatures, $\{\mathbf{m}_j\}_{j=1}^p$. Obviously, each of $\{\mathbf{m}_j\}_{j=1}^p$ represents one distinct spectral class and is not necessarily an endmember. Some of $\{\mathbf{m}_j\}_{j=1}^p$ may be even mixed signatures. Accordingly, the term of endmembers commonly used in LSMA to represent these basic constituent spectra is misleading. A better terminology may be *virtue endmembers* (VEs) to reflect this fact that VEs represent spectrally distinct signatures in correspondence to the definition of VD. Because of that VD-estimated value is generally greater than the number of endmembers. In other words, the number of VEs, n_{VE}, is generally not the same as the number of endmembers, n_{E}. In this case, an endmember extraction algorithm may not be effective to extract all necessary VEs. The ULSMA presented in Chapter 17 is specifically developed to resolve issues of determining the value of p and finding a desired set of VEs, referred to as virtual signatures (VSs) to reflect that VSs are real pixels directly extracted in the data.

33.4 Anomaly Detection

Anomaly detection received little interest in early days in remote sensing. There are several reasons attributed to this incident. One is due to applications which are mainly focused on geographic information processing such as land cover/use classification. Another is the use of multispectral imagery which has very low spatial and spectral resolutions. As a consequence, anomalies may be very much likely either embedded or mixed with other material substances in a single pixel vector. Under such circumstances, anomalies may have been contaminated or smeared by other dominant substances. Third, most remote sensing image processing techniques developed for such applications are spatial domain-based methods which are designed to take advantage of spatial correlation to perform image analysis. Since anomalies usually appear in a very limited spatial extent, such spatial domain-based methods can hardly capture their existence. Finally, anomalies do not provide much information to multispectral image analysts who are more interested in geographic information rather than target information. However, the advances of hyperspectral imaging sensors have revolutionized the way to process multispectral imagery. Due to the very high spectral resolution provided by hyperspectral sensors, many subtle substances that are generally unknown *a priori* or cannot be identified by visual assessment can now be uncovered for data analysis. On many occasions such substances are most interesting targets and provide crucial and critical information that can assist image analysts to solve many problems which cannot be resolved by multispectral imaging processing analysis. These applications may include agriculture, ecology, geology, environmental monitoring, law enforcement, military, and medical diagnosis. For example, subtle targets of interest can be special species in agriculture, unusual migrations in ecology, rare minerals in geology, toxic wastes in environments, drug trafficker or smugglers in law enforcement, vehicles/tanks in battlefields, cancerous cells or tumors in medical diagnosis, etc., just name a few.

Such target signal sources generally appear in a form of abnormalities that are distinct from their surroundings. Therefore, how to characterize these types of target signal sources becomes crucial in image analysis. Three features can be suggested to characterize these target signal sources as anomalies: (1) existence, (2) presence, and (3) population. The reason of being anomalies is because they are not known *a priori*. On the other hand, anomalies usually occur with low probabilities. Therefore, their existence generally cannot be detected by any supervised means or visual inspection. As for presence, the spatial extent of anomalies is generally overlooked since they can be present as subpixel targets with their size smaller than pixel size or as mixed pixels mixing with the background or other substances. Most importantly, once anomalies do appear, their population cannot be large due to the nature of being anomalies. As a result, anomalies are generally considered as insignificant targets compared to the entire data. Unfortunately, several issues arising in anomalies have not been investigated or explored in the past. First, how large is it for a target to be considered as an anomaly in terms of its size? Second, how much spectral distinction is it for an anomaly to respond to its surrounding neighborhood? Third, how much sensitivity is it for an anomaly to noise? Fourth, how effective is it for an anomaly to distinguish itself from other anomalies? In particular, if two or more anomalies are close together, how can these anomalies be detected as separate anomalies? Finally, how can these anomalies be detected effectively by taking into account all the above-mentioned issues?

Many algorithms have been developed for anomaly detection over the past years and can be roughly categorized into two classes, second-order statistics methods and high-order statistics methods in terms of spectral statistics among data sample vectors. The detectors in the first class can be considered as either Mahalanobis distance-based filters which are variants of an algorithm developed by Reed and Yu (1990), referred to as RX detector (RXD) (Chang, 2003a) or matched filter-based detectors. The RXD-like detectors are primarily derived from generalized likelihood ratio test and adaptive subspace detectors via Gaussian noise assumption. One the other hand, the matched filter-based detectors do not make such assumption and simply makes use of the data sample spectral correlation matrix to whiten the sample data prior to implementation of a matched filter. Interestingly, the detectors of both types arrive at the same functional form due to the use of second-order statistics. The detectors in the second class take advantage of higher order statistics such as skewness (third order), kurtosis (fourth order), statistical independence (mutual information), or projection pursuit (relative entropy). It has been shown in Chang and Hsueh (2006) that when an early detected anomaly had a strong signature, it would affect detectability of follow-up anomalies, specifically for those anomalies with weak signatures. One way to mitigate this problem was to perform target discrimination once an anomaly was detected before searching for next anomalies. Such anomaly detection combined with target discrimination gives rise to anomaly classification (Chang and Chiang, 2002). However, in doing so, the process must be carried out in real time which cannot be implemented by most anomaly detection algorithms such as the RXD. The "real-time" referred here means that the pixel to be processed must be carried out in a causal manner that the pixels involved in data processing are only those that were already processed. To resolve this issue, a causal RX detector (CRXD) was further developed in to make the RXD a real-time processor. Recently, an adaptive anomaly detector, referred to dual window-based eigen separation transform (DWEST) was developed by Kwon et al. (2003) to address local properties of anomalies using an inner/outer window approach, and it was further investigated by a nested spatial window-based approach (Liu and Chang, 2004). These window-based adaptive anomaly detectors were designed to adapt local variability. However, some other issues such as size of anomalies, the response of anomalies to their surroundings, sensitivity of anomalies to noise, still remain and unsolved. One is of particular interest, "what is an anomaly?" In other words, what types of targets are considered as anomalies? A first attempt to address this issue was made in



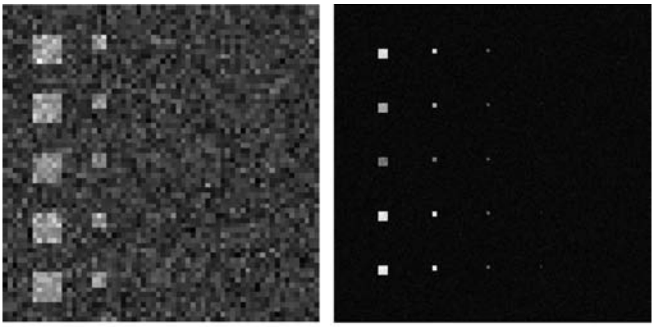
(a) Image of 64 × 64 pixels

(b) Image of 200 × 200 pixels

Figure 33.5 Target panels with four various sizes implanted in two uniform image backgrounds with sizes of 64 × 64 pixel vectors and 200 × 200 pixel vectors.

Chang and Hsueh (2006), which concluded that an anomaly should be closely related to its size relative to the image to be processed. The following simple example provides a clue of how controversial this issue is. Figure 33.5(a) and (b) shows a set of the same various target panels with four different sizes implanted in two simulated Gaussian noise-corrupted image backgrounds with sizes of 64 × 64 pixel vectors and 200 × 200 pixel vectors, respectively, where the five panels in the first column are size of 6 × 6 pixel vectors, the five panels in the second column are size of 3 × 3 pixel vectors, the five panels in the third column are size of 2 × 2 pixel vectors, and then the five panels in the fourth column are size of 1 × 1 pixel vectors.

The target implantation is done by replacing the background pixels with target panel pixels. Figure 33.6(a) and (b) shows the results of operating RXDF on these two images in Figure 33.5(a) and (b), respectively.



(a) Image of 64 × 64 pixels

(b) Image of 200 × 200 pixels

Figure 33.6 Results of operating RXD on images in Figure 33.3(a) and (b).

An immediate finding by comparing the results in Figure 33.6(b) to that in Figure 33.6(a) leads to an interesting observation: the target panels of sizes 2 × 2 and 1 × 1 that are detected by RXDF in Figure 33.6(b) as anomalies now become undetectable and are no longer anomalies in Figure 33.6(a) where two images in Figure 33.6(a) and (b) are shown in the same size for clear and better visual assessment. Moreover, the target panels of sizes 6 × 6 and 3 × 3 detected in Figure 33.6(a) also become smeared and blurred due to noise compared to their counterparts in Figure 33.6(b) which are detected clearly as anomalies. Why does the same RXDF produce so different results for the same set of target panels except that the processed image size is different? This simple example sheds light on a tricky issue in anomaly detection, "what does really it mean by anomaly?" Some answers can be found in the work by Chang and Hsueh (2006).

Despite the fact that many approaches described above have been developed for anomaly detection, an interesting approach to anomaly detection has been recently proposed by Liu and Chang (2008). It is called multiple-window anomaly detection (MWAD) which implements multiple a set of multiple windows to detect targets with varying sizes in order to avoid being trapped in the controversial definition of anomaly as demonstrated by the example presented in Figures 33.5 and 33.6. The idea is inspired by Figure 33.6(a) and (b), where the same target panels implanted in two different image sizes are detected by the same RXD in different abundance fractions. Since the size of the image to be processed must be fixed, it cannot be changed in the same way that is conducted in Figure 33.6 by the same RXD. To resolve this issue, the images processed in Figure 33.6(a) and (b) by RXD are now replaced by the images that are used to form the sample spectral correlation matrix for RXD to mimic two different image sizes. In other words, our proposed idea makes RXD adaptive to variable sizes of the sample covariance matrix used in RXD while keeping the original image size unchanged. More specifically, the sample covariance matrix used by RXD is formed by only image pixels within a designated window, in which case the window size determines the sample covariance matrix. Accordingly, adjusting the size of the sample covariance matrix used in RXD is similar to adjusting the original image size except that the sample covariance matrix used by RXD now varies with each image pixel vector that it processes. With this interpretation, the Reed-Yu's RXD can be viewed as a global anomaly detector where the entire image is considered as a single window to form the sample covariance matrix to capture the spectral variation provided by the entire image. More details about using various windows to perform anomaly detection are included in Chang (2013).

For many applications such as military/intelligence gathering or combat fields anomalies generally appear as moving targets. Therefore, developing real time processing algorithms for anomaly detection is highly desirable and critical. A detailed study on this subject can be found in Chang (2013).

33.5 Support Vector Machines and Kernel-Based Approaches

In remote sensing image classification, one of most widely used techniques is maximum likelihood classifier (MLC), which is based on Mahalanobis distance. Since MLC is essentially a weighted distance measure with the weighting matrix specified by sample covariance matrix, it is not particularly designed for classification. Consequently, it does not necessarily yield the best classification. It has been shown in pattern classification that Fisher's linear discriminant analysis (FLDA) is one of best classifiers due to the fact that the Fisher's ratio used by FLDA as a criterion is specifically to be designed for classification where the Fisher ratio is defined by a ratio of the between-class scatter matrix to the within-class scatter matrix. In both cases, these two classifiers require second-order statistics to calculate the weighting matrix for MLC and scatter matrices for FLDA. Accordingly, MLC or FLDA may not work effectively if either data statistics cannot be well characterized by second-order statistics or the second-order statistics used to characterize the data are not reliable. In the former case, classifiers must go beyond second-order statistics such as high-order statistics-based classifiers, projection pursuit, neural networks, etc. As for the latter case, two circumstances may occur. One is that the used training samples are not appropriately selected. The other is that the pool of selected training samples is too small to constitute credible statistics. The support vector machines (SVMs) arises from such needs. First of all, since an SVM is designed to find an optimal hyperplane to maximize the margin between two classes of training samples, referred to as support vectors for separation, it does not require a large number of training samples to achieve the best maximal separation margin. As a result, the SVM performance in classification is very sensitive to selection of support vectors. For example, consider the following example shown in Figure 33.7 where two datasets are denoted by S_1, which consists of all data sample vectors belonging to two classes marked by open circles "$\circ$" and crosses "$\times$" and S_2 which is set S_1 with exclusion of two data sample vectors, $D_\circ$ and $F_\times$ where the subscripts "$\circ$", and "$\times$" indicates classes to which theses support vectors belong. In addition, two sets of training samples $\{A_x, B_\times, G_\circ, H_\circ\}$ and $\{C_\times, E_\circ\}$ defined as support vectors are used for training SVMs.

Assume that the dotted line in Figure 33.7 is the optimal hyperplane. There are four cases worth being discussed.

1. Implement an SVM using $\{C_\times, E_\circ\}$ as support vectors to classify the dataset S_2 in which case it reaches a perfect classification.
2. Implement an SVM using $\{A_x, B_\times, G_\circ, H_\circ\}$ as support vectors to classify the dataset S_2 in which case the data sample vectors are labeled by $C_\times$ and $E_\circ$ are within the margin but still in correct sides of the optimal hyperplane.
3. Implement an SVM using $\{C_\times, E_\circ\}$ as support vectors to classify the dataset S_1 in which case the SVM misclassified $D_\circ$ and $F_\times$.
4. Implement an SVM using $\{A_x, B_\times, G_\circ, H_\circ\}$ as support vectors to classify the dataset S_1 in which case the SVM not only misclassified $D_\circ$ and $F_\times$, but also $C_\times$ and $E_\circ$.

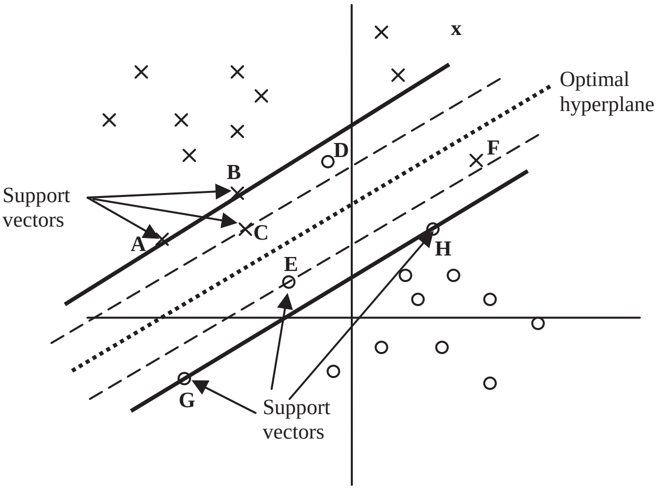


Figure 33.7 Different sets of support vectors used by SVMs.

Cases 1 and 2 demonstrate the significance of selecting appropriate support vectors. In addition, these two cases also show that if the worst sample vectors are selected as support vectors, SVM performs its best where in our cases, the set of support vectors, $\{C_\times, E_\circ\}$ is the worst sample vectors compared to the set of $\{A_x, B_\times, G_\circ, H_\circ\}$ which are not. Because of that the SVM can be considered as the worst-case classifier compared to the statistics-based MLC and FLDA, which can be considered average-case classifiers. To penalize those training samples which are inappropriately selected, a set of slack variables ξ_i specified by (2.63) in Chapter 2 are introduced in Case 2 to account for the costs incurred by each of these training samples with $0 \leq \xi_i \leq 1$.

Cases 3 and 4 demonstrate linear nonseparability encountered in pattern classification where linear classifiers have difficulty with classifying certain data sample vectors using linear decisions regardless of how training samples are selected. In Figure 33.7, $D_\circ$ and $F_\times$ are such data sample vectors. Like Case 2, a set of slack variables ξ_i must be introduced in Cases 3 and 4 to penalize those training samples which are linear nonseparable to account for their costs with $0 \leq \xi_i \leq 1$ and $\xi_i > 1$, respectively. So, the four cases can be summarized as follows.

1. Case 1 achieves perfect classification without a need of slack variables.
2. Case 2 requires a set of slack variable $\{\xi_i\}$ to penalize inappropriately selected training samples, $\{A_x, B_\times, G_\circ, H_\circ\}$, with $0 \leq \xi_i \leq 1$. However, if $C_\times$ and $E_\circ$ are used as support vectors, the SVM can achieve perfect classification without using slack variables as Case 1.
3. Case 3 requires a set of slack variables $\{\tilde{\xi}_i\}$ to penalize linear nonseparable training samples, $D_\circ$ and $F_\times$ with $\tilde{\xi}_i > 1$.
4. Case 4 requires a set of slack variables, $\{\xi_i\} \cup \{\tilde{\xi}_i\}$ to penalize inappropriately selected training samples $C_\times$ and $E_\circ$ and linear nonseparable training samples, $D_\circ$ and $F_\times$, respectively, where a set of slack variable $\{\xi_i\}$ with $0 \leq \xi_i \leq 1$ similar to Case 2 is used to penalize inappropriately selected training samples, $C_\times$ and $E_\circ$ and another set of slack variables with $\tilde{\xi}_i > 1$ similar to Case 3 to penalize linear nonseparable training samples, $D_\circ$ and $F_\times$. In this particular case, a kernel must be used to resolve linear nonseparability issue to account for the case of using a set of slack variables with $\tilde{\xi}_i > 1$.

As noted in Figure 33.7, it is impossible to find linear lines to separate the two training samples, $D_\circ$ and $F_\times$ no matter how linear decision boundaries are drawn in the data space. This implies that using slack variables to penalize such training samples still cannot change the fact that, $\{A_x, B_\times, G_\circ, H_\circ\}$, SVM is a linear machine. To further resolve linear nonseparable problems, kernel-based approaches have been developed for this purpose. The key idea is to map the original data space into a high dimensional feature space via a nonlinear kernel mapping so that linear nonseparable problems in the original space can be resolved in this new feature space. Generally speaking, two types of kernel-based approaches have been investigated. One is kernel-based transformations which map component analysis-based transformations into a feature space such as kernel-based PCA (KPCA) (Scholkopf et al., 1999b) described in Section 30.4.2, which is the most popular kernel-based component analysis transformation. Its idea is to kernelize the projection vectors produced by a component analysis in the original space. Using the PCA as an illustrative example, the eigenvectors considered as projection vectors of the PCA are kernelized via Equations (30.9)–(30.13) by the sample covariance matrix formed by all the kernelized data sample vectors in the new feature space. However, one of major difficulties with such kernel-based transforms is computational complexity due to its use of all data sample vectors to perform the kernelized component analysis transformation. If the number of data sample vectors is huge, kernel-based transformations will incur computational problems which require tremendous computer power to find kernelized projection vectors. This is a severe obstacle to overcome for hyperspectral imagery whose data size is generally very large. This may be one of reasons that kernel-based transformations are not preferred in applications of hyperspectral data exploitation.

As an alternative, a second kernel-based approach is to kernelize the training samples to be used for classification instead of kernelizing component analysis transformation for entire data space. One of significant advantages resulting from such training sample-kernelization is that only training samples rather than entire data set are required to perform kernelization. This approach turns out to be a major trend in developing new hyperspectral imaging algorithms, for example, kernel-based SVM (KSVM) and kernel-based FLDA (KFLDA) discussed in Chapter 2. Similarly, kernel-based FLSMA (KFLSMA) and kernel-based WACLSMA (KWACLSMA) can also be derived for FLSMA and WACLSMA respectively by the same treatment (Liu, 2011). Nevertheless, it should be noted that the benefit of using the second training samples-based approach to kernelization is also traded for a need of judicious selection of training samples which involves two issues, how many training samples are sufficiently enough and how these training samples are selected. In addition, there are also two issues in using kernels as well. One is selection of parameters and their appripriate values for kernels to be used. The other is computational complexity which is generally very high since the kernel-mapped feature space usually has very high dimensions. In supervised classification, these two issues in selection of training samples are resolved by prior knowledge obtained from ground truth such as spectral libraries, data bases, or visual inspection. But as noted in Section 33.2.2, supervised classification may not be reliable if the training samples are neither representative nor accurate. This is often the case in real applications, specifically for hyperspectral images where many subtle material substances cannot be identified by ground truth and visual assessment. Accordingly, unsupervised classification may be a better approach. However, for unsupervised classification to be effective, the above-mentioned two issues must be appropriately addressed. In doing so, one is to reduce the prior knowledge as much as possible so that its impact on classification can be minimal. The constrained minimum energy (CEM) described in Chapter 2 is derived for this purpose where only desire target knowledge is required. The other is anomaly detection such as RX detector (RXD) developed by Reed and Yu (1990) discussed in Section 33.4 where no prior knowledge is required at all. Unfortunately, when CEM and RXD are extended to their kernel counterparts, KCEM and KRXD also run into the same issues that the kernel-based component analysis transformations have where the sample covariance/correlation matrix must be calculated from the entire dataset to account for *a posteriori* information. To mitigate this dilemma, current research being conducted in the Remote Sensing Signal and Image Processing Laboratory (RSSIPL) at University of Maryland, Baltimore County (UMBC), has been directed to addressing the issue of finding unsupervised training samples which can eventually solve issues arising in the kernel-based component analysis transformations. Most recently, a third approach is to kernelize classifiers instead of training samples where LSMA can be further extended to kernel-based LSMA where the three spectral unmixing methods, LSOSP, NCLS, and FCLS are extended to their kernel counterparts in Chapter 15. Using similar ideas, the three least-squares unsupervised algorithms, ATGP, UNCLS, and UFCLS, developed for finding unsupervised target signal sources in Chapters 8 and 17 can also be extended to kernel-based ATGP (KATGP), kernel-based UNCLS (KUNCLS), and kernel-based UFCLS (KUFCLS) (Wong, 2011).

Finally, an interesting approach to extending kernel-based classifiers on the used kernels is worth pursuing (Liu, 2011). In Section 2.4.1, three types of commonly used kernels, radial basis function (RBF) kernel, polynomial kernel, and sigmoid function kernel, are described. Among them the RBF kernel given by

$$\exp\left(-\frac{1}{2\sigma^2}||\mathbf{x}-\mathbf{y}||^2\right) \tag{33.27}$$

is the most popular one used by many kernel-based classifiers. Analogous to the treatment for the WACLSMA presented in Chapter 13, Equation (33.27) can be further extended to weighted RBF kernel by including a weighting matrix **A** as follows:

$$\exp\left(-\frac{1}{2\sigma^2}(\mathbf{x}-\mathbf{y})^T\mathbf{A}(\mathbf{x}-\mathbf{y})\right) \tag{33.28}$$

where **A** is a positive definite matrix similar to the one defined in (14.2). Three special cases can be derived from (33.28):

1. When $\mathbf{A}=\sigma_{\mathbf{A}}^2\mathbf{I}$, Equation (33.28) is reduced to (33.27) where $\sigma_{\mathbf{A}}^2$ is different from σ^2 in (33.28).
2. When $\mathbf{A}=\mathbf{R}^{-1}$ or $\mathbf{K}^{-1}$, Equation (33.28) can be considered as Mahalanobis distance kernel.
3. When **A** is set to $\mathbf{S}_W^{-1}$, which is the inverse of the within-class scatter matrix, $\mathbf{S}_W$ defined in (13.1), Equation (33.28) becomes Fisher's kernel similar to the one developed for Fisher's LSMA (FLSMA) in Chapter 14.

33.6 Hyperspectral Compression

Hyperspectral compression becomes a necessity when hyperspectral data volume goes beyond the capability of used computers. This is particularly crucial for space-borne hyperspectral imaging sensors with limited computing power due to the payload constraint. In addition, due to limitation on bandwidth and data transmission rate, data preprocessing is generally required prior to data being down-linked to the ground stations for transmission and communication. Data compression provides an effective means of reducing data volumes and can be interpreted in various ways. From an information theory point of view, data compression performs data compaction with either no loss of entropy or entropy reduction. From a signal/image processing point of view, data compression performs lossless and lossy coding in the sense of some optimal criteria such as MSE, SNR, and peak SNR (PSNR). Eventually, the effectiveness of a data compression technique must be evaluated by a certain performance measure. One generally used for this purpose is compression ratio (CR), which is defined by the ratio of the original data size to compressed data size. Unfortunately, this criterion may not be applicable to hyperspectral data exploitation where subsample and mixed-sample signal sources may be inadvertently suppressed if no extra care is taken due to the fact that such signal sources may occupy only a single data sample vector or very few data sample vectors. To address this issue, Chapter 19 develops the so-called exploitation-based hyperspectral data compression (EHDC) for hyperspectral information compression (HIC), which performs compression in terms of information not data itself such as CR. Since how much spectral information provided by hundreds of spectral bands to be retained primarily varies with applications in data exploitation, EHDC is particularly designed for this purpose to meet various tasks. Because of that, such an information compression is called exploitation-based compression in terms of preserving information rather than reducing data size. To accomplish this goal, Chapters 20 and 21 are developed to derive two types of EHDC for HIC, DR, and band selection (BS) with a significant difference between them. The main task of DR is to reduce the original data space to a lower data space via various means such as transformations (e.g., PCA), feature extraction (e.g., FLDA), and classification (e.g., LSMA). This type of EHDC compacts the complete set of data sample vectors in the original data into a manageable data space in a lossless or lossy manner so that the desired spectral information can be retained and preserved for future data exploitation. In contrast to DR, BS performs a completely different form of EHDC, which only selects a subset of desirable spectral bands from the full band set. So, technically speaking, BS actually performs data reduction instead of data compression since no compression is involved in data processing other than BS. On the other hand, BS is in fact an EHDC technique because it performs information compression by only selecting a desired set of spectral bands for information preservation while completely discarding the information provided by unselected spectral bands.

One common issue in implementing DR and BS is required knowledge of how many dimensions needed to be retained, q or how many bands needed to be selected, $\tilde{q}$, *a priori*. Another common issue is the DR-retained dimensions and BS-selected bands are determined by and should vary with the values of q and $\tilde{q}$. In other words, when these values are changed, all the DR-retained dimensions and BS-selected bands are also different. As a consequence, traditional processes to perform DR and BS may be applicable but may not be effective. In addition, since DR performs a transformation to compact the entire original data into a new transformed data space compared to BS, which only selects desired subset from a full set of bands where each band has its own information that may not be shared by other bands, the number of bands required by BS, $\tilde{q}$, is generally greater than the number of DR-transformed dimensions, q. EHDC develops a new concept of prioritizing DR-processed dimensions and BS-selected spectral bands in Chapters 20 and 21 to resolve this dilemma. The key idea is to design prioritization criteria to produce priority scores that can be used to rank the order of DR-processed dimensions and BS-selected spectral bands. By virtue of this ranking order DR and BS can be performed progressively in the sense that previous selected dimensions and bands can be used for dimensionality expansion and reduction. To materialize such a utility two dual processes, progressive dimensionality expansion and progressive dimensionality reduction are then derived to perform progressive DR and BS in a forward or a backward manner. It is worth noting that the progressive processes presented in Chapters 20 and 21 are different from sequential processes commonly used in signal and image processing, which process data sample vectors one at a time in a one-shot process but do not process all data sample vectors in a progressive fashion where subsequent processes improve the results produced by previous processes. Theoretically, the progressive dimensionality expansion process can start off with the first dimension or the first band until it reaches the last dimension or last band. Such a progressive process operates in a forward manner by expanding dimensionality. Similarly, the progressive dimensionality reduction process can carried be out in an exactly reverse order where the progressive process operates in a backward manner by starting full bands and the gradually reducing dimensionality. Obviously, this progressive process can be made more efficient and effective if it does not always begin with the first dimension or the first band for dimensionality expansion and with the last dimension and last band for dimensionality reduction. In this case, a feasible range for selected dimensions and bands can be very useful. The VD developed in Chapter 5 seems to provide good estimates for lower and upper bounds on this range as demonstrated by experiments in Chapters 20 and 21 where the initial dimensionality estimates for both bounds can be determined by the value of VD, denoted by n_{VD}, as a lower bound, while the upper bound can be determined by $2n_{\mathrm{VD}}$, that is, $[n_{\mathrm{VD}}, 2n_{\mathrm{VD}}]$.

Up to now, the traditional DR and BS are performed by fixing the value of q or $\tilde{q}$ at a constant value during data processing. In reality, this may not be necessarily the case. For example, in LSMA different signatures pose different degrees of difficulty to be unmixed. Therefore, unmixing all signatures using the same number of dimensions or bands may not be adequate. Accordingly, q and $\tilde{q}$ should be considered as variables and can adapt their values with different target signature signal sources for data analysis, even for the same application. In this case, the traditional DR and BS are not applicable to such dynamic changes. Fortunately, the two dual progressive processes for dimensionality reduction and dimensionality expansion developed in Chapters 20 and 21 offer a solution. Despite that n_{VD} and $2n_{\mathrm{VD}}$ can be used to provide lower and upper bounds on these two dual progressive processes to perform dimensionality expansion and reduction, it will be interesting and desirable to see if the range of $[n_{\text{VD}}, 2n_{\text{VD}}]$ can be narrowed down to a reason estimate to save further computational complexity and cost because on some occasions the value of VD, the range of $[n_{\text{VD}}, 2n_{\text{VD}}]$, may be large. For example, for the Cuprite reflectance data in Figure 1.12, the value of VD is estimated as $n_{\text{VD}} = 22$ where $[n_{\text{VD}}, 2n_{\text{VD}}]$ and the range may be still too broad. In doing so a new concept, referred to as dynamic dimensionality allocation (DDA) is further developed in Chapter 22. By taking advantage of DDA, the traditional DR and BS, referred to as fixed-dimensionality reduction (FDR) and fixed dimensionality band section (FDBS) can be extended to variable dimensionality reduction (VDR) and variable dimensionality band selection (VDBS), respectively.

Suppose that there are p spectral signatures, $\{\mathbf{m}_j\}_{j=1}^{p}$ assumed to be present in the data and d_j is the band dimensionality required for $\mathbf{m}_j$ to be discriminated from other spectral signatures $\{\mathbf{m}_i\}_{i=1,i\neq j}^{p}$. Then DDA is developed to produce a variable dimensionality d_j for VBDS to allocate a number of spectral bands particularly for the signature $\mathbf{m}_j$ where $d_j = n_{\text{VD}} + q_j$. The idea of DDA originates from the Hamming code, which uses 4 information bits and 3 parity check bits for single-bit correction and double-bit detection. Once again, the interpretation of the pigeon-hole principle can also be used to shed light on the concept. Assume that the use of a spectral band is dictated by a binary bit with "1" being "in use" and "0" being "no use." Then the value of VD, n_{VD} in d_j indicates how many information bits required to retain the desired information provided by $\{\mathbf{m}_j\}_{j=1}^{p}$ and q_j additional bands in d_j produced by DDA, referred to as parity-check bands correspond to parity-check bits to be used by the Hamming codes to clarify the confusion among the p spectral signatures, $\{\mathbf{m}_j\}_{j=1}^{p}$. With this interpretation, the upper bound $2n_{\text{VD}}$ can be actually decreased to $n_{\text{VD}} + \log n_{\text{VD}}$ so that the range of $[n_{\text{VD}}, 2n_{\text{VD}}]$ can be further reduced to $[n_{\text{VD}}, n_{\text{VD}} + \log n_{\text{VD}}]$. However, it should be noted that the benefit of using parity-check bands is derived from a need of parity signature discrimination or classification. If an application only involves signatures without performing further signature analysis such as endmember extraction, target detection, then finding q_j may not be necessary because there is no need for parity check bands and n_{VD} may suffice for BS.

Similarly, the exactly same concept of DDA can also be applied to DR where the parity-check bands are now defined as parity-check transformed dimensions. However, as shown in Safavi (2010), the q_j produced in d_j by DDA for an additional number of parity-check transformed dimensions did not offer the same benefits for DR as it did for BS in Chapter 22. There is one major reason attributed to the ineffective use of q_j. For BS case, each spectral band is considered as a separate signal source providing a different level of spectral information. This is not true for DR because each dimension is a new dimension transformed by the complete set of data sample vectors. As a result, each transformed dimension already contains necessary discrimination information among $\{\mathbf{m}_j\}_{j=1}^{p}$. Therefore, additional q_j parity-check transformed dimensions do not necessarily increase discriminatory power among $\{\mathbf{m}_j\}_{j=1}^{p}$. However, it does not imply that DR cannot be benefited by DDA. Besides, due to nature of DR the information in a lower transformed data space is always included in a higher transformed data space compared to BS with each band providing its own information that may not be shared by other bands. Accordingly, parity-check bands works more effectively for VDBS than parity-check transformed dimensions for VDR. Nevertheless, the additional parity-check transformed dimensions may not have lost their advantages in DR. For example, according to the experiments conducted in Safavi (2010), LSMA was supervised with the signature matrix provided by prior knowledge which remains the same during the DR. In reality the knowledge of the signature matrix $\mathbf{M}$ is unknown, and the signatures in $\mathbf{M}$ must be obtained directly from the data and the $\mathbf{M}$ should adapt to the values of q. Under such circumstance, the parity-check transformed dimensions derived from DDA will provide benefits in reducing instability and uncertainty caused by unsupervised knowledge. How much effect or impact on improvement is yet to be seen and needs further investigation.

The concepts of VDR and VDBS were first introduced in conjunction with DDA as dynamic dimensionality reduction and dynamic band selection in Safavi et al. (2011) and Liu et al. (2011), respectively, where no de-correlation of dimensions and bands was included to remove possible redundant dimensions and bands. As noted earlier, dimensionality de-correlation may not be of great concern since a projection pursuit-based DR technique that produces mutually OP index components can be considered as a dimensionality de-correlation process. However, this is not true for BS where the selected bands are original spectral bands which have not been processed by any de-correlation technique. As a result, if a band is selected because of its high priority score, its adjacent bands will also be very likely to be selected due to the fact that they are closely correlated with the selected band and may also have high priority scores. In order to address this issue, a new approach to BS, called progressive band selection (PBS), is particularly investigated in Chapter 23 as a result of PBDP implemented in conjunction with DDA and band de-correlation during the process of BS. More details on progressive band processing can be found in Chang (2013).

33.7 Hyperspectral Signal Processing

The hyperspectral data considered in Chapters 7–23 are three-dimensional image cubes where each data sample is actually a pixel vector. So, the data processing carried out in this manner can be viewed as hyperspectral image processing in Category A where two types of correlation are of interest. One is correlation provided by data samples in terms of their spatial locations while paying no attention to interband spectral correlation. This is generally referred to as intersample spatial correlation and is commonly used in traditional image processing to develop spatial domain-based algorithms to perform tasks such as edge detection, region growing, clustering, segmentation, etc. Early data processing for remote sensing data, for example, geographical information system (GIS), belongs to such an approach, which can be considered as spatial domain-based data analysis. The other type of correlation is provided by data samples regardless of their spatial location. It is a complete opposite to the above-mentioned intersample spatial correlation and can be referred to as intrasample spectral correlation which has been explored and investigated in great detail in Chapter 17. The key difference between these two types of correlation can be well explained by the following example. Let $S = \{\mathbf{r}_i\}_{i=1}^{N}$ be a set of N data sample vectors where $\mathbf{r}_i = (r_{i1}, r_{i2}, \ldots, r_{iL})^T$ is the ith data sample vector, L is the total number of spectral bands, and the i indicates the ith spatial location of $\mathbf{r}_i$. Let $\{P(1), P(2), \ldots, P(N)\}$ denote a permutation of N spatial locations, $\{1, 2, \ldots, N\}$. There are $N!$ permutations of $\{1, 2, \ldots, N\}$ with $\mathbf{S}^P = \left[\mathbf{r}_{P(1)}\mathbf{r}_{P(2)} \cdots \mathbf{r}_{P(N)}\right]$ denoting one data matrix with data samples arranged in a particular order of $\{1, 2, \cdots, N\}$ specified by a permutation $\{P(1), P(2), \cdots, P(N)\}$. Now, the intersample correlation matrix and intrasample correlation matrix provided by $\mathbf{S}^P$ are $(1/N)\sum_{i=1}^{N} \mathbf{S}^P\left(\mathbf{S}^P\right)^T$ and $(1/N)\sum_{i=1}^{N} \mathbf{r}_{P(i)}\mathbf{r}_{P(i)}^T$, respectively. In comparison between these two sample correlation matrices, it is very clear that $\sum_{i=1}^{N} \mathbf{r}_{P(i)}\mathbf{r}_{P(i)}^T$ is independent of permutations and remains the same, that is, $\sum_{i=1}^{N} \mathbf{r}_{P(i)}\mathbf{r}_{P(i)}^T = \sum_{i=1}^{N} \mathbf{r}_i\mathbf{r}_i^T$ for any permutation P even though the samples reshuffled by permutations, while $\sum_{i=1}^{N} \mathbf{S}^Q\left(\mathbf{S}^Q\right)^T$ will result in another different sample spatial correlation matrix if another permutation Q is used. That is, $\sum_{i=1}^{N} \mathbf{S}^P\left(\mathbf{S}^P\right)^T \neq \sum_{i=1}^{N} \mathbf{S}^Q\left(\mathbf{S}^Q\right)^T$ if P and Q are two different permutations of $\{1, 2, \ldots, N\}$ since it alters spatial correlation among data samples when data samples are re-arranged in different spatial coordinates. Nevertheless, intersample spatial correlation and intrasample spectral correlation share one thing in common, that is, they both produce sample statistics when $N > 1$. In this case, the intersample spatial correlation and intrasample spectral correlation are more specifically referred to as sample intersample spatial correlation and sample intrasample spectral correlation to indicate the vital role of sample size plays in generation of sample statistics by $S = \{\mathbf{r}_i\}_{i=1}^{N}$. Throughout this book, only sample intrasample spectral correlation is of interest, specifically, interband spectral information (IBSI) introduced in Chapter 17. Three scenarios can be developed to effectively use such IBSI(S) with various sizes of training sample pool S and can be described as follows:

1. The size of S, $N = |S|$, is very large:
   In this scenario, the sample pool is the entire data sample vectors. This type of scenario is generally to use IBSI(S) to perform background suppression such as RX detector by Reed and Yu (1990) and subpixel detector such as CEM in Chang (2003a) where the S consists of entire data sample vectors.
2. The size of S, $N = |S|$, is relative small:
   This scenario generally makes use of IBSI(S) to perform subtle target detection as demonstrated in Chapter 17, to find targets of interest such as anomalies, endmembers, man-made objects, particularly, those targets which are generally scattered and may not be clustered together as a group or a region.
3. The size of S, $N = |S|$, is one, that is, $N = 1$:
   The third scenario is the case that no sample statistics is available, that is, S is a singleton made up of the data sample vector $\mathbf{r}$ itself only. Under such a circumstance, the only available data information is the intrasample spectral correlation IBSI($\mathbf{r}$) that can be used for data analysis. As a result, processing the data sample vector $\mathbf{r}$ becomes a one-dimensional (1D) signal processing of $\mathbf{r}$ along wavelengths, which is exactly the topic to be covered in the second category of this book, Category B: *Hyperspectral Signal Processing.*

Assume that a given data sample vector is a signature vector without reference to others (i.e., $N = 1$). What the best we can do is to explore as much spectral information across the entire wavelength range as possible to specify the data sample vector for spectral characterization. For this purpose, we can consider a general case that a spectral signature $s(\lambda)$ is a 1D continuous-wavelength real-valued signal defined on a wavelength range of $[a,b]$. For $s(\lambda)$ to be implemented in discrete signal processing, $s(\lambda)$ must be sampled according to a sampling interval, $\Delta\lambda$. For example, for a HYDICE data sample vector, there are 210 spectral channels over [0.4 μm, 2.5 μm] with spectral resolution 10 nm, in which case, the sampling interval is $\Delta\lambda = 10$ nm where s($l\Delta\lambda$) where $s(l\Delta\lambda) = s_l$ can be considered as the spectral value of the lth spectral channel to form a spectral signature vector given by $\mathbf{s} = (s_1, s_2, \ldots, s_L)^T$ where L is the total number of samples corresponding to the total number of spectral channels. So, $\Delta\lambda$ actually determines the spectral resolution. When the interval is relatively small, the spectral bands can be considered as contiguous spectral channels and the data sample vector is a hyperspectral signature. Otherwise, the spectral bands are considered as discrete spectral channels, and the data sample vector is a multispectral signature.

Once a spectral signature $\mathbf{s} = (s_1, s_2, \ldots, s_L)^T$ is acquired; there are two ways to process the spectral value s_l of the lth spectral channel in either a discrete manner, referred to as signal coding or in a continuous manner, referred to as signal estimation.

33.7.1 Signal Coding

Signal coding represents a signal by a finite number of discrete values, to be called code words. For example, a real value can be encoded as a binary representation used by computers as bits. In communications, coding is carried out by a quantizer with quantization levels specified by a desired set of discrete values. So, the simplest encoder or quantizer is sign detector, known as hard-limiter, which only detects the sign change of a signal or delta modulator with only two discrete values. In hyperspectral signal processing, the signals are represented by real-valued spectral values $\{s_l\}_{l=1}^{L}$ across the wavelength range $\{l\Delta\}_{l=1}^{L}$ and IBSI(**s**) is spectral correlation provided by L signals, $\{s_l\}_{l=1}^{L}$ within the signature vector $\mathbf{s} = (s_1, s_2, \ldots, s_L)^T$. One earliest attempt to perform signal coding on $\{s_l\}_{l=1}^{L}$ was the spectral analysis manager (SPAM) by Mazer et al. (1988), which is actually implemented as two sign detectors to detect a sign change between the current spectral signal s_l and spectral mean and a sign change in spectral signals between the two adjacent bands, $s_{l\,-\,1}$ and $s_{l\,+\,1}$, of the current being processed band, s_l. SPAM was later improved by spectral feature-based binary coding (SFBC) developed by Qian et al. (1996) by adding a third sign detector, which detects a sign change in spectral deviation from the spectral mean by a prescribed threshold. Since each sign detector requires one bit to dictate the sign change, SPAM and SFBC can be viewed as 2-bit encoder/qunatizer or 3-bit encoder/quantizer or sign detector, respectively. The ideas of SPAM and SFBC are further explored in Chapter 24 to derive many variations from information theory perspectives such as median partition (MP) binary coding, halfway partition (HP) binary coding, and equal probability partition (EPP) binary coding. While these binary coding methods may be effective on many occasions, they may not be so if the signature vector $\mathbf{s} = (s_1, s_2, \ldots, s_L)^T$ has a sophisticated and complex spectrum across L spectral signals $\{s_l\}_{l=1}^{L}$. This is because binary coding only uses up to 2-bit memory to store spectral changes of two adjacent bands and may fail to capture and characterize their spectral behaviors. Nevertheless, these coding methods are still considered as memoryless coding since they are implemented as 2-bit or 3-bit quantizers as scalar quantizers (Gersho and Gray, 1992). In order to resolve this dilemma, a concept of vector coding similar to vector quantization (Gersho and Gray, 1992) is introduced in Chapter 25. One such vector coding is developed based on texture analysis, called spectral derivative feature coding (SDFC) where the sign detector used by binary coding is replaced with a spectral texture descriptor. Another is derived from arithmetic coding, called spectral feature probabilistic coding (SFPC) which can keep track of all spectral changes across the entire set of $\{s_l\}_{l=1}^{L}$. However, implementing vector coding generally requires high computational complexity. Chapter 26 further develops a progressive coding, called multiple-stage PCM (MPCM)-based progressive spectral signature coding (MPCM-PSSC), which takes advantages of simplicity of scalar coding in implementation as well as capability of vector coding in capturing spectral changes in $\{s_l\}_{l=1}^{L}$.

33.7.2 Signal Estimation

The basic goal of the above-mentioned signal coding is designed to generate a credible fingerprint of each of spectral signals, $\{s_l\}_{l=1}^{L}$ so that these fingerprints provide sufficient information for their own identities. But such signal coding does not necessarily tell you what the real signal is. In other words, a signal identity and its fingerprint is one-to-one correspondence such as one-to-one identification between a person and his unique nickname, known as code words. Using a more specific example for illustration, let $\{s_l\}_{l=1}^{L}$ be used for data transmission. When one of these signals is selected for transmission, it is its subscript instead of the signal itself being transmitted. According to information theory, if a fixed length coding is used, only $\log L$ bits required to derive a set of L code words corresponding to fingerprints of the L signals, $\{s_l\}_{l=1}^{L}$ for signal transmission without actually transmitting the signal itself. This is because a signal can be identified by its subscript through its code word used as its fingerprint. Such processing performed by signal coding is indeed 1D discrete-value signal processing and can be only used for hard-decision-made applications such as signal detection, discrimination, classification, identification, but certainly cannot be used for continuous value (real-valued) signal processing-based applications, which require soft decisions such as signal quantification, in which case signal coding does not suffice to characterize $\{s_l\}_{l=1}^{L}$ provided by a signature vector. In order to address this issue, the encoder used by signal coding must be replaced by an operator that can reconstruct the signal to be processed for signal recovery. This is particularly important and critical in applications in chemical/biological agent detection where the agent concentration is crucial to determine various levels of threat and fatality. In this case, each of $\{s_l\}_{l=1}^{L}$ must be approximated by its estimate $\hat{s}_l$ rather than its code word. PART VII in this book is particularly developed for this purpose where three approaches, variable number variable band selection for hyperspectral signal charcaterization in Chapter 27, Kalman filtering-based estimation for hyperspectral signals in Chapter 28, and wavelet representation for hyperspectral signals in Chapter 29, are investigated.

33.8 Applications

Despite that only a few applications are presented in Category C in this book many other applications such as agriculture food quality and safety inspection, homeland security, chemical/biological sensing, and medical imaging, have emerged as major players of hyperspectral imaging sensors in various *SPIE* conferences, *International Symposium on Spectral Sensing Research* (ISSSR) and IEEE symposia, conferences, and workshops. The applications presented in this book are only of particular interest to the author's preference. The two applications considered in Chapter 30 are subpixel target size estimation and concealed target detection, both of which pose difficulties for conventional spatial domain-based techniques since targets of interest in these applications cannot be visualized by inspection and can only be resolved by their spectral not spatial information. As noted in Sections 1.2 and 1.3, HyperSpectral imaging (HSI) is not a direct extension to Multi-Spectral imaging (MSI). As a result, techniques developed for hyperspectral imaging are not necessarily applicable to multispectral imagery. To make hyperspectral imaging techniques also effective in multispectral image analysis, Chapter 31 explores differences between HSI and MSI and further develops two approaches to nonlinear dimensionality extension, band dimensionality expansion (BDE), and kernelization to expand the ability of HSI in dealing with multispectral imagery. Chapter 32 presents one of promising applications to make HSI techniques also work for multispectral imagery where the techniques developed in Chapter 31 are directly applied to magnetic resonance images, which can be considered as multispectral images.

33.9 Further Topics

It is nearly impossible to write a book to cover all interesting topics as it was originally planned. This is certainly the case for this book. In particular, real-time hyperspectral image processing has become an emerging and booming area in recent years, and many research efforts have been reported and published in the literature on a fast pace track. The following sections only provide a preview of this topic. More details can be found in Chang (2013).

33.9.1 Causal Processing

Causal processing is a prerequisite to real-time processing. It only uses the data samples that were already visited in the past but not those in the future for data processing. With this content, spatial domain-based literal techniques in traditional image processing are generally not causal because they are primarily developed to take advantage of spatial correlation among data samples. For example, texture-based and window-based image processing techniques are usually not applicable to real-time processing. This is also true for most anomaly detection techniques such as the commonly used anomaly detector, RX detector (Reed and Yu, 1990), and CEM detector (Harsanyi, 1993) which are not real-time detectors because they require the entire dataset to compute a global covariance or correlation matrix prior to detection. By contrast, the pixel-based nonliteral techniques developed for hyperspectral imagery are actually causal processing techniques due to the fact that no interpixel spatial correlation is involved in which case, the processing can be carried out pixel-by-pixel on a single pixel basis. A good representative of this type of processing is least-squares-based LSMA techniques, LSOSP, NCLS, and FCLS method, all of which can be implemented in real time since they process data pixel by pixel without using any interpixel spatial correlation.

33.9.2 Real-Time Processing

The importance of real-time processing has been recently realized and recognized in many applications. In some applications, for example, on-board spacecraft data processing system, it is very useful to have high levels of processing throughput. Specially, as data rate generated by spacecraft instruments is increasing at speed, on-board data processing has been largely absent from remote sensing missions. Many advantages can be benefited from real-time processing. One is detection of moving targets. This is critical and crucial in battlefield when moving targets such as tanks or missile launching vehicles pose real threat to ground troops. The real-time data processing provides timely intelligence information, which can help to reduce casualty. Another is on-board data processing. For space-borne satellites, real-time data processing can significantly reduce mass storage of data volume. A third advantage is chip design. Since it can be implemented in real time the computational load can be largely reduced. Furthermore, it can also provide a payload relief in aircrafts and satellites. Over the past years, many subpixel detection and mixed pixel algorithms have been developed and shown to be very versatile. However, their applicability to real-time processing problems is generally restricted by the very complex computational workloads.

One key issue in designing a real-time processing algorithm is that real-time processing must be performed in such a fashion that the output should be produced immediately at the same time as an input comes in. As a matter of fact, no such a real-time processing algorithm exists in real world applications since there is always a time lag resulting from data processing. However, form a practical appoint of view, such a time delay is determined by a specific application. For example, in surveillance and reconnaissance applications, finding moving targets is imminent for decision making and the responding time must be very short. In this case, very little time delay should be allowed. As another example, for applications in fire damage management/assessment, the time to respond can be minutes or hours in which case the allowable time delay can be longer. So, an algorithm that can meet such a time constraint can be considered as a real-time processing algorithm. Another key issue is that a real-time processing algorithm must be carried out *causally* in the sense that the data samples used for processing are only those up to the data sample vector currently being processed (Gelb, 1974). More specifically, no future data sample vectors after the current data sample vector can be allowed to be used for data processing. In many applications, the concept of such causality is not considered as a prerequisite to real-time processing as long as an algorithm can be fast processed with negligible time. In this case, it is generally considered as a real-time processing algorithm. Nevertheless, technically speaking, such an algorithm is neither a real-time processing algorithm nor a causal processing algorithm.

33.9.3 FPGA Designs for Hardware Implementation

Over the last few years, several research efforts have been directed toward the incorporation of specialized hardware for accelerating remote sensing-related calculations aboard airborne and satellite sensor platforms. Enabling on-board data processing introduces many advantages, such as the possibility to reduce the data downlink bandwidth requirements at the sensor by both preprocessing data and selecting data to be transmitted based upon predetermined content-based criteria. On-board processing also reduces the cost and the complexity of ground processing systems so that they can be affordable to a larger community. Other remote-sensing applications that will soon greatly benefit from onboard processing are future sensor missions as well as future Mars and planetary exploration missions, for which on-board processing would enable autonomous decisions to be taken on board.

In recent years, rapid advances in VLSI technology have large impact on modern digital signal processing. Over the past years, we have witnessed that the number of transistors per chip has doubled about once a year. Therefore, VLSI design of complex algorithms becomes more and more feasible. A major difficulty with implementing these algorithms in real time is computation of the inverse of a matrix. Systolic arrays provide a possibility. But it requires a series of Givens rotations to decompose a matrix into triangular matrices that can be implemented in real time. Unfortunately, such Givens rotations cannot be realized in hardware. To resolve this issue, the Givens rotations must be performed by operations such as adds, ORs, XORs, shifts that can be realized in hardware architecture. In order to do so, the COordinate Rotation DIgital Computer (CORDIC) algorithm is developed by Volder (1959), which allows us to convert a Givens rotation to a series of shifts-adds operations. Using systolic arrays architecture in conjunction with the CORDIC algorithm, we can implement a matrix inverse computation in a set of shifts-adds operations. As a result, it makes field programmable gate arrays (FPGA) design feasible to become a versatile technique for a wide range of applications in hyperspectral data exploitation because of its hardware advantages for fast data processing. The insights into FPGA design with real-time implementation have also been investigated in recent years. However, it also requires specific applications to realize its benefits such as application-specific integrated circuit (ASIC). One is to make use of orthogonal subspace projection (OSP) to design various hyperspectral imaging algorithms for real-time implementation. Its FPGA design was investigated in Wang (2003) and Chang and Wang (2008) where a QR-decomposition (QRD)-based matrix triangularization approach along with the well-known Feddeeva algorithm is used to calculate the pseudoinverse, $\mathbf{U}^{\#} = \left(\mathbf{U}^{T}\mathbf{U}\right)^{-1}\mathbf{U}^{T}$ for the OSP detector and a CORDIC, key to the QR-decomposition in FPGA, with systolic array architecture was also proposed. This FPGA design architecture has been successfully simulated with real hyperspectral imagery.

Another application is to design real-time implementation of CEM which has shown promise in hyperspectral data exploitation where its systolic array implementation was developed in Chang and Wang (2007). Unlike the OSP, CEM involves the computation of the inverse of the sample spectral correlation matrix instead of computation of the pseudo-inverse of a matrix, $\mathbf{U}^{\#} = \left(\mathbf{U}^{T}\mathbf{U}\right)^{-1}\mathbf{U}^{T}$ as does OSP. In this case, the CORDIC algorithm is readily applicable and there is no need of using the Faddeeva algorithm for conversion. Depending upon how to compute the input stream, two methods are of interest. Method 1 computes the input stream from image pixel vectors directly, while Method 2 computes the sample spectral correlation matrix $\mathbf{R}$. These two module-based method were proposed in Chang and Wang (2007). For Method 1, five modules are required. The first module is to design an array of CORDIC circuits where the pixel stream is fed into the module and the upper triangular matrix $\mathbf{R}_{t}^{\text{upper}}$ is updated in real time. This is followed by the second module which applies backsubstitution to compute the inverse of $\mathbf{R}_t^{\text{upper}}$, inv$\mathbf{R}$. Then Module 3 uses a distributed arithmetic to calculate $\mathbf{c} = \left[(\mathbf{R}_t^{\text{upper}})^T\right]^{-1}\mathbf{d} = \text{inv}\mathbf{R}^{T*}\mathbf{d}$ where the $\mathbf{d}$ is the desired target signature. Next, Module 4 is developed to obtain the desired filter vector $\mathbf{w}$ by finding $\mathbf{w} = \text{inv}\mathbf{R}^{*}\mathbf{c}$. Finally, Module 5 is to produce the results by applying an FIR filter to the current input pixel streams. Method 2 takes an alternative approach by first computing the auto-correlation matrix $\mathbf{R}$. Four modules are proposed for this method. Module 1 is the design of auto-correlator that calculates the sample correlation matrix $\mathbf{R}$. It is then followed by Module 2, which uses the CORDIC circuits to triangularize $[\mathbf{R}|\mathbf{d}]$. Next, Module 3 applies the backsubstitution to obtain the desired filter vector $\mathbf{w}$. Finally, Module 4 produces the filter output energy $\delta^{\text{CEM}}(\mathbf{r}) = \mathbf{w}^T\mathbf{r}$ for target detection by applying an FIR filter to the current input pixel streams. Details of FPGA implementations of OSP and CEM can be found in Wang (2003), Chang and Wang (2007), Chang and Wang (2008), and Chang (2013).

33.9.4 Parallel Processing

Parallel processing is sometimes confused with real-time processing because it can also be implemented in real time. As a matter of fact, these two processes are completely different. The need of real-time processing arises from requirement of processing huge volumes of data and large data storage resulting from data communication. The real-time processing provides a great advantage of *process-then-forget* benefit in the sense that the processed data are immediately output and will not be stored during the entire process. All pixel-based LSMA techniques can be generally implemented to unmix data in real time. The need of parallel processing also arises from the demand of processing enormous volumes provided by hyperspectral data. Parallel processing implements a *divide-and-conquer* strategy in dividing the data to be processed into a number of small data subsets so that all the data subsets can be processed in parallel so as to increase processing speed. So, if a real-time processing algorithm does not have a parallel structure, it cannot be implemented as a parallel processing algorithm. On the other hand, a parallel processing algorithm does not require to be implemented in real time if it is not causal. While these two processes are not correlated with each other, it is certainly desirable to have both in algorithm design by taking advantage of their strengths to improve better data processing performance. As for designing and developing real-time hyperspectral imaging algorithms, readers can consult the book by Chang (2013). With regard to high performance computing via parallel processing, Plaza and Chang (2007a) provides a good reference for those who are interested in this topic.

33.9.5 Progressive Hyperspectral Processing

Progressive hyperspectral data processing is a new concept that has not been much explored in the past. Despite the fact that several chapters, Chapter 20–21, 23 and 26 in this book are devoted to this particular subject it seems still in its infancy in many applications, specifically, satellite data communication and transmission which must handle enormous data in an effective and efficient manner. A progressive process decomposes hyperspectral data processing into a number of sequential stages and processes data stage-by–stage progressively in the sense that the results obtained by previous stages are retained and also used to improve subsequent stages where only the new innovations information that is not available in previous stages is used for data updating. This is quite different from a sequential processing which processes data samples in a sequential manner where each data sample is fully processed not gradually processed. In addition to those discussed in this book several new developments for progressive processing have been recently explored and investigated. Two major trends are of interest. One is progressive band processing in various applications, for example, anomaly detection, endmemeber extraction, and linear spectral unmixing. It is different from progressive band/spectral dimensionality processing in Chapters 20–21, progressive band selection in Chapter 23 and progressive signature coding in Chapter 26 in the sense that progressive band processing is developed for operators particularly designed for specific applications where the operators work like a Kalman filter which can be updated by a recursive equation and innovations information. Another trend is real-time causal processing where data sample vectors are progressively processed in real time as well as causality in the sense that operators particularly designed for specific applications are updated by results obtained by those data sample vectors already visited and the information provided by the current being processed data sample vector via an iterative equation. These two trends pave the way of developing a new theory for progressive hyperspectral processing which is one of main themes in Chang (2013).

Glossary

2D ROC	two-dimensional receiver operating characteristic, Chapter 3
3D ROC	three-dimensional receiver operating characteristic, Chapter 3
AC-FLSMA	abundance-constrained FLSMA, Chapter 13
ACE	adaptive coherence estimation, Chapter 16
AC-LSMA	abundance-constrained linear spectral mixture analysis, Chapters 14, 32
ACLS-FLDA	abundance-constrained least squares FLDA, Chapters 13, 14
AMD	adaptive matched detector, Chapter 2
ANC	abundance non negativity constraint, Chapter 14
ASC	abundance sum-to-one constraint, Chapter 14
ASD	adaptive subspace detector, Chapter 2
AVIRIS	airborne visible/infrared imaging spectrometer, Chapter 1
BD	band de correlation, Chapter 23
BBOPC	between band orthogonal projection criterion, Chapter 30
BDE	band dimensionality expansion, Chapter 31
BEP	band generation process, Chapter 31
BEP-CEM	Chapter 31
BEP-KLSMA	Chapter 31
BEP-LSMA	Chapter 31
BEP-OSP	generalized OSP, Chapter 31
BP	band prioritization, Chapters 21, 23
BS	band selection
C-SFPC	circular-spectral feature probabilistic coding, Chapter 25
CA	component analysis, Chapters 6, 17
CADCA	computer-aided detection and classification algorithm, Chapter 30
CA-ULSMA	component analysis-based unsupervised LSMA, Chapter 17
CBS	constrained band selection, Chapters 6, 23
CCA	convex cone analysis, Chapter 7
CEM	constrained energy minimization, Chapters 2, 12
CMD	covariance-based Mahalanobis distance, Chapter 16
CMFD	covariance-based matched filter distance, Chapter 16
DDA	dynamic dimensionality allocation, Chapter 22

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

DP	dimensionality prioritization, Chapter 20
DR	dimensionality reduction, Chapters 6, 19, 20
DRBS	dimensionality reduction by band selection, Chapters 19, 21, 23
DRT	dimensionality reduction by transform, Chapters 6, 19, 20
DWT	discrete wavelet transform, Chapter 29
EHDC	exploitation-based hyperspectral data compression, Chapter 19
EEA	endmember extraction algorithm, Chapters 7–11
EIA	endmember initialization algorithm, Chapter 9
FDE	feature dimensionality expansion, Chapter 31
FLSMA	Fisher's linear spectral mixture analysis, Chapter 13
FCLS	fully constrained least squares method, Chapters 2, 14, 15
FPGA	field programming gate arrays, Chapter 33
FVC-FLSMA	feature vector constrained FLSMA, Chapter 13
GLRT	generalized likelihood ratio test, Chapter 2
GML	Gaussian maximum likelihood, Chapters 2, 12, 16
HFC	Harsanyi–Farrand–Chang, Chapters 5, 33
HOS	high-order statistics, Chapters 6, 8, 17, 33
HSI	hyperspectral imaging
HYDICE	hyperspectral digital imagery collection experiment, Chapter 1
IBSI	interband spectral information, Chapter 17
IC	independent component, Chapters 6, 19, 20, 32
ICA	independent component analysis, Chapters 6, 19, 20, 32
ID	intrinsic dimensionality, Chapter 5
IDICA-DR	initialization-driven ICA-DR, Chapters 6, 8–11, 32, 33
ID-EEA	initialization-driven EEA, Chapter 9
ID-PIPP	initialization-driven projection index-based prioritized projection pursuit, Chapters 6, 20
IED-EEA	initial endmember-driven EEA, Chapter 9
IN-FINDR	iterative N-FINDR, Chapters 7–11
KFCLS	kernel-based FCLS, Chapters 15, 31, 32
KNCLS	kernel-based NCLS, Chapters 15, 31, 32
KFSCSP	Kalman filter-based spectral characterization signal processing, Chapter 28
KFSSE	Kalman filter-based spectral signature estimation, Chapter 28
KFSSD	Kalman filter-based spectral signature discriminator, Chapter 28
KFSSI	Kalman filter-based spectral signature identifier, Chapter 28
KFSSQ	Kalman filter-based spectral signature quantifier, Chapter 28
KCEM	kernel-based CEM, Chapter 31
KLSMA	kernel-based LSMA, Chapters 15, 31, 32
KLSOSP	kernel-based LSOSP, Chapters 15, 31, 32
KOSP	kernel-based OSP, Chapters 15, 31
LCDA	linearly constrained discriminant analysis, Chapter 12
LCMV	linearly constrained minimum variance, Chapters 2, 12
LCVF	lunar crater volcanic field, Chapter 1
LRT	likelihood ratio test, Chapter 2
LS-LSMA	least squares-based linear spectral mixture analysis, Chapters 12–15, 17
LSE	least squares error, Chapters 2, 12
LSMA	linear spectral mixture analysis, Chapters 2, 12–15, 17
LSOSP	least squares OSP, Chapters 12–15

LS-ULSMA	least squares-based unsupervised LSMA, Chapter 17
LSU	linear spectral unmixing
MD	Mahalanobis distance, Chapter 16
M-PIPP	mixed projection index-based prioritized projection pursuit, Chapters 6, 20
MLC	maximum likelihood classifier, Chapters 12, 16
MNF	maximum noise fraction, Chapter 6
MPCM	multistage PCM, Chapter 26
MRI	magnetic resonance imaging, Chapter 32
MSI	multispectral imaging, Chapter 31
NAPC	noise adjusted principal component, Chapter 6
NCLS	non negativity constrained least squares method, Chapters 2, 13–15
N-FINDR	N-finder algorithm, Chapters 7–11
NWHFC	noise-whitened Harsanyi–Farrand–Chang, Chapters 5, 33
OC-ICA	over-complete ICA, Chapters 31, 32
OC-LSU	over-complete linear spectral unmixing, Chapter 31
OSP	orthogonal subspace projection, Chapters 2, 12–15, 17, 27, 31, 32
PBDE	progressive band dimensionality expansion, Chapter 21
PBDP	progressive band dimensionality process, Chapter 21
PBDR	progressive band dimensionality reduction, Chapter 21
PBS	progressive band selection, Chapter 23
PC	principal component, Chapters 6, 20, 31
PCA	principal components analysis, Chapters 6, 20, 31
PIC	projection index components, Chapters 6, 20
PIPP	projection index-based projection pursuit, Chapters 6, 20
PI-PRPP	projection index-based prioritized projection pursuit, Chapters 6, 20
PP	projection pursuit, Chapters 6, 20
PPI	pixel purity index, Chapters 7–11
PSC	progressive signature coding, Chapter 26
PSDE	progressive spectral dimensionality expansion, Chapter 20
PSDP	progressive spectral dimensionality process, Chapter 20
PSDR	progressive spectral dimensionality reduction, Chapter 20
PSSC	progressive spectral signature coding, Chapter 26
PVE	partial volume estimation, Chapter 32
REEA	random EEA, Chapter 10
RICA-DR	random ICA-DR, Chapter 10
RPPI	random pixel purity index, Chapter 10
RMD	correlation-based Mahalanobis distance, Chapter 16
RMFD	correlation-based matched filter distance, Chapter 16
ROC	receive operating characteristic, Chapter 3
RXD	RX detector, Chapters 2, 16, 33
S-SFPC	split-spectral feature probabilistic coding, Chapter 25
SAM	spectral angle mapper, Chapters 16, 24
SC-N-FINDR	successive N-FINDR, Chapters 8–11
SC-PCA	successive PCA, Chapter 8
SDFC	spectral derivative feature coding, Chapter 25
SFBC	spectral feature-based binary coding, Chapters 24, 25
SFPC	spectral feature probabilistic coding, Chapter 25
SGA	simplex growing algorithm, Chapters 8–11

SID	spectral information divergence, Chapter 16
SM-EEA	simultaneous EEA, Chapter 7
SM-N-FINDR	simultaneous N-FINDR, Chapter 7
SPAM	spectral analysis manager, Chapters 24, 25
SPICA-DR	statistics prioritized ICA-DR, Chapter 6
SQ-EEA	sequential EEA, Chapter 8
SQ-N-FINDR	sequential N-FINDR, Chapter 8
SSE	signal subspace estimate, Chapter 5
SM-PCA	simultaneous PCA, Chapter 20
SSC	spectral signature coding, Chapters 24–26
SVD	singular value decomposition, Chapter 6
TCIMF	target-constrained interference-minimized filter, Chapters 2, 12
TE	target embededness, Chapter 4
TI	target implantation, Chapter 4
UC-ICA	under-complete ICA, Chapter 32
UC-LSMA	under-complete linear spectral mixture analysis, Chapter 32
UFCLS	unsupervised fully constrained least squares, Chapters 8, 17
UNCLS	unsupervised non negativity constrained least squares, Chapters 8, 17
UTD	uniform target detector, Chapter 16
UTSFA	unsupervised training sample finding algorithm, Chapter 17
VCA	vertex component analysis, Chapters 8, 10, 11
VD	virtual dimensionality, Chapter 5
VS	Virtual signature, Chapter 17
WAC-LSMA	weighted abundance-constrained linear spectral mixture analysis, Chapter 14
WSCA	wavelet-based signature characterization algorithm, Chapter 29
WSCA-SSC	wavelet-based signature characterization algorithms for signature self-correction, Chapter 29
WSCA-SST	wavelet-based signature characterization algorithms for signature self-tuning, Chapter 29

Appendix

Algorithm Compendium

This appendix compiles many algorithms described in this book, most of which have been developed in the Remote Sensing Signal and Image Processing Laboratory (RSSIPL) at the University of Maryland, Baltimore County. In order to help readers implement these important algorithms, their MATLAB codes are also included for reference so that readers can write their own program codes without relying on software packages such as ENVI, ERDAS, etc. Each algorithm is described according to its functionality and its categorization and is then followed by its MATLAB codes.

A.1 Estimation of Virtual Dimensionality

The concept of virtual dimensionality (VD) was first coined in the book by Chang (2003a) and later published by Chang and Du (2004). It is defined as the number of spectrally distinct signatures present in the data and has received considerable interest in unsupervised hyperspectral data exploitation since it was introduced in 2003. Many approaches have been developed for estimating the value of VD. Nevertheless, the most popular algorithm to be used for this purpose is the one developed by Harsanyi et al. (1994a), whose idea was the origin of VD. All the details of VD along with techniques developed to estimate VD can be found in Chapter 5.

A.1.1 Harsanyi–Farrand–Chang Method

- **Algorithm name:** Harsanyi–Farrand–Chang (HFC) method
- **Authors:** J. C. Harsanyi and Chein-I Chang
- **Category:** preprocessing
- **Designed criteria:** eigenvlaues of sample correlation/covariance matrix
- **Designed method:** Neyman–Pearson detection theory
- **Typical use (LOI's addressed):** estimation of number of spectral distinct signatures
- **Inputs:** reflectance or radiance cube
- **Outputs:** a positive integer and false alarm probability
- **Assumptions:** no prior knowledge required
- **Sensitivity to LOI (target knowledge):** moderate
- **Sensitivity to noise:** moderate
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

- **Effectiveness:** high
- **Implementation:** simple, easy to use, but not real time
- **Brief description:**
  The HFC method was first envisioned in Harsanyi et al. (1994a) to detect spectral signatures present in AVIRIS data. It was then used to find the now-popular terminology, VD (Chapter 17 in Chang (2003a) and Chapter 5 in this book), which is defined as the number of spectral distinct signatures and later published in Chang and Du (2004). It calculates the difference between eigenvlaues in sample correlation matrix and sample covariance matrix and makes use of Neyman–Pearson detector to determine the value of VD.

MATLAB Codes of HFC Method

```
function number=HFC(HIM,t);
%
% HFC gives the VD number estimated by given false alarm property using HFC
% method.
%
% There are two parameters,HFC(HIM,t) where the HIM is the
% Hyperspectral image cube, which is a 3-D data matrix
% [XX,YY,bnd] = size(HIM), XX YY are the image size,
% bnd is the band number of the image cube.
% t is the false alarm probability.
%
% HFC uses the HFC algorithm developed by Dr. Chein-I Chang,
% see http://www.umbc.edu/rssipl/. The Matlab code was
% programmed by Jing Wang in Remote Sensing Signal and
% Image Processing Lab.
%

[XX,YY,bnd] = size(HIM);
pxl_no = XX*YY;
r = (reshape(HIM,pxl_no,bnd))';

R = (r*r')/pxl_no;
u = mean(r,2);
K = R-u*u';

%======HFC=====
D1=sort(eig(R));
D2=sort(eig(K));
sita=sqrt((D1.^2+D2.^2)*2/pxl_no);
P_fa=t;
Threshold=(sqrt(2))*sita*erfinv(1-2*P_fa);

Result=0;
for m=1:bnd
  if ((D1(m,1)-D2(m,1))>Threshold(m,1))
    Result(m,1)=1;
  else
    Result(m,1)=0;
  end
end
fprintf(’The VD number estimated is’);
disp(sum(Result));
number=sum(Result);
```

A.1.2 Noise-Whitened Harsany–Farrand–Chang (NWHFC) Method

- **Algorithm name:** noise-whitened Harsanyi–Farrand–Chang (HFC) method.
- **Authors:** Chein-I Chang
- **Category:** preprocessing
- **Designed criteria:** eigenvlaues of sample correlation/covariance matrix
- **Designed method:** Neyman–Pearson detection theory
- **Typical use (LOI’s addressed):** estimation of number of spectral distinct signatures
- **Inputs:** reflectance or radiance cube
- **Outputs:** a positive integer and false alarm probability
- **Assumptions:** no prior knowledge required
- **Sensitivity to LOI (target knowledge):** moderate
- **Sensitivity to noise:** low
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, but not real time
- **Brief description:**
  The NWHFC method is a noise-whitened version of the HFC method and was developed in Chang (2003a) and Chang and Du (2004). It requires a reliable method to estimate the noise covariance matrix.

MATLAB Codes of the NWHFC Method

```
function number=NWHFC(HIM,t)
%
% NWHFC gives the VD number estimated by given false alarm property using
% NWHFC method.
%
% There are two parameters, NWHFC(HIM,t) where the HIM is the
% Hyperspectral image cube, which is a 3-D data matrix
% [XX,YY,bnd] = size(HIM), XX YY are the image size,
% bnd is the band number of the image cube.
% t is the false alarm probability.
%
% HFC uses the NWHFC algorithm developed by Dr. Chein-I Chang,
% see http://www.umbc.edu/rssipl/. The Matlab code was
% programmed by Jing Wang in Remote Sensing Signal and
% Image Processing Lab.
%
```

```
[XX,YY,bnd] = size(HIM);
pxl_no = XX*YY;
r = (reshape(HIM,pxl_no,bnd))’;

R = (r*r’)/pxl_no;
u = mean(r,2);
K = R-u*u’;

%======Noise estimation=====
K_Inverse=inv(K);
tuta=diag(K_Inverse);
K_noise=1./tuta;
K_noise=diag(K_noise);

%=====Noise whitening===
y=inv(sqrtm(K_noise))*r;
y=reshape(y’,XX,YY,bnd);
%=====Call HFC to estimate===
number=HFC(y,t);
```

A.2 Data Sphering

The purpose of data sphering is to allow users to analyze data structure characterized by high-order statistics. Before doing so, the data samples characterized by the first two orders of statistics, that is, mean and variances/covariances must be removed. The data sphering is designed to accomplish this task. It first removes the data sample mean by setting data set centered at the origin and then de-correlates data samples by zeroing all covariances via diagonalization of data sample covariance matrix. Finally, it normalizes data sample variances to 1 by placing all de-correlated data samples on the unit sphere. So, by means of matrix diagonalization and variance normalization, all data samples characterized by high-order statistics are either inside the sphere characterized by sub-Gaussian or outside the sphere characterized by super-Gaussian. Technically speaking, a whitening processing is a part of data sphering that only de-correlates data samples without normalization. However, in statistical signal processing and communications community as well as in many application whitening is indeed data sphering. In this book, we particularly make a distinction between them. In other words, whitening only de-correlates data samples by making co-variances zero but does not normalize variances to 1. It is only a part of data sphering. Details of data sphering can be found in Chapter 6.

- **Algorithm name:** data sphering
- **Category:** preprocessing
- **Designed criteria:** eigenvalues
- **Designed method:** sample covariance matrix
- **Typical use (LOI’s addressed):** preprocessing
- **Inputs:** reflectance or radiance cube
- **Outputs:** gray-scale images
- **Assumptions:** no prior knowledge required
- **Sensitivity to LOI (target knowledge):** high to high-order statistics

- **Sensitivity to noise:** low
- **Operating Bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, but not real time
- **Brief description:**
  Data sphering, also known as a whitening processing in statistical signal processing and communication, is a commonly used method to remove data sample vectors characterized by the first two-order statistics. It is a required preprocessing step prior to ICA (Hyvarinen and Oja, 2001).

MATLAB Codes of Data Sphering

```
function sphered_data=data_sphering(HIM)

% Initial variables
[row column band]=size(HIM);
% Center the data
HIM_row=reshape(HIM,row*column,band);
HIM_zeromean=HIM_row-repmat(mean(HIM_row),row*column,1);
cov=HIM_zeromean’*HIM_zeromean/(row*column);

% Eigen decomposition
[V D]=eig(cov);

% Transform the data set
for i=1:band,
  sphered_data(i,:)=(V(:,i)’*HIM_zeromean’)./(D(i,i)^.5*row);
end

% Transform the data back
sphered_data=reshape(sphered_data’,row,column,band);
```

A.3 Dimensionality Reduction by Transform

Four transformations are used to perform data dimensionality reduction (DR). Two are of second-order statistics-based transformations, data variance-based principal components analysis (PCA), and signal-to-noise ratio (SNR)-based maximum noise fraction (MNF). The other two are of high-order statistics (HOS)-based transformations, independent component analysis (ICA), and high-order statistical moment-based transforms. Details of these four DR transformations can be found in Chapter 6.

A.3.1 PCA

- **Algorithm name:** principal components analysis (PCA)
- **Category:** component analysis-based transform
- **Designed criteria:** eigenvalues
- **Designed method:** sample covariance matrix

- **Typical use (LOI's addressed):** data representation by eigenvectors
- **Inputs:** reflectance or radiance cube
- **Outputs:** gray-scale images
- **Assumptions:** no prior target knowledge required
- **Sensitivity to LOI (target knowledge):** targets of high-order statistics
- **Sensitivity to Noise:** low
- **Operating Bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, but not real time
- **Brief description:**
  PCA (Gonzalez and Woods, 2002) is probably the most widely used component analysis transform that allows users to data in terms of eigenvalues/eigenvectors via eigen-decomposition where data are preserved and retained according to data variances in the descending order.

MATLAB Codes of PCA

```
function [PCs]=PCA(HIM,M);

bnd=size(HIM,3);
xx=size(HIM,1);
yy=size(HIM,2);

x=reshape(HIM,xx*yy,bnd);
x=x';
L=size(x,1);
K=size(x,2);

u=mean(x,2); %dimension of u is 1*L
x_hat=x-u*ones(1,K);
m=mean(x,2);
C=(x*x')/size(x,2)-m*m';
%===========
[V,D]=eig(C);

d=(diag(D))';
[oo,Index]=sort(d);

for m=1:L
  D_sort(1,m)=d(1,Index(1,Lm));
  V_sort(:,m)=V(:,Index(1,Lm));
end

D=diag(D_sort);
V=V_sort;

D=D(1:M,1:M);
V=V(:,1:M);
```

```
%====for the matrix with full column rank, so the
A=V';
x_whitened=A*(x_hat);
PCs=x_whitened;
```

A.3.2 MNF

- **Algorithm name:** maximum noise fraction (MNF)
- **Authors:** A.A. Green, M. Berman, P. Switzer, and M.D. Craig
- **Category:** component analysis-based transform
- **Designed criteria:** signal-to-noise ratio
- **Designed method:** sample covariance matrix
- **Typical use (LOI's addressed):** data representation by SNR
- **Inputs:** reflectance or radiance cube
- **Outputs:** gray-scale images
- **Assumptions:** no prior target knowledge required
- **Sensitivity to LOI (target knowledge):** high-order statistics
- **Sensitivity to Noise:** high
- **Operating Bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, but not real time
- **Brief description:**
  MNF was developed by Green et al. (1988) and further modified by Lee et al. (1990) and referred to as Noise Adjusted Principal Component (NAPC) transform. It represents data in terms of SNR rather than data variances as PCA does.

MATLAB Codes of MNF

```
function [ImageCub_MNF,Matrix_of_Vector]=MNF(ImageCub);
Last_volumn=-10000000000.0000001;
[height,width,NumberOfSpectrum]=size(ImageCub);
ImageCub_MNF=zeros(height,width,NumberOfSpectrum);
%——————————————begin to compute Matrix_of_Vector——————————————%
meanSpect=zeros(NumberOfSpectrum,1);
for II=1:height
  for JJ=1:width
    meanSpect=meanSpect+squeeze(ImageCub(II,JJ,:))/height/width;
  end
end

TotalCovariance=zeros(NumberOfSpectrum,NumberOfSpectrum);

for II=1:height
  for JJ=1:width
    TotalCovariance=TotalCovariance+(squeeze(ImageCub(II,JJ,:))-
meanSpect)*(squeeze(ImageCub(II,JJ,:))-meanSpect)'/height/width;
  end
```

```
end
Matrix_F=zeros(NumberOfSpectrum,NumberOfSpectrum);
Cov_inv=inv(TotalCovariance);
for II=1:NumberOfSpectrum
  Matrix_F(II,II)=sqrt(Cov_inv(II,II));
end
adjusted_Cov=Matrix_F'*TotalCovariance*Matrix_F;

[V,D]=eig(adjusted_Cov);
eig_value=zeros(NumberOfSpectrum,2);
for II=1:NumberOfSpectrum
  eig_value(II,1)=D(II,II);
  eig_value(II,2)=II;
end
%disp(eig_value);
V_sort_min_to_max=sortrows(eig_value,1);
Matrix_of_Vector_before=zeros(NumberOfSpectrum,NumberOfSpectrum);
for II=1:NumberOfSpectrum
   Matrix_of_Vector_before(:,II)=squeeze(V(:,V_sort_min_to_max(NumberOf-
Spectrum-II+1,2)));
end
Matrix_of_Vector=Matrix_F*Matrix_of_Vector_before;
% ——————————————— end of computing Matrix_of_Vector ———————————————— %
for II=1:height
  for JJ=1:width
    r=squeeze(ImageCub(II,JJ,:))-meanSpect;
    ImageCub_MNF(II,JJ,:)=Matrix_of_Vector'*r;
  end
end
```

A.3.3 ICA

- **Algorithm name:** ICA
- **Category:** component analysis-based transform
- **Designed criteria:** infinite-order statistics
- **Designed method:** blind source separation via a linear mixture model
- **Typical use (LOI's addressed):** no prior knowledge is required
- **Inputs:** reflectance or radiance cube
- **Outputs:** gray-scale images
- **Assumptions:** no prior target knowledge required
- **Sensitivity to LOI (target knowledge):** high
- **Sensitivity to noise:** low
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, but not real time
- **Brief description:**

> ICA (Hyvarinen et al., 2001) is widely used in blind source separation via a linear mixture model, which assumes that there exists at most one Gaussian source. A fast algorithm developed by Hyvarinen and Oja (1997), called FastICA, is the most used algorithm to implement ICA.

When ICA is implemented to perform DR, it suffers from a serious issue that the results are not repeatable and inconsistent due to its use of random initial conditions. As a result, the ICA-generated independent components (ICs) generally appear in a random order. In this case, the IC appearing earlier does not necessarily imply that it is more important or significant than that generated later. In order to resolve this issue, three versions of ICA, referred to as ICA-DR1, ICA-DR2, and ICA-DR3, have been proposed by Wang and Chang (2006a, 2006b) and re-named as statistics prioritized ICA-DR (SPICA-DR), random ICA (RICA-DR), and initialization driven ICA-DR (IDICA-DR) in Sections 6.4.1–6.4.3, respectively. In all the three algorithms, an algorithm developed by Hyvarinen and Oja (1997), called FastICA, is modified and implemented in MATLAB codes to realize ICA. To distinguish it from the original FastICA, it is named “My FastICA” and is provided as follows.

MATLAB Codes of My FastICA

```
function [ICs]=My_fastica_v5(HIM,M);

bnd=size(HIM,3);
xx=size(HIM,1);
yy=size(HIM,2);

x=reshape(HIM,xx*yy,bnd);
x=x’;
L=size(x,1);
K=size(x,2);
%====Data sphering =====
u=mean(x,2); %dimension of u is 1*L
x_hat=x-u*ones(1,K);
m=mean(x,2);
C=(x*x’)/size(x,2)-m*m’;
%===========
[V,D]=eig(C);
A=inv(sqrtm(D))*V’;  % A is the whitening matrix....
x_whitened=A*(x_hat);
%=======
clear x;
clear x_hat;
%====rank the eigenvalues, which is used for using the eigenvector as
%initialization
%
% d=(diag(D))’;
% [oo,Index]=sort(d);
%
%
% for m=1:L
%   D_sort(1,m)=d(1,Index(1,L+1-m));
%   V_sort(:,m)=V(:,Index(1,L+1-m));
% end
%
% D=diag(D_sort);
% V=V_sort;
%=====

%====Sphering finished

threshold = 0.0001;
B=zeros(L);
for round=1:M
  fprintf(’IC %d’, round);
  %===initial condition ===
 w=rand(L,1)-0.5;
% w=V(:,round);   Eigenvectors initialization
  %===
  w=w-B*B’*w;
  w=w/norm(w);
  wOld=zeros(size(w));
  wOld2=zeros(size(w));
  i=1;
  while i<=1000
    w=w-B*B’*w;
    w=w/norm(w);
    fprintf(’.’);
    if norm(w-wOld)<threshold | norm(w+wOld)<threshold
      fprintf(’Convergence after %d steps\n’, i);
      B(:,round)=w;
      W(round,:)=w’;
      break;
    end
    wOld2=wOld;
    wOld=w;
    w=(x_whitened*((x_whitened’*w).^3))/K-3*w;
    w=w/norm(w);
    i=i+1;
  end
  if (i>1000)
      fprintf(’Warning! can not converge after 1000 steps \n, no more
components’);
      break;
  end
  round=round+1;
end

ICs=W*x_whitened;
```

```
% figure;
% for m=1:M
% s=reshape(abs(ICs(m,:)),xx,yy); subplot(6,8,m);
% imagesc(s); axis off;colormap(gray);
% end
```

MATLAB Codes of SPICA-DR (ICA-DR1)

```
function [IC_sorted]=sort_IC_DR1(HIM,M);
clear J;

bnd=size(HIM,3);
xx=size(HIM,1);
yy=size(HIM,2);
[ICs]=My_fastica_v5(HIM,M);
ICs=abs(ICs);

% %====Show the ICs in the original order===
% figure;
% for m=1:size(ICs,1)
%   s=reshape(abs(ICs(m,:)),xx,yy);
%   s=255*(s-min(min(s))*ones(size(s,1),size(s,2)))/(max(max(s))
-min(min(s)));
%   temp=mean(reshape(s,xx*yy,1));
%   subplot(6,8,m); imshow(uint8(s));
% %   title(m);
% %
% end

%======Calculate the contrast function
for m=1:size(ICs,1);
  s=ICs(m,:);
  var1=var(s);
  mean1=mean(s);
  sita=sqrt(var1);
  skew_temp=sum((s-mean1).^3)/(xx*yy-1);
  kurt_temp=sum((s-mean1).^4)/(xx*yy-1);

  J(1,m)=(skew_temp.^2)/12+((kurt_temp-3).^2)/48;

end

%======IC sorting ========
[a,b]=sort(J);
b=flipud(b');
IC_sorted=ICs(b',:);

%=========Show the ICS after sorting...
```

```
figure;

for m=1:size(ICs,1)
  s=reshape(abs(IC_sorted(m,:)),xx,yy);
  s=255*(s-min(min(s))*ones(size(s,1),size(s,2)))/(max(max(s))
-min(min(s)));
  temp=mean(reshape(s,xx*yy,1));
  subplot(6,8,m); imshow(uint8(s));
 % title(m);
%
end
```

MATLAB Codes of RICA-DR (ICA-DR2)

```
function [IC_selected]=sort_IC_DR2(HIM,M,run_times);

bnd=size(HIM,3);
xx=size(HIM,1);
yy=size(HIM,2);

fprintf(’first ICA run \n’);
[ICs]=My_fastica_v5(HIM,M*2);
set1=abs(ICs);
set_com=set1;

size1=ones(1);
for round=1:run_times
  fprintf(’ ICA run, order is %d \n’,round+1);
  [ICs]=My_fastica_v5(HIM,2*M);
  set2=abs(ICs);
  set_com_new=[];
  distance=0;
  for m=1:size(set_com,1)
    for n=1:size(set2,1)
      temp1=sqrt(sum(set_com(m,:).^2));  % SAM
      temp2=sqrt(sum(set2(n,:).^2));
      distance(m,n)=acos(sum(set_com(m,:).*set2(n,:))/(temp1*temp2));
    end
    t=distance(m,:)<=0.5;
    if (sum(t)>=1)
       set_com_new=cat(1,set_com_new,set_com(m,:));
    end
  end
  set_com=set_com_new;
  fprintf(’the size of set_Com is’);
  size(set_com_new)
  size1(round,1)=size(set_com_new,1);
  if (size(set_com_new,1)<=M)
```

```
    break;
  end

end
IC_selected=set_com;
```

MATLAB Codes of IDICA-DR (ICA-DR3)

```
function [Loc,Sig]=My_ATGP(HIM,M);

bnd=size(HIM,3);
xx=size(HIM,1);
yy=size(HIM,2);

r=reshape(HIM,xx*yy,bnd);
r=r';
%=====Find the first point

temp=sum(r.*r);
[a,b]=max(temp);

if (rem(b,xx)==0)
  Loc(1,1)=b/xx;
  Loc(1,2)=xx;
elseif (floor(b/xx)==0)
  Loc(1,1)=1;
  Loc(1,2)=b;
else
  Loc(1,1)=floor(b/xx)+1; % y
  Loc(1,2)=b-xx*floor(b/xx); % x
end

Sig(:,1)=r(:,b);
fprintf('1\n');
%==========
for m=2:M

  U=Sig;
  P_U_perl=eye(bnd)-U*inv(U'*U)*U';
  y=P_U_perl*r;
  temp=sum(y.*y);
  [a,b]=max(temp);
  if (rem(b,xx)==0)
    Loc(m,1)=b/xx;
    Loc(m,2)=xx;
  elseif (floor(b/xx)==0)
    Loc(m,1)=1;
    Loc(m,2)=b;
  else
    Loc(m,1)=floor(b/xx)+1; % y
    Loc(m,2)=b-xx*floor(b/xx); % x
  end
  Sig(:,m)=r(:,b);
  disp(m)
end
%
% figure; imagesc(HIM(:,:,30)); colormap(gray); hold on
% axis off
% axis equal
% for m=1:size(Loc,1)
%   plot(Loc(m,1),Loc(m,2),'o','color','g');
%   text(Loc(m,1)+2,Loc(m,2),num2str(m),'color','y','FontSize',12);
% end
%
```

Since IDICA-DR requires an initialization algorithm to generate a specific set of initial condition, the following My FastICA implements the FastICA using ATGP-generated data sample vectors (program is called My_ATGP) as its initial condition

MATLAB Codes of My FastICA for IDICA-DR (ICA-DR3)

```
function [ICs]=My_fastica_DR3(HIM,M);

bnd=size(HIM,3);
xx=size(HIM,1);
yy=size(HIM,2);

x=reshape(HIM,xx*yy,bnd);
x=x';

L=size(x,1);
K=size(x,2);

%====Sphering =====
u=mean(x,2); %dimension of u is 1*L
x_hat=x-u*ones(1,K);
m=mean(x,2);
C=(x*x')/size(x,2)-m*m';
%===========
[V,D]=eig(C);
A=inv(sqrtm(D))*V';  % A is the whitening matrix....
x_whitened=A*(x_hat);
%====for cuprite data===
clear x;
clear x_hat;
%=========for initialization
```

```
[Loc,Sig]=My_ATGP(reshape(x_whitened’,xx,yy,L),M);

W_initial=Sig;

threshold = 0.0001;
B=zeros(L);
%===============find the first point=========

for round=1:M
  fprintf(’IC %d’, round);
  %===Initial condition
  w=W_initial(:,round);
  %===
  w=w-B*B’*w;
  w=w/norm(w);
  wOld=zeros(size(w));
  wOld2=zeros(size(w));
  i=1;
  while i<=1000
    w=w-B*B’*w;
    w=w/norm(w);
    fprintf(’.’);
    if norm(w-wOld)<threshold | norm(w+wOld)<threshold
      fprintf(’Convergence after %d steps\n’, i);
      B(:,round)=w;
      W(round,:)=w’;
      break;
    end
    wOld2=wOld;
    wOld=w;
    w=(x_whitened*((x_whitened’*w).^3))/K-3*w;
    w=w/norm(w);
    i=i+1;
  end
  if (i>1000)
        fprintf(’Warning! cannot converge after 1000 steps \n, no more
components’);
        break;
  end
  round=round+1;
end

ICs=W*x_whitened;

figure
for m=1:M
  s=reshape(abs(ICs(m,:)),xx,yy);
  s=255*(s-min(min(s))*ones(size(s,1),size(s,2)))/(max(max(s))
-min(min(s)));
  temp=mean(reshape(s,xx*yy,1));
  subplot(5,6,m); imshow(uint8(s));
%
end

A=W;

% x_re_ICA=V*sqrtm(D)*(A*ICs)+u*ones(1,xx*yy);
```

A.3.4 HOS-DR

Variance-based PCA and SNR-based MNF represent second-order statistics-based component analysis transformations to perform DR. At the other extreme, ICA is an infinite-order statistics-based component analysis that makes use of mutual information to measure statistical independency among all ICs. However, due to complexity of implementing mutual information, the commonly used FastICA is actually developed by combining the third and fourth moments as a criterion to measure the dependency among its generated components. So, technically speaking, the FastICA-generated components are not really statistically independent. Instead, they can be only considered as high-order statistically dependent. HOS-DR is developed to extend DR transformation with order of statistics higher than 2 (Ren et al., 2006). In this context, the FastICA (Hyvarinen and Ojha, 1997) can be considered as a special case of HOS-DR transformation.

- **Algorithm name:** high-order statistics DR (HOS-DR)
- **Authors:** H. Ren and Chein-I Chang
- **Category:** component analysis-based transform
- **Designed criteria:** statistical moments higher than 2
- **Designed method:** moment projection
- **Typical use (LOI's addressed):** no prior knowledge is required
- **Inputs:** reflectance or radiance cube
- **Outputs:** gray-scale images
- **Assumptions:** no prior target knowledge required
- **Sensitivity to LOI (target knowledge):** high
- **Sensitivity to noise:** low
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, but not real time
- **Brief description:**

  ICA is widely used in blind source separation via a linear mixture model, which assumes that there exists at most one Gaussian source. A fast algorithm developed by Hyvarinen and Oja (1997), called FastICA, is the most used algorithm to implement ICA.

MATLAB Codes of HOS-DR

```
function [ICs]=high_order(HIM,M,k,Initial);
% HIM is the image cube.
% M is the number of components to generated using high order
% k is the order of statistics. For example, k=3, is the skewness, k=4, is
% the kurtosis, k=5 is the 5th moment, and so on...
% Initial condition preference, 0 is the random initial, 1 is the eigen
% initial, 2 is the unity initial. default is 0
if nargin < 3
  fprintf('Please identify the order of statistics!');
  ICs=[];
else

  if nargin < 4
    Initial=0;
  end

  bnd=size(HIM,3);
  xx=size(HIM,1);
  yy=size(HIM,2);

  x=reshape(HIM,xx*yy,bnd);
  x=x';

  L=size(x,1);
  K=size(x,2);

  %===input x is a matrix with size=L*K;
  %====Sphering =====
  u=mean(x,2); %dimension of u is 1*L
  x_hat=x-u*ones(1,K);

  %===jing's code of cov
  m=mean(x,2);
  C=(x*x')/size(x,2)-m*m';
  %===========
  [V,D]=eig(C);
  A=inv(sqrtm(D))*V';  % A is the whitening matrix....
  x_whitened=A*(x_hat);

  clear x;
  clear x_hat;

  % Seperating , using high-order
  threshold = 0.01;
  B=zeros(L);
  y=x_whitened;
  W=ones(1,L);

  P_U_perl=eye(bnd);
  for round=1:M
  fprintf('IC %d', round);
  %===initial condition ===
  switch (Initial)
    case 0
      w=rand(L,1);
    case 1
      w=V(:,round); % Final version
    case 2
      w=ones(L,1);
    otherwise
      w=rand(L,1);
  end

  i=1;
  while i<=100 % maximum times of trying..
    a=(y.*repmat((w'*y).^(k-2),bnd,1))*y'; %skewness
    a=a/K;  % get the sample mean as expectation
    [V,D]=eig(a);
    D=abs(D);
    [C,I]=max(diag(D));
    V1 = V(:,I);
    fprintf('.');
    distance(round,1,i)=norm(w-V1);
    distance(round,2,i)=norm(w+V1);

    if norm(w-V1)<threshold | norm(w+V1)<threshold
      fprintf('Convergence after %d steps\n', i);

      B(:,round)=w;
      W(round,:)=w';
      break;
    end
    w=V1;
    i=i+1;
  end
  %===if not converge. then use the results after 10 iterations
  B(:,round)=w;
  W(round,:)=w';
  %=======
  P_U_perl=eye(bnd)-W'*inv(W*W')*W;
  y=P_U_perl*y;
end

ICs=W*x_whitened;

figure;

for m=1:M
```

```
    s=reshape(abs(ICs(m,:)),xx,yy);
    s=255*(s-min(min(s))*ones(size(s,1),size(s,2)))/(max(max(s))
-min(min(s)));
    temp=mean(reshape(s,xx*yy,1));
    subplot(5,6,m); imshow(uint8(s));
    %
  end

end
```

A.4 Endmember Extraction Algorithms

The MATLAB codes of three major endmember extraction algorithms, pixel purity index (PPI) algorithm with fast iterative PPI (FIPPI), N-finder algorithm (N-FINDR), simplex growing algorithm (SGA) are provided in this section.

A.4.1 Pixel Purity Index

- **Algorithm name:** PPI
- **Authors:** J. Boardman
- **Category:** convex geometry
- **Designed criteria:** orthogonal projection
- **Designed method:** randomly generated vectors, called skewers
- **Typical use (LOI's addressed):** endmember extraction with number of endmembers to be known
- **Inputs:** reflectance or radiance cube
- **Outputs:** endmembers
- **Assumptions:** the number of skewers to be generated, *K* must be sufficiently large
- **Sensitivity to LOI (target knowledge):** high to *K* as well as skewers
- **Sensitivity to noise:** moderate
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, but not real time
- **Brief description:**

  PPI was first developed by Boardman to extract endmembers (Boardman, 1994) and is available in ENVI software. Unfortunately, its detailed steps in implementation are not available for users who would like to make modifications or changes at their discretion. Its MATLAB codes provided in the following serve this purpose. It is developed based on the concept of the convex geometry and the criterion of orthogonal projection. It first generates a set of *K* random unit vectors, called skewers, to cover all possible projection directors and then orthogonally projects all data sample vectors on these skewers to find the maximal and minimal orthogonal projections of each skewer. For each data sample vector, it counts the number of skewers on which its orthogonal projections yield either maximal or minimal projections. This count is referred to as the PPI count, which will be used to determine whether or not a particular data sample vector is an endmember. In doing so, a threshold needs to be specified in advance. In ENVI, this is done manually. In addition, since its skewers are generated randomly, the results are not repeatable. In other words, the same user running PPI in different times or different users running PPI at the same time will all have different results. This serious drawback can further be fixed by initialization-driven PPI (ID-PPI), such as a fast iterative PPI (FIPPI) algorithm (Chang and Plaza, 2006), and random PPI (RPPI) (Chang et al., 2010).

MATLAB Codes of PPI

```
function [eeindex score duration]=PPI(imagecub,skewer_no)
% The Matlab PPI algorithm
% ——— Input variables ——————
% ’imagecub’ - The hyperspectral image cube
% ’skewer_no’ - The number of skewers
%
% ——— Output variables —————
% ’eeindex’ - The locations of the final endmembers (x,y)
% ’score’ - The PPI score of each pixel
% ’duration - The number of seconds used to run this program

% Initial Variables
[rows columns bands]=size(imagecub);
score=zeros(rows*columns,1);
switch_results=1;

% Record the start CPU time
start=cputime();

% Separate the total number of skewers into several sets and each set uses 500
skewers
skewer_sets=floor(skewer_no/500)+1;
last_skewer_no=mod(skewer_no,500);

for i=1:skewer_sets

  if (skewer_sets-i) == 0,
    skewer_no=last_skewer_no;
  else
    skewer_no=500;
  end

  % Generate skewers
  rand(’state’,sum(100*clock));
  skewers=rand(bands,skewer_no)-0.5;

  % Normalize skewers
  for i=1:skewer_no,
    skewers(:,i)=skewers(:,i)/norm(skewers(:,i));
  end

  % project every sample vector to the skewers
  projcub=reshape(imagecub, rows*columns, bands);
  proj_result=projcub*skewers;


  % Find the extrema set for each skewer and add 1 to their score
  for i=1:skewer_no,
    max_pos=find(proj_result(:,i)==max(proj_result(:,i)));
    min_pos=find(proj_result(:,i)==min(proj_result(:,i)));
    score(max_pos)=score(max_pos)+1;
    score(min_pos)=score(min_pos)+1;
  end
end

% Find the pixel which has score larger than 0
result=find(score>0);

% Find the position of the p highest scores
%result=[];
%for i=1:skewer_no,
%     result=[result find(max(score)==score,1)];
%     score(find(max(score)==score,1))=0;
%end

% Convert one dimension to two dimension index
if(switch_results),
 eeindex=translate_index(result,rows,columns,1);
else
  if(mod(result,rows)==0)
    eeindex(2,:)=floor(result./rows);
  else
    eeindex(2,:)=floor(result./rows)+1;
  end
  eeindex(1,:)=mod(result-1,rows)+1;
end

duration=cputime()-start;
```

MATLAB Codes of Fast Iterative Pixel Purity Index

```
function [FinalPositions running_time]=FIPPIoptimized(Image,
InitialSkewers)

% Fast Iterative Pixel Purity Index Algorithm
%
% Input parameters:
%-------------------
% Image: Hyperspectral image data after MNF dimensionality reduction
% InitialSkewers: Positions of ATGP-generated pixels
```

```
%
% Output parameter:
% ————————————————
%  FinalPositions: Positions of FIPPI-generated endmember pixels
%  running_time - The total running time used by this run
%
% Authors: Chein-I Chang and Antonio Plaza
% Minor Modified by Chao-Cheng Wu

% Check CPU time at the beginning
start=cputime;

% Code initialization for data and visualization
[ns,nl,nb]=size(Image);
[VD,kk]=size(InitialSkewers);
Extrema=zeros(ns,nl);
ProjectionScores=zeros(ns,nl);
subplot(2,1,1);
imagesc(Image(:,:,1)); colormap(gray);
title('Pixels extracted by FPPI:');
set(gca,'DefaultTextColor','black','xtick',[],'ytick',[],'data-
aspectratio',[1 1 1]);
po1 = get(gca,'position');

% Use ATGP-generated pixels as the initial skewers
NewSkewers=InitialSkewers;

% Begin iterative process
Other = 1;
SkewersUsed = [];
while (Other >= 1)

  [ne,np]=size(NewSkewers);
  disp(['Iteration: ' int2str(Other)]);
  disp(['Skewers: ' int2str(ne)]);
  for k = 1:ne

    [ne_old,kk]=size(SkewersUsed);
    skewer=squeeze(Image(NewSkewers(k,1),NewSkewers(k,2),:));
    skewer=skewer/norm(skewer);
    SkewersUsed = union(SkewersUsed,skewer);
    [ne_new,kk]=size(SkewersUsed);

    subplot(2,1,2);
    drawnow;
    plot(skewer);
    title(['Current skewer: ' int2str(k)]);
  if (ne_new~=ne_old)
  % Project all the sample data vectors onto this particular skewer
   for i=1:ns
    for j=1:nl
     pixel = squeeze(Image(i,j,:));
     ProjectionScores(i,j) = dot(skewer,pixel);
    end
   end

   % Obtain the extrema set for each skewer (maximum and minimum
   % projection)
   [vals,mpos] = max(ProjectionScores(:));
     [vals,pos] = min(ProjectionScores(:));
   mposx = floor((mpos-1)/ns)+1; mposy = mod(mpos-1,ns)+1;
       posx = floor((pos-1)/ns)+1; posy = mod(pos-1,ns)+1;

   % Display the pixel positions of the pixels in the extrema set
   drawnow;
   subplot(2,1,1);
text(mposx,mposy,’o’,’Margin’,1,’HorizontalAlignment’,’center’,’
FontSize’,22,’FontWeight’,’light’,’FontName’,’Garamond’,’Color’,
’yellow’);

     drawnow;
   subplot(2,1,1);

text(posx,posy,’o’,’Margin’,1,’HorizontalAlignment’,’center’,’
FontSize’,22,’FontWeight’,’light’,’FontName’,’Garamond’,’Color’,
’yellow’);

   % Increase PPI count of extrema pixels
   Extrema(posy,posx)=Extrema(posy,posx)+1;
    Extrema(mposy,mposx)=Extrema(mposy,mposx)+1;

   % Incorporate sample vectors with PPI count greater than zero to
   % the skewer set
   vnew = [ mposx mposy ; posx posy ];
   NewSkewers = union(NewSkewers,vnew,’rows’);
  end
 end

 % Check stopping rule
 [ne2,np]=size(NewSkewers);
 if (ne2==ne)
  Other = 0;
 else
  Other = Other+1;
```

```
        % Extrema=zeros(ns,nl);
  end
end


% Produce the positions of the final endmember set
Binary = Extrema>0;
ne = sum(Binary(:));
disp(['Extracted endmembers: ' int2str(ne)]);
FinalPositions = zeros(ne,2);
Current = 1;
for i=1:ns
  for j=1:nl
    if Binary(i,j)>0
      FinalPositions(Current,1)=i;
      FinalPositions(Current,2)=j;
      Current = Current+1;
    end
  end
end

% Check CPU time at the end
stop=cputime;
running_time=stop-start;
```

A.4.2 N-finder Algorithm

- **Algorithm name:** N-finder algorithm (N-FINDR)
- **Authors:** M.E. Winter
- **Category:** convex geometry
- **Designed criteria:** maximum simplex volume
- **Designed method:** finding a simplex with maximum volume
- **Typical use (LOI's addressed):** endmember extraction with number of endmembers to be known
- **Inputs:** reflectance or radiance cube
- **Outputs:** endmembers
- **Assumptions:** All the vertices of a simplex with maximal volume should be specified by endmembers
- **Sensitivity to LOI (target knowledge):** high to value of *p*
- **Sensitivity to noise:** moderate
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, but not real time
- **Brief description:**
  The N-FINDR (Winter, 1999a, 1999b, 2004) is a popular endmember extraction algorithm (EEA) other than PPI. It is quite different from PPI in terms of its design criterion. The N-FINDR makes use of maximum simplex volume as a criterion as opposed to orthogonal projection used by PPI. So, its design criteria make PPI an unconstrained EEA and N-FINDR a fully

constrained EEA. Like PPI, which requires prior knowledge of the number of skewers, *K*, N-FINDR also needs to know the number of endmembers, *p*, *a priori*. Similar to PPI, N-FINDR also suffers the same issue encountered in PPI, which is its use of random initial conditions that result in unrepeatable and inconsistent endmember results. This issue is also addressed by Chang et al. (Plaza and Chang, 2006; Chang et al., 2011b). Another serious issue arising in N-FINDR implementation that is not encountered in PPI is its very high computational complexity. Many research efforts have been reported to address this issue. Details can be found in Xiong et al. (2011) and Chang (2013). Also, real-time implementation of N-FINDR has also been proposed in Wu et al. (2010) with details in Chang (2013).

MATLAB Codes of N-FINDR

```
function [endmemberindex duration]=NFINDR(imagecube,p)
% The N-FINDR algorithm
% ——— Input variables ——————
% 'imagecube' - The data transformed components [row column band]
% 'p' - The number of endmembers to be generated
%
% if band > p, then the program will automatically use Singular Value Decomposi-
tion to calculate the volume
% ——— Output variables —————
% 'endmemberindex - The locations of the final endmembers (x,y)
% 'duration - The number of seconds used to run this program

% Set initial condition
endmemberindex=[];
newvolume = 0;
prevolume = -1;
[row, column, band]=size(imagecube);
switch_results=1;

% Determine to use SVD to calculate the volume or not
if(band > p),
  use_svd=1;
else
  use_svd=0;
end
% Start to count the CPU computing time
start=cputime();

% Randomly select p initial endmembers
rand('state',sum(100*clock));
for i=1:p
  while(1)
    temp1=round(row*rand);
    temp2=round(column*rand);
    if(temp1>0 & temp2>0)
```

```
         break;
      end
   end
   endmemberindex=[endmemberindex;[temp1 temp2]];
end
endmemberindex=endmemberindex’;

% Generate endmember vector from reduced cub
display(endmemberindex);
endmember=[];
for i=1:p
   if(use_svd)
      endmember=[endmember squeeze(imagecube(endmemberindex(1,i),
endmemberindex(2,i),:))];
   else
      endmember=[endmember squeeze(imagecube(endmemberindex(1,i),
endmemberindex(2,i),1:p-1))];
   end
end

% calculate the endmember’s volume
if(use_svd)
   s=svd(endmember);
   endmembervolume=1;
   for i=1:p,
      endmembervolume=endmembervolume*s(i);
   end
else
   jointmatrix=[ones(1,p) ; endmember];
   endmembervolume=abs(det(jointmatrix))/factorial(p-1);
end

% The main algorithm
while newvolume > prevolume, % if the new generated endmember volume is larger
than the old one, continue the algorithm

   % Use each sample vector to replace the original one, and calculate new volume
   for i=1:row,
      for j=1:column,
         for k=1:p,
            caculate=endmember;
            if(use_svd),
               caculate(:,k)=squeeze(imagecube(i, j, :));
               s=svd(caculate);
               volume=1;
               for z=1:p,
                  volume=volume*s(z);
               end
            else
               caculate(:,k)=squeeze(imagecube(i, j, 1:p-1));
               jointmatrix=[ones(1,p);calculate];
               volume=abs(det(jointmatrix))/factorial(p-1); % The formula of
Simplex volume
            end
            if volume > endmembervolume,
               endmemberindex(:,k)=[i;j];
               endmember=calculate;
               endmembervolume=volume;
            end
         end
      end
   end
   prevolume=newvolume;
   newvolume=endmembervolume;
end

stop=cputime();
duration=stop-start;
% Switch results for the standard
if(switch_results)
   endmemberindex(3,:)=endmemberindex(1,:);
   endmemberindex(1,:)=[];
   endmemberindex=endmemberindex’;
end
```

A.4.3 Simplex Growing Algorithm

- **Algorithm name:** simplex growing algorithm (SGA)
- **Authors:** C.-I Chang
- **Category:** convex geometry
- **Designed criteria:** maximum simplex volume
- **Designed method:** growing maximum-volume simplexes
- **Typical use (LOI’s addressed):** endmember extraction with number of endmembers to be known
- **Inputs:** reflectance or radiance cube
- **Outputs:** endmembers
- **Assumptions:** All the vertices of simplexes generated by SGA must be endmembers
- **Sensitivity to LOI (target knowledge):** high to value of *p*
- **Sensitivity to noise:** moderate
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, but not real time
- **Brief description:**
  The SGA was developed in Chang et al. (2006) to address several serious implementation issues in N-FINDR. It produces one endmember at a time by growing simplexes vertex by vertex to

resolve the computational issue. In addition, the issue caused by random initial conditions can also be addressed by implementing SGA in real time (Chang et al., 2010).

MATLAB Codes of SGA

```
function [endmemberindex duration]=SGA(imagecube,p)
% Simplex Growing Algorithm
% - - - - - - - Input variables - - - - - - - - - - - - -
% 'imagecube' - The data transformed components [row column band]
% 'p' - The number of endmembers to be generated
%
% if band > p, then the program will automatically use Singular Value Decomposi-
tion to calculate the volume
% - - - - - - - Output variables - - - - - - - - - - -
% 'endmemberindex - The locations of the final endmembers (x,y)
% 'duration - The number of seconds used to run this program

% Set initial condition
n=1;
initial=0;
[row, column, band]=size(imagecube);

% Determine to use SVD to calculate the volume or not
if(band > p),
  use_svd=1;
else
  use_svd=0;
end

% Start to count the CPU computing time
start_time=cputime();
% Randomly Select a point as the initial point
endmemberindex=[ceil(row*rand);ceil(column*rand)];

% The main algorithm
while n<p, % if get enough endmember group, it stops

      % Generate endmember vector from reduced cub
      endmember=[];
      for i=1:n
   if(use_svd)
     endmember=[endmember squeeze(imagecube(endmemberindex(1,i),
endmemberindex(2,i),:))];
   else
     endmember=[endmember squeeze(imagecube(endmemberindex(1,i),
endmemberindex(2,i),1:n))];
   end
      end
      % Use each sample vector to calculate new volume
      newendmemberindex=[];
      maxvolume=0;
      for i=1:row,
            for j=1:column,
      if(use_svd)
        jointpoint=[endmember squeeze(imagecube(i,j,:))];
        s=svd(jointpoint);
        volume=1;
        for z=1:n+1,
          volume=volume*s(z);
       end
     else
      jointpoint=[endmember squeeze(imagecube(i,j,1:n))];
      jointmatrix=[ones(1,n+1);jointpoint];
      volume=abs(det(jointmatrix))/factorial(n); % The formula of a simplex
volume
     end
         if volume > maxvolume,
             maxvolume=volume;
             newendmemberindex=[i;j];
          end
        end
    end
    endmemberindex=[endmemberindex newendmemberindex]; % Add this pixel into
the endmember group
     %nfinder_plot(endmemberindex);
    n=n+1;
    if initial==0, % Use new pixel as the initial pixel
         n=1;
         endmemberindex(:,1)=[];
         initial=initial+1;

     end
end

duration=cputime()-start_time;

% Switch the results back to X and Y
endmemberindex(3,:)=endmemberindex(1,:);
endmemberindex(1,:)=[];
endmemberindex=endmemberindex’;
```

A.5 Supervised LSMA and KLSMA

Linear spectral mixture analysis (LSMA) is a mathematical theory that models a data samples as linear mixtures of a finite number of basic spectral constituents with appropriate weights from which data samples can be solved by finding these weights via a linear inverse problem. Linear spectral unmixing is one of its applications. It assumes that there are p basic material substances $\{\mathbf{m}_j\}_{j=1}^{p}$ that can be used to represent data sample vectors in linear forms with their corresponding abundance fractions $\{\alpha_j\}_{j=1}^{p}$ that are unknown parameters. The spectral unmixing is then performed by finding best estimates of $\{\alpha_j\}_{j=1}^{p}$, denoted by $\{\hat{\alpha}_j\}_{j=1}^{p}$, and referred to as unmixed abundance fractions. In real applications, two physical constraints must be imposed on the used linear mixing model, which are abundance sum-to-one constraint (ASC), $\sum_{j=1}^{p}\alpha_j = 1$, and abundance nonnegativity constraint (ANC), $\alpha_j \geq 0$ for all $1 \leq j \leq p$. Three LSMA-based techniques developed in Chang (2003a) have been widely used for spectral unmixing. These are unconstrained orthogonal subspace projection (OSP)/least squares OSP, partially ANC-constrained method, Least Squares nonnegativity-constrained least squares (NCLS) and a fully abundance-constrained method, Constrained Least Squares (FCLS). Since OSP/LSOSP, NCLS, and FCLS are linear techniques, they may have difficulty solving linear nonseparable problems. To address this issue, these three techniques are further extended to their kernel-based counterparts, called Kernel OSP/LSOSP (KOSP/KLSOSP), Kernel NCKS (KNCLS), and Kernel FCLS (KFCLS).

A.5.1 OSP, LSOSP, KOSP, and KLSOSP

- **Algorithm name:** orthogonal subspace projection (OSP)
- **Authors:** J.C. Harsanyi and Chein-I Chang
- **Category:** spectrally matched filter
- **Designed Criteria:** SNR ratio
- **Designed Method:** *A priori*, supervised and unconstrained least squares-based linear spectral mixture analysis
- **Typical use (LOI's addressed):** detection, classification, discrimination, identification
- **Inputs:** reflectance or radiance cube, complete target knowledge
- **Outputs:** gray-scale abundance fractional images
- **Assumptions:** complete prior target knowledge required
- **Sensitivity to LOI (target knowledge):** high
- **Sensitivity to noise:** moderate
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, real Time
- **Brief description:**

  The OSP approach was first developed in Harsanyi's Ph.D. dissertation in 1993 (Harsanyi, 1993) and was later published in *IEEE Transaction on Geoscinece and Remote Sensing*, July 1994 (Harsanyi and Chang, 1994). It is a linear unmixng method that takes advantage of a linear mixture model to detect, classify, and identify targets of interest. The idea is to separate target sources into desired and undesired targets and then use an orthogonal project to reject the desired targets before a matched filtration takes place. So, it operates two functions in sequence: an undesired target rejecter followed by a spectral matched filtration. Since OSP was originally designed as a detector and cannot accurately estimate signature abundance fractions, a least squares version of OSP, referred to as least squares OSP (LSOSP) was further developed in Chang et al. (1998b) for abundance fraction estimation. The only difference between OSP and LSOSP is that LSOSP includes a normalization constant to account for estimation error incurred in the OSP-derived detector (Chang, 2009). So, the MATLAB codes provided in the following is a more general version of OSP, LSOSP.

MATLAB Codes of LSOSP

```
% Least Square Orthogonal Subspace Projection
% input: image = image cube
%      d = desire signature vector
%      U = undesired signature matrix
% output: temp = resulting image cube

function temp=LSOSP(image,d,U)

[x y z]=size(image);
temp = zeros(x,y);

%Find the projectors that is orthogonal complement of U

[l,J] = size(U);
I = eye(l,l);
Pu=I-U*inv(U'*U)*U';
lsosp = (d'*Pu)/(d'*Pu*d);
% perform least-squares-based estimator on all image vectors
for i = 1:x
  for j = 1:y
    for k = 1:z
      r(k) = image(i,j,k);
    end;
    temp(i,j)=lsosp*r';
  end;
end;
```

MATLAB codes of KLSOSP

```
%% Kernel based LSOSP function
% Input:
%    image = image cube input
%    d   = desired signature, example: [2;3;4]
%    U   = undesired signature matrix
%    sig  = parameter that control RBF kernel function
% output:
%    temp  = resulting map

function temp=KOSP(image,d,U,sig)

[x y z]=size(image);
temp = zeros(x,y);

% perform least squares-based estimator on all image vectors

KdU = kernelized(d,U,sig,0);%disp(KdU),
KUU = kernelized(U,U,sig,0);%disp(KUU),
Kdd = kernelized(d,d,sig,0);
KUd = kernelized(U,d,sig,0);%disp(KUd),
for i = 1:x
  for j = 1:y
    for k = 1:z
      r(k,1) = image(i,j,k);
    end;
    Kdr = kernelized(d,r,sig,0);%disp(Kdr),
    KUr = kernelized(U,r,sig,0);%disp(KUr),
    temp(i,j)=(Kdr-KdU*inv(KUU)*KUr);%/(Kdd-KdU*inv(KUU)*KUd);
  end;
end;

%% kernelization function
function results = kernelized(x,y,d,chk)
x_l = size(x,2);
y_l = size(y,2);
results = zeros(x_l,y_l);
for i = 1:x_l
  for j = 1:y_l
    results(i,j)= exp((-1/2)*(norm(x(:,i)-y(:,j))^2)/(d^2));
%RBF kernel (can be changed)
  end
end
if chk == 1
  results = results-(sum(sum(results))/(x_l*y_l))*ones(x_l,y_l);
elseif chk == 2
  N = (1/(x_l*y_l))*ones(x_l,y_l);
  results = results-N*results-results*N+N*results*N;
end
```

A.5.2 NCLS and KNCLS

- **Algorithm name:** nonnegativity least squares (NCLS)
- **Authors:** Chein-I Chang and Daniel Heinz
- **Category:** least squares error-based spectral filter
- **Designed criteria:** least squares error
- **Designed method:** *A priori*, supervised and ANC-constrained LSMA
- **Typical use (LOI's addressed):** detection, classification, discrimination, identification
- **Inputs:** reflectance or radiance cube, complete target knowledge
- **Outputs:** gray-scale abundance fractional images
- **Assumptions:** complete prior target knowledge required
- **Sensitivity to LOI (target knowledge):** high
- **Sensitivity to noise:** moderate
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational

- **Effectiveness:** high
- **Implementation:** simple, easy to use, real time
- **Brief description:**

  Nonnegativity constrained least squares (NCLS) was developed to improve OSP in signal detection (Chang and Heinz, 2000b). It imposes ANC on the linear mixing model to make sure that the unmixed abundance fractions are nonnegative. Despite not being fully constrained, on many occasions NCLS can perform abundance estimation as well as a fully abundance-constrained method in. However, in signal detection, NCLS generally performs better than unconstrained and fully constrained methods.

MATLAB Codes of NCLS

```
function [abundance,error_vector]=NCLS(MatrixZ,r1)
% input MatrixZ is the signatures of endmembers. It is of size [ bands p].
% input x is the signature whose abundance is to be estimated.
% output abundance is the abundance of each material in r1. It is of size [p 1].
% output error_vector is the error vector of size [bands 1].
% This function is written according to Dr. Chang's first book , P 47

x=r1; %rename r1 as x;
M=size(MatrixZ,2);
count_R=0;
count_P=M;
R=zeros(M,1);
P=ones(M,1);
%tolerance=0.000001;
d=zeros(M,1);
Alpha_ls=inv(MatrixZ'*MatrixZ)*MatrixZ'*x;
Alpha_ncls=Alpha_ls;
min_Alpha_ncls=min(Alpha_ncls);
M_t_r=MatrixZ'*x;
invMtM=inv(MatrixZ'*MatrixZ);
while(min_Alpha_ncls<-0.000000001)
  for II=1:M
    if((Alpha_ncls(II)<0)&(P(II)==1))
      R(II)=1;
      P(II)=0;
    end %%% end of if (Alpha_ncls(II)<0)
  end % end of for II=1:M
  S=R;

  goto_step6=1;
  while(1)

    sum_R=sum(R);
    Alpha_R=zeros(sum_R,1);
    count_for_Alpha_R=0;
    for II=1:M
  if (R(II)==1)
    count_for_Alpha_R=count_for_Alpha_R+1;
    Alpha_R(count_for_Alpha_R)=Alpha_ls(II);
    index_for_Lamda(count_for_Alpha_R)=II;
  end
end

count_1_for_P=0;
Sai_column=[];
for II=1:M
  if (P(II)~=1)
    Sai_column=[Sai_column squeeze(invMtM(:,II)) ];

  end
end
Sai=[];
for II=1:M
  if (P(II)~=1)
    Sai=[Sai
      squeeze(Sai_column(II,:)) ];
  end
end

Lamda=inv(Sai)*Alpha_R;
if(max(Lamda)<0)
  break;
end
[max_Lamda,index_Max_Lamda]=max(Lamda);
P(index_for_Lamda(index_Max_Lamda))=1;
R(index_for_Lamda(index_Max_Lamda))=0;

sum_R=sum(R);
Alpha_R=zeros(sum_R,1);
count_for_Alpha_R=0;
for II=1:M
  if (R(II)==1)
    count_for_Alpha_R=count_for_Alpha_R+1;
    Alpha_R(count_for_Alpha_R)=Alpha_ls(II);
    index_for_Lamda(count_for_Alpha_R)=II;
  end
end

Sai_column=[];

for II=1:M
  if (P(II)~=1)
    Sai_column=[Sai_column squeeze(invMtM(:,II)) ];

  end
  end

  Sai=[];
  for II=1:M
    if (P(II)~=1)
      Sai=[Sai
        squeeze(Sai_column(II,:)) ];
    end
  end

  Lamda=inv(Sai)*Alpha_R;

  Phai_column=[];
  for II=1:M
    if (P(II)~=1)
      Phai_column=[Phai_column squeeze(invMtM(:,II)) ];
    end
  end
  if (size(Phai_column,2)~=0)

    Alpha_s=Alpha_ls-Phai_column*Lamda;
  else
    Alpha_s=Alpha_ls;
  end

  goto_step6=0;
  find_smallest_in_S=zeros(M,2);
  find_smallest_in_S(:,1)=Alpha_s;
  find_smallest_in_S(:,2)=[1:M]';
  sort_find=sortrows(find_smallest_in_S,1);

  for II=1:M
    if ((S(II)==1)&(Alpha_s(II)<0))
      P(II)=0;
      R(II)=1;

      goto_step6=1;
    end
  end

end % end of while (gotostep6==1)

Phai_column=[];
for II=1:M
  if (P(II)~=1)
    Phai_column=[Phai_column squeeze(invMtM(:,II)) ];

  end
end
```

```
  if (size(Phai_column,2)~=0)

    Alpha_ncls=Alpha_ls-Phai_column*Lamda;

  else
    Alpha_ncls=Alpha_ls;

  end
  min_Alpha_ncls=min(Alpha_ncls);
end % end of while


abundance=zeros(M,1);
for II=1:M
  if (Alpha_ncls(II)>0)
    abundance(II)=Alpha_ncls(II);
  end
end
error_vector=MatrixZ*abundance-x;
```

MATLAB codes of KNCLS

```
function ab = KNCLS(r,M,d)
% Non-negative constrain Least-square abundance estimator
% input M = signature matrix to be estimated.
% input r = image pixel vector.
% output abundance = abundance vector correspondence to the signature
% matrix.

% initial stage k = 0
k = 0;
[x,y] = size(M);
P = 1:y;
R = [];

%least square estimate of abundance
inv_MTM = inv(kernelized(M,M,d,0));
a_ls = inv_MTM*kernelized(M,r,d,0);
%initially set abundance to least square abundance
ab = a_ls;
%iterate until all ab is positive
while any(ab<-0.5) && k<50
%   disp(k)
%   clc,
  k = k+1;
  % find negative index and move them from P to R
  neg_ind = find(ab<0);
  [P,R] = move_index(P,R,neg_ind);
  S = R;

  %initialize lum so the for loop can run at least once
  lum = 1;
  % iterate until all lum is smaller or equal to zero
  j=0;
  while any(lum>=0) && j<50
%  disp(j)
    j = j+1;
    a_R = a_ls(R);
    % define the steer matrix by removing row and columns defined by P
    a_steer_mat = inv_MTM;
    a_steer_mat(:,P) = [];
    a_steer_mat(P,:) = [];

    % calculate Lagrange multiplier lum
    lum = ((a_steer_mat)^(-1))*a_R;
    % if lum contains positive value
    if any(lum>0) && j<1000
      % find the maximum lum move its index from R to P create a new
      % lum by removing the max lum
      j = j+1;
      lum_max_ind = R(find(lum == max(lum)));
      [R,P]=move_index(R,P,lum_max_ind);
      a_steer_mat = inv_MTM;
      a_steer_mat(:,P) = [];
      a_steer_mat(P,:) = [];
      a_R = a_ls(R);
      lum_new = ((a_steer_mat)^(-1))*a_R;

      % form another lum steering matrix
      lum_steer_mat = inv_MTM;
      lum_steer_mat(:,P) = [];
      if length(lum_steer_mat) == 0
        lum_steer_mat = 0;
      end;
      % calcuate the abundance base on R and P and lum_new. If any
      % value specify by index S is negative move it from P to R.
      a_S = a_ls - lum_steer_mat*lum_new;
      s_neg_ind = S(find(a_S(S)<0));
      [P,R] = move_index(P,R,s_neg_ind);
    end;
  end;

  % calculate the new abundance base on the new lum found
  lum_steer_mat = inv_MTM;
  lum_steer_mat(:,P) = [];
  ab = a_ls - lum_steer_mat*lum;

end;
return;

% Move index function
function [P_new,R_new] = move_index(del_ind, get_ind, move_ind)

n = length(move_ind);
if n>0
  for i = 1:n
    a = find(del_ind == move_ind(i));
    del_ind(a)=[];
    b = get_ind - move_ind(i);
    if any(b == 0)
      get_ind = get_ind;
    else
      get_ind = [get_ind,move_ind(i)];
    end;
  end;
end;

P_new = sort(del_ind);
R_new = sort(get_ind);
return

function results = kernelized(x,y,d,chk)
x_l = size(x,2);
y_l = size(y,2);
results = zeros(x_l,y_l);
for i = 1:x_l
  for j = 1:y_l
    results(i,j)= exp((-1/2)*(norm(x(:,i)-y(:,j))^2)/(d^2));
  end
end
if chk == 1
  results = results-(sum(sum(results))/(x_l*y_l))*ones(x_l,y_l);
elseif chk == 2
  N = (1/(x_l*y_l))*ones(x_l,y_l);
  results = results-N*results-results*N+N*results*N;
end
return,
```

A.5.3 FCLS and KFCLS

- **Algorithm name:** fully constrained least squares (FCLS)
- **Authors:** Daniel Heinz and Chein-I Chang

- **Category:** least squares error-based spectral filter
- **Designed criteria:** least squares error (LSE)
- **Designed Method:** *a priori*, supervised and fully constrained LSMA
- **Typical use (LOI's addressed):** detection, classification, discrimination, identification
- **Inputs:** reflectance or radiance cube, complete target knowledge
- **Outputs:** gray-scale abundance fractional images
- **Assumptions:** complete prior target knowledge required
- **Sensitivity to LOI (target knowledge):** high
- **Sensitivity to noise:** moderate
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, real time
- **Brief description:**

  Fully constrained least squares (FCLS) was developed in Heinz and Chang (2001) as an abundance estimator to accurately estimate abundance fractions. Therefore, as far as unmixing is concerned, FCLS is one of best designed linear estimator. But it does not imply that FCLS is also best for other applications such as detection, discrimination, classification, etc. As a matter of fact, NCLS generally outperforms FCLS in these applications, where NCLS does not comply with the constraint of ASC, specifically, in signal detection, where signals to be detected are corrupted by noise and ASC is certainly violated.

MATLAB Codes of FCLS

```
%- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
% - Fully Constrain Least-Squares (FCLS) abundances estimate -
%
% function: results = FCLS(image,M,tol)
%
% input:  image  = image of size [X,Y,Z]
%     M    = endmember matrix of size [Z,p]
%     tol   = NCLS tolerance, e.g. -1e-6
%
% output:  results = resulting abundance map of size [X,Y,p]
%- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
function results = FCLS_v2(image,M,tol)
[x,y,z] = size(image);
num_p = size(M,2);
results = zeros(x,y,num_p);
for i = 1:x
  for j = 1:y
    r = reshape(image(i,j,:),z,1);
    delta = 1/(10*max(max(M)));
    s = [delta.*r;1];
    N = [delta.*M;ones(1,num_p)];

    [ab] = NCLS(s, N, tol);
    ab = reshape(ab,[1,1,num_p]);
    results(i,j,:) = ab;
  end;
end;
return;
function [abundance]=NCLS(x, MatrixZ, tol)
% input MatrixZ is the signatures of endmembers. It is of size [bands p].
% input x is the signature whose abundance is to be estimated.
% output abundance is the abundance of each material in r1. It is of size [p 1].
% This function is written according to Dr. Chang's first book , P 47

M=size(MatrixZ,2);
R=zeros(M,1);
P=ones(M,1);
invMtM=(MatrixZ'*MatrixZ)^(-1);
Alpha_ls=invMtM*MatrixZ'*x;
Alpha_ncls=Alpha_ls;
min_Alpha_ncls=min(Alpha_ncls);
j=0;
while(min_Alpha_ncls<-tol && j<500)
  j = j+1;
  for II=1:M
    if((Alpha_ncls(II)<0)&&(P(II)==1))
      R(II)=1;
      P(II)=0;
    end %%% end of if (Alpha_ncls(II)<0)
  end % end of for II=1:M
  S = R;

  goto_step6=1;
  counter = 0;
  while(goto_step6==1)
    index_for_Lamda = find(R==1);
    Alpha_R = Alpha_ls(index_for_Lamda);
    Sai = invMtM(index_for_Lamda,index_for_Lamda);

    inv_Sai = (Sai)^(-1);    % remember inversion of Sai
    Lamda=inv_Sai*Alpha_R;

    [max_Lamda,index_Max_Lamda]=max(Lamda);
    counter = counter+1;
    if ( max_Lamda<=0 || counter == 200 )
      break;
    end

    temp_i = inv_Sai;     % simplify the inversion of matrix
    temp_i(1,:) = inv_Sai(index_Max_Lamda,:);
    if index_Max_Lamda>1
```

```
      temp_i(2:index_Max_Lamda,:) = inv_Sai(1:index_Max_Lamda-1,:);
    end
    inv_Sai_ex = temp_i;
    inv_Sai_ex(:,1) = temp_i(:,index_Max_Lamda);
    if index_Max_Lamda>1

      inv_Sai_ex(:,2:index_Max_Lamda) = temp_i(:,1:index_Max_Lamda-1);
    end
     inv_Sai_next = inv_Sai_ex(2:end,2:end) - inv_Sai_ex(2:end,1)*inv_Sai_ex
(1,2:end)/inv_Sai_ex(1,1);

    P(index_for_Lamda(index_Max_Lamda))=1;
    R(index_for_Lamda(index_Max_Lamda))=0;
    index_for_Lamda(index_Max_Lamda) = [];

    Alpha_R = Alpha_ls(index_for_Lamda);
    Lamda=inv_Sai_next*Alpha_R;

    Phai_column = invMtM(:,index_for_Lamda);

    if (size(Phai_column,2)~=0)
      Alpha_s=Alpha_ls-Phai_column*Lamda;
    else
      Alpha_s=Alpha_ls;
    end

    goto_step6=0;

    for II=1:M
      if ((S(II)==1)&&(Alpha_s(II)<0))
        P(II)=0;
        R(II)=1;
        goto_step6=1;
      end
    end
  end % end of while (gotostep6==1)

  index_for_Phai = find(R==1);
  Phai_column = invMtM(:,index_for_Phai);

  if (size(Phai_column,2)~=0)
    Alpha_ncls=Alpha_ls-Phai_column*Lamda;
  else
    Alpha_ncls=Alpha_ls;
  end

  min_Alpha_ncls=min(Alpha_ncls);
end % end of while
```

```
abundance=zeros(M,1);
for II=1:M
  if (Alpha_ncls(II)>0)
    abundance(II)=Alpha_ncls(II);
  end
end
return;
```

MATLAB Codes of KFCLS

```
function [abundance,error_vector]= KFCLS(M,r1,delta,d)
% input M is the signatures of endmembers. It is of size [ bands p].
% input r1 is the signature whose abundance is to be estimated.
% input delta is the parameter to control ASC: see explanation on Dr. Chang's
first book , P 183
% usually, delta is 1/(10*max(max(A));
% output abundance is the abundance of each material in r1. It is of size [p 1].
% output error_vector is the error vector of size [bands 1].
tic
%Dan's: optimal
A=M;
numloop=size(A,2);
e=delta;
eA=e*A;
E=[ones(1,numloop);eA];
EtE=kernelized(E,E,d,0);
[m,n] = size(EtE);
One=ones(m,1);
iEtE=inv(EtE);
iEtEOne=iEtE*One;
sumiEtEOne=sum(iEtEOne);
weights=diag(iEtE);

c=0;
sample=r1;
er=e*sample;
f=[1;er];
Etf=kernelized(E,f,d,0);

tol=1e-7;
%fcls1a

%%%% THIS IS lamdiv2
ls=iEtE*Etf;
lamdiv2=-(1-(ls'*One))/sumiEtEOne;
x2=ls-lamdiv2*iEtEOne;
x2old=x2;
if (any(x2<-tol))
```

```
  Z=zeros(m,1);
  iter=0;
  while(any(x2<-tol) && iter >(m))
    Z(x2<-tol)=1;
    zz=find(Z);
    x2=x2old;      % Reset x2
    L=iEtE(zz,zz);
    ab=size(zz);
    lastrow=ab(1)+1;
    lastcol=lastrow;
    L(lastrow,1:ab(1))=(iEtE(:,zz)'*One)';
    L(1:ab(1),lastcol)=iEtEOne(zz);
    L(lastrow,lastcol)=sumiEtEOne;
    xerow=x2(zz);
    xerow(lastrow,1)=0;
    lagra=L\xerow;
    while (any(lagra(1:ab(1))>0))  % Reset Lagrange multipliers
      maxneg=weights(zz).*lagra(1:ab(1));
      [yz,iz]=max(maxneg);  % Remove the most positive
      Z(zz(iz))=0;
      zz=find(Z);      % Will always be at least one (prove)
      L=iEtE(zz,zz);
      ab=size(zz);
      lastrow=ab(1)+1;
      lastcol=lastrow;
      L(lastrow,1:ab(1))=(iEtE(:,zz)'*One)';
      L(1:ab(1),lastcol)=iEtEOne(zz);
      L(lastrow,lastcol)=sumiEtEOne;
      xerow=x2(zz);
      xerow(lastrow,1)=0;
      lagra=L\xerow;
    end
    %problem with lamscls zz may be null
    if ~isempty(zz)
      x2=x2-iEtE(:,zz)*lagra(1:ab(1))-lagra(lastrow)*iEtEOne;
    end
    iter=iter+1;
  end
end
abundance=x2;
error_vector=A*abundance-r1;

function results = kernelized(x,y,d,chk)
x_l = size(x,2);
y_l = size(y,2);
results = zeros(x_l,y_l);
for i = 1:x_l
  for j = 1:y_l
```

```
    results(i,j)=exp((-1/2)*(norm(x(:,i)-y(:,j))^2)/(d^2));
  end
end
if chk == 1
  results = results-(sum(sum(results))/(x_l*y_l))*ones(x_l,y_l);
elseif chk == 2
  N = (1/(x_l*y_l))*ones(x_l,y_l);
  results = results-N*results-results*N+N*results*N;
end
```

A.6 Unsupervised Hyperspectral Target Detection

One of major strengths resulting from hyperspectral imaging is the ability to find subtle targets of interest such as subpixel targets, anomalies that cannot be resolved by multispectral imaging. However, it also comes with an issue of how to find them, since such targets are generally not visualized by inspection and must be found in an unsupervised manner without prior knowledge. This part extends the three supervised LSMA-based techniques, OSP/LSOSP, NCLS, and FCLS, to their unsupervised counterparts, automatic target generation process (ATGP), unsupervised NCLS (UNCLS), and unsupervised FCLS (UFCLS).

A.6.1 ATGP

- **Algorithm name:** automatic target generation process (ATGP)
- **Authors:** H. Ren and Chein-I Chang
- **Category:** unsupervised least squares-based filter
- **Designed criteria:** orthogonal projection (OP)
- **Designed method:** maximum OP
- **Typical use (LOI's addressed):** detection, classification, discrimination, identification
- **Inputs:** reflectance or radiance cube
- **Outputs:** gray-scale abundance fractional images
- **Assumptions:** no target knowledge required
- **Sensitivity to LOI (target knowledge):** high
- **Sensitivity to noise:** moderate
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, real time
- **Brief description:**

  The ATGP is derived from an algorithm, called automatic target detection, and classification (ATDCA) developed by Ren and Chang to find target pixels of interest for recognition without having prior target knowledge (Ren and Chang, 2003). It performs successive orthogonal projections to find a data sample vector, considered as a target of interest, which has the maximal projection after each orthogonal projection. The potential of ATGP has been shown to have a wide range of applications, such as VD determination, anomaly detection, endmember extraction, unsupervised LSMA, etc.

MATLAB Codes of ATGP

```
function [loc,Sig]=ATGP_new(HIM,num_targets);

% input HIM is the Hyperspectral data cube of size [height width bands].
% input M is the number of target p that you need to extract.
% output Sig is the signatures corresponding to the extracted targets. It is of
size [bands p].
% output loc is the positions of the targets. It is of size [p 2]
% with the first column being the vertical position, the second column being the
horizontal position.

bnd=size(HIM,3);
xx=size(HIM,1);
yy=size(HIM,2);

r=reshape(HIM,xx*yy,bnd);
r=r';
%=====Find the first point

temp=sum(r.*r);
[a,b]=max(temp);
if (rem(b,xx)==0)
  Loc(1,1)=b/xx;
  Loc(1,2)=xx;
elseif (floor(b/xx)==0)
  Loc(1,1)=1;
  Loc(1,2)=b;
else
  Loc(1,1)=floor(b/xx)+1; % y
  Loc(1,2)=b-xx*floor(b/xx); % x
end

Sig(:,1)=r(:,b);
%==========
for m=2:num_targets
  U=Sig;
  P_U_perl=eye(bnd)-U*inv(U'*U)*U';
  y=P_U_perl*r;
  temp=sum(y.*y);
  [a,b]=max(temp);
  if (rem(b,xx)==0)
    Loc(m,1)=b/xx;
    Loc(m,2)=xx;
  elseif (floor(b/xx)==0)
    Loc(m,1)=1;
    Loc(m,2)=b;
```

```
else
    Loc(m,1)=floor(b/xx)+1; % y
    Loc(m,2)=b-xx*floor(b/xx); % x
  end
  Sig(:,m)=r(:,b);
  %disp(m)
end
%
% figure; imagesc(HIM(:,:,30)); colormap(gray); hold on
% axis off
% axis equal
% for m=1:size(Loc,1)
%   plot(Loc(m,1),Loc(m,2),'o','color','g');
%   text(Loc(m,1)+2,Loc(m,2),num2str(m),'color','y','FontSize',12);
% end
%

loc(:,1)=Loc(:,2);
loc(:,2)=Loc(:,1);
```

A.6.2 UNCLS

- **Algorithm name:** unsupervised non-negativity constrained least squares (UNCLS)
- **Authors:** Chein-I Chang and Daniel Heinz
- **Category:** unsupervised least-squares-based filter
- **Designed criteria:** least squares error
- **Designed method:** least squares constrained method
- **Typical use (LOI's addressed):** detection, spectral unmixing, classification, quantification
- **Inputs:** reflectance or radiance cube
- **Outputs:** gray-scale abundance fractional images
- **Assumptions:** no target knowledge required
- **Sensitivity to LOI (target knowledge):** moderate
- **Sensitivity to noise:** moderate
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, but not real time
- **Brief description:**

  The UNCLS is an unsupervised version of the NCLS developed by Chang and Heinz (2000b). Unlike NCLS, which requires complete *a priori* target knowledge to perform spectral unmixing, UNCLS performs unsupervised target detection by using the NCLS to generate a set of potential targets directly from the data to be processed. So, its main goal is to find targets of interest without any prior knowledge. Because of that, UNCLS is primarily used for target detection and endmember extraction as opposed to NCLS, whose main functionality is spectral unmixing. That is, the NCLS and the UNCLS have rather different applications.

MATLAB Codes of UNCLS

```
function [location_pair,spectrum_of_targets]=UNCLS(ImageCub,
NumberOfClass);
```

```
% input ImageCub is of size [height width bands].
% input NumberOfClass is the number of target p that you need to extract.
% output spectrum_of_targets is the signatures corresponding to the extracted
targets. It is of size [bands p].
% output location_pair is the positions of the targets. It is of size [p 2] with
the first column being the vertical position, the second column being the
horizontal position.

NumberOfClass=NumberOfClass-1;
[height, width, NumberOfSpectrum]=size(ImageCub);
[xx, yy, bnd]=size(ImageCub);

r=reshape(ImageCub,xx*yy,bnd);
r=r';
%height=size(ImageCub,1);
%width=size(ImageCub,2);
%MatrixH=zeros( NumberOfSpectrum, NumberOfClass+1);

location_pair=zeros(NumberOfClass,2);
Brightest=10^(-100);
count=0;
NumberOfExisted=1;

%=====Find the first point=====

temp=sum(r.*r);
[a,b]=max(temp);

if (rem(b,xx)==0)
  Loc(1,1)=b/xx;
  Loc(1,2)=xx;
elseif (floor(b/xx)==0)
  Loc(1,1)=1;
  Loc(1,2)=b;
else
  Loc(1,1)=floor(b/xx)+1; % y
  Loc(1,2)=b-xx*floor(b/xx); % x
end

%location_pair(1:NumberOfExisted,:)=squeeze(BrightForEachClass(1,:));
location_pair(:,1)=Loc(:,2);
location_pair(:,2)=Loc(:,1);
MatrixH=r(:,b);

for II=1:NumberOfClass
%  disp([ ' Class : ' int2str(II) ]);
%  disp(II+NumberOfExisted);
  minError=0.000000000000000000000001;
```

```
 for b=1:xx*yy
   trypixel=r(:,b);
   max_value=max(trypixel);
   if (max_value>0)
    [abundance]= NCLS(MatrixH,trypixel);
    error_vector=MatrixH*abundance-trypixel;
    error=error_vector’*error_vector;
    if(error>minError)
     minError=error;
     if (rem(b,xx)==0)
      location_pair(II+NumberOfExisted,1)=xx;
      location_pair(II+NumberOfExisted,2)=b/xx;
     elseif (floor(b/xx)==0)
      location_pair(II+NumberOfExisted,1)=b;
      location_pair(II+NumberOfExisted,2)=1;
     else
      location_pair(II+NumberOfExisted,1)=b-xx*floor(b/xx);
      location_pair(II+NumberOfExisted,2)=floor(b/xx)+1;
     end
    end
  end
 end
 MatrixH=[MatrixH
squeeze(ImageCub(location_pair(II+NumberOfExisted,1),location_pair(II+
NumberOfExisted,2),:))];
end
spectrum_of_targets=MatrixH;
```

A.6.3 UFCLS

- **Algorithm name:** unsupervised fully constrained least squares (UFCLS)
- **Authors:** Daniel Heinz and Chein-I Chang
- **Category:** unsupervised least-squares-based filter
- **Designed criteria:** least squares error
- **Designed method:** least squares constrained method
- **Typical use (LOI’s addressed):** detection, spectral unmixing, classification, quantification
- **Inputs:** reflectance or radiance cube
- **Outputs:** gray-scale abundance fractional images
- **Assumptions:** no target knowledge required
- **Sensitivity to LOI (target knowledge):** moderate
- **Sensitivity to noise:** moderate
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:** simple, easy to use, but not real time
- **Brief description:**

  The UFCLS is an unsupervised version of the FCLS developed by Heinz and Chang (2001). Unlike FCLS, which requires complete *a priori* target knowledge to perform spectral unmixing for abundance fraction estimation, UFCLS performs unsupervised target detection by using

FCLS to generate a set of potential targets directly from the data to be processed. It is identical to UNCLS with the only difference that UFCLS uses FCLS instead of NCLS to find targets of interest. So, its main goal is to find targets of interest without any prior knowledge. Because of that, UFCLS is primarily used for target detection and endmember extraction as opposed to FCLS, whose main functionality is to unmix data sample vectors in terms of abundance fractions. Therefore, FCLS and UFCLS indeed have different applications.

MATLAB Codes of UFCLS

```
function [location_pair,spectrum_of_targets]=UFCLS(ImageCub,
NumberOfClass);
% input ImageCub is of size [height width bands].
% input NumberOfClass is the number of target p that you need to extract.
% output spectrum_of_targets is the signatures corresponding to the extracted
targets. It is of size [bands p].
% output location_pair is the positions of the targets. It is of size [p 2] with
the first column being the vertical position, the second column being the
horizontal position.

NumberOfClass=NumberOfClass-1;
[height, width, NumberOfSpectrum]=size(ImageCub);
[xx, yy, bnd]=size(ImageCub);

r=reshape(ImageCub,xx*yy,bnd);
r=r';
%height=size(ImageCub,1);
%width=size(ImageCub,2);
%MatrixH=zeros( NumberOfSpectrum, NumberOfClass+1);

location_pair=zeros(NumberOfClass,2);
Brightest=10^(-100);
count=0;
NumberOfExisted=1;

%=====Find the first point

temp=sum(r.*r);
[a,b]=max(temp);

if (rem(b,xx)==0)
  Loc(1,1)=b/xx;
  Loc(1,2)=xx;
elseif (floor(b/xx)==0)
  Loc(1,1)=1;
  Loc(1,2)=b;
else
  Loc(1,1)=floor(b/xx)+1; % y
  Loc(1,2)=b-xx*floor(b/xx); % x
```

```
end

%location_pair(1:NumberOfExisted,:)=squeeze(BrightForEachClass(1,:));
location_pair(:,1)=Loc(:,2);
location_pair(:,2)=Loc(:,1);
MatrixH=r(:,b);
delta=1/10/max(max(MatrixH));
for II=1:NumberOfClass
%  disp([ ’ Class : ’ int2str(II) ]);
%  disp(II+NumberOfExisted);
  minError=0.0000000000000000000000001;
  for b=1:xx*yy
     trypixel=r(:,b);
     max_value=max(trypixel);
     if (max_value>0)
       [abundance]= FCLS(MatrixH,trypixel,delta);
       error_vector=MatrixH*abundance-trypixel;
       error=error_vector’*error_vector;
       if(error>minError)
        minError=error;
        if (rem(b,xx)==0)
          location_pair(II+NumberOfExisted,1)=xx;
          location_pair(II+NumberOfExisted,2)=b/xx;
        elseif (floor(b/xx)==0)
          location_pair(II+NumberOfExisted,1)=b;
          location_pair(II+NumberOfExisted,2)=1;
        else
          location_pair(II+NumberOfExisted,1)=b-xx*floor(b/xx);
          location_pair(II+NumberOfExisted,2)=floor(b/xx)+1;
        end
      end
    end
  end
  MatrixH=[MatrixH
squeeze(ImageCub(location_pair(II+NumberOfExisted,1),location_pair(II+
NumberOfExisted,2),:))];
end
spectrum_of_targets=MatrixH;
```

A.7 Constrained Band Selection

DR and band selection (BS) have been widely used for data compression. MATLAB codes of several component analysis-based DR techniques have been provided in Section A.3. This section presents a new approach to BS, called constrained band selection (CBS), which is based on the constrained energy minimization (CEM) developed for target detection in Chapter 2.

- **Algorithm name:** constrained band selection (CBS)
- **Authors:** Chein-I Chang and Su Wang

- **Category:** spectral filter
- **Designed criteria:** least squares error
- **Designed method:** linearly least squares constrained method
- **Typical use (LOI's addressed):** BS
- **Inputs:** reflectance or radiance cube
- **Outputs:** gray-scale abundance fractional images
- **Assumptions:** prior knowledge of the number of bands to be selected
- **Sensitivity to LOI (target knowledge):** moderate
- **Sensitivity to noise:** moderate
- **Operating bands:** VNIR through LWIR
- **Maturity:** mature/operational
- **Effectiveness:** high
- **Implementation:**
- **Brief description:**
  CBS was introduced in Chang and Wang (2006) to explore the idea of CEM that can be used for BS. It interprets a band image as a desired target signature vector while considering other band images as unknown signature vectors. As a result, the proposed CBS linearly constrains a band image while also minimizing band correlation or dependence provided by other band images is referred to as CEM-CBS. Four different criteria, referred to as band correlation constraint (BCC), band correlation minimization (BCM), band dependence constraint (BDC), and band dependence minimization (BDM), are derived for CEM-CBS. Since dimensionality resulting from conversion of a band image to a vector may be huge, the CEM-CBS is further reinterpreted as linearly constrained minimum variance (LCMV)-based CBS by constraining a band image as a matrix, where the same four criteria BCM, BCC, BDC, and BDM can also be used for LCMV-CBS. In order to determine the number of bands required to select, *p*, a recently developed concept, called virtual dimensionality (VD) is used to estimate the *p*. Once the *p* is determined, a set of *p* desired bands can be selected by the CEM/LCMV-CBS. In what follows, only MATLAB codes of four versions of CEM-CBS are provided, but these codes can be easily modified for their counterparts of LCMV-CBS.

MATLAB Codes of CEM/BCC

```
function [ band_select, newcube ] = CEM_BCC(imagecube, num);

close all;
[ xx, yy, band_num ] = size(imagecube);
%%%% get band image correlation %%%%
test_image = reshape(imagecube, xx*yy, band_num);
R = test_image * test_image'/band_num;
tt = inv(R);

%%%% get prioritization score for each band %%%%
for i = 1: band_num
  endmember_matrix = reshape(squeeze(imagecube(:, :, i)), xx*yy, 1);
  W = tt * endmember_matrix * inv(endmember_matrix' * tt * endmember_matrix);
  for j = 1: band_num
    if (i ~= j)
```

```
      test = reshape(squeeze(imagecube(:, :, j)), xx*yy, 1);
      score(i, j) = test' * W;
    else
      score(i,j) = 1;
    end
  end
end
clear endmember_matrix;
clear W;
clear i,j;

%%%% select small subset of original bands %%%%
weight = zeros(1, band_num);
for i = 1: band_num
  test = score(i,:);
  scalar = sum(test) - score(i,i);
  weight(i) = scalar;
end
%weight = abs(weight);
original = 1:band_num;
coefficient_integer = weight * 100000;
band_select = zeros(1, num);
i = 1;
while (i <= num)
  max_coe = max(coefficient_integer(:));
  index = find(coefficient_integer == max_coe);
  if (length(index) == 1)
    band_select(i) = original(index);
    i = i + 1;
  else
    j = 1;
    while (j <= length(index)) & (i <= num )
      band_select(i) = original(index(j));
      i = i + 1;
      j = j + 1;
    end
  end
  coefficient_integer(index) = -10^10;
end
band_sort = sort(band_select);
clear coefficient_integer;
clear max_coe
clear index;
clear i;
clear j;

%get new imagecube
newcube = zeros(xx,yy, num);
```

```
for i = 1: xx
  for j = 1: yy
    test = squeeze(imagecube(i,j,:));
    newcube(i,j,:) = test(band_sort);
  end
end
```

MATLAB Codes of CEM/BCM

```
function [ band_select, newcube ] = CEM_BCM(imagecube, num);

close all;
[ xx, yy, band_num ] = size(imagecube);

%%%% get band image correlation %%%%
test_image = reshape(imagecube, xx*yy, band_num);
R = test_image * test_image'/band_num;

%%%% get prioritization score for each band %%%%
for i = 1: band_num
  endmember_matrix = reshape(squeeze(imagecube(:, :, i)), xx*yy, 1);
  W = tt * endmember_matrix * inv(endmember_matrix' * tt * endmember_matrix);
  score(i) = W' * R * W;
end
clear endmember_matrix;
clear W;
clear i;

%%%% select small subset of original bands %%%%
%weight = abs(score);
original = 1:band_num;
coefficient_integer = weight * 100000;
band_select = zeros(1, num);
i = 1;
while (i <= num)
  max_coe = max(coefficient_integer(:));
  index = find(coefficient_integer == max_coe);
  if (length(index) == 1)
    band_select(i) = original(index);
    i = i + 1;
  else
    j = 1;
    while (j <= length(index)) & (i <= num )
      band_select(i) = original(index(j));
      i = i + 1;
      j = j + 1;
    end
  end
  coefficient_integer(index) = -10^10;
end
band_sort = sort(band_select);
clear coefficient_integer;
clear max_coe
clear index;
clear i;
clear j;

%get new imagecube
newcube = zeros(xx,yy, num);
for i = 1: xx
  for j = 1: yy
    test = squeeze(imagecube(i,j,:));
    newcube(i,j,:) = test(band_sort);
  end
end
```

MATLAB Codes of CEM/BDM

```
function [ band_select, newcube ] = CEM_BDM(imagecube, num);

close all;
[ xx, yy, band_num ] = size(imagecube);

%%%% get band image correlation %%%%
test_image = reshape(imagecube, xx*yy, band_num);
R = test_image * test_image'/band_num;

%%%% get prioritization score for each band %%%%
for i = 1: band_num
  endmember_matrix = reshape(squeeze(imagecube(:, :, i)), xx*yy, 1);
  R_new = R - endmember_matrix * endmember_matrix';
  R_new = R_new/(band_num -1);
  tt = inv(R_new);
  W = tt * endmember_matrix * inv(endmember_matrix' * tt * endmember_matrix);
  score(i) = W' * R * W;
end
clear endmember_matrix;
clear W;
clear i;

%%%% select small subset of original bands %%%%
%weight = abs(score);
original = 1:band_num;
coefficient_integer = weight * 100000;
```

```
band_select = zeros(1, num);
i = 1;
while (i <= num)
  max_coe = max(coefficient_integer(:));
  index = find(coefficient_integer == max_coe);
  if (length(index) == 1)
    band_select(i) = original(index);
    i = i + 1;
  else
    j = 1;
    while (j <= length(index))&(i <= num )
      band_select(i) = original(index(j));
      i = i + 1;
      j = j + 1;
    end
  end
  coefficient_integer(index) = -10^10;
end
band_sort = sort(band_select);
clear coefficient_integer;
clear max_coe
clear index;
clear i;
clear j;
%get new imagecube
newcube = zeros(xx,yy, num);
for i = 1: xx
  for j = 1: yy
    test = squeeze(imagecube(i,j,:));
    newcube(i,j,:) = test(band_sort);
  end
end
```

References

Acito, N., M. Diani and G. Corsini (2009), "A new algorithm for robust estimation of the signal subspace in hyperspectral images in presence of rare signal components," *IEEE Trans. Geosci. Rem. Sens.*, vol. 47, no. 11, pp. 3844–3856, Nov.

Acito, N., M. Diani and G. Corsini (2010), "Hyperspectral signal subspace identification in the presence of rare signal components," *IEEE Trans. Geosci. Rem. Sens.*, vol. 48, no. 4, pp. 1940–1954, Apr.

Adams, J.B. and M.O. Smith (1986), "Spectral mixture modeling: a new analysis of rock and soil types at the Viking Lander 1 suite," *J. Geophys. Res.*, vol. 91, no. B8, pp. 8098–8112, Jul.

Adams, J.B., M.O. Smith and A.R. Gillespie (1989), "Simple models for complex natural surfaces: a strategy for hyperspectral era of remote sensing," *Proc. IEEE Int. Geoscience and Remote Sensing Symposium'89*, pp. 16–21.

Adams, J.B., M.O. Smith and A.R. Gillespie (1993), "Image spectroscopy: interpretation based on spectral mixture analysis," *Remote Geochemical Analysis: Elemental and Mineralogical Composition*, edited by C.M. Pieters and P. A. Englert, Cambridge University Press, Cambridge, pp. 145–166.

Ahmed, M.N., S.M. Yamany, N. Moharmed, A.A. Farag and T. Moriarty (2002), "A modified fuzzy c-means algorithm for bias field estimation and segmentation of MRI data," *IEEE Trans. Med. Imag.*, vol. 21, no. 3, pp. 193–199, March.

Aho, A.V., J.E. Hoptcraft and J.D. Ullman (1974), *Design and Analysis of Computer Algorithms*, Addison.

Akaike, H. (1974), "A new look at the statistical model identification," *IEEE Trans. Auto. Cont.*, vol. AC-19 pp. 716–723.

Akam Bita, I.P., M. Barret and D.T. Pham (2010), "On optimal transforms in lossy compression of multicomponent images with JPEG2000," *Signal Process.*, vol. 90, no. 3, pp. 759–773.

Akansu, A.N. and R.A. Haddad (1992), *Multiresolution Signal Decomposition*, Academic Press, New York.

Alsing, S., E.P. Blasch and R. Bauer (1999), "3D ROC surface concepts for evaluation of target recognition algorithms faced with the unknown target detection problem," *SPIE Int. Sym. on Aerospace/Defense Simulation and Control*, vol. 3718, Orlando, FL, pp. 449–458, 13–17 Apr.

Althouse, M.L.G. and C.-I Chang (1991), "Chemical vapor detection with a multispectral thermal imager," *Opt. Eng.*, vol. 30, no. 11, pp. 1725–1733, Jul. (invited paper).

Amari, S. (1999), "Natural gradient learning for over- and under-complete bases in ICA," *Neural Comput.*, vol. 11, pp. 1875–1883.

Anderson, T.W. (1984), *An Introduction to Multivariate Statistical Analysis*, 2nd ed., John Wiley & Sons, New York.

Ashton, E.A. and A. Schaum (1998), "Algorithms for the detection of sub-pixel targets in multispectral imagery," *Photogramm. Eng. Rem. Sens.* pp. 723–731, Jul.

Bajorski, P. (2009), "Dose virtual dimensionality work in hyperspectral images?" *Proc. of SPIE*, vol. 7334, pp. 73341J1–73341J11.

Basedow, R., P. Silverglate, W. Rappoport, R. Rockwell, D. Rosenberg, K. Shu, R. Whittlesey and E. Zalewski (1992), "The HYDICE instrument design," *Proc. Int. Symp. Spectral Sens. Res.*, vol. 1, pp. 430–445.

Bates, C., A.P. Asner and C.A. Wessman (2000), "Endmember bundles: a new approach to incorporating endmember variability into spectral mixture analysis," *IEEE Trans. Geosci. Rem. Sens.*, vol. 38, pp. 1083–1094.

Bates, C. and B. Curtis (1996), "A method for manual endmember selection and spectral unmixing," *Rem. Sens. Environ.*, vol. 55, pp. 229–243.

Bauman, J., E. Blasch, J Jackson and G. Sterling (2005), "Real-time ROC: a near-real-time performance evaluation tool," *SPIE 05*, vol. 5807, pp. 380–390, Apr.

Bayliss, J., J.A. Gualtieri and R.F. Cromp (1997), "Analyzing hyperspectral data with independent component analysis," *Proc. SPIE*, vol. 3240, pp. 133–143.

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.

Behrens, R.T. and L.L. Scharf (1994), "Signal processing applications of oblique projections operators," *IEEE Trans. Signal Process.*, vol. 42, no. 6, pp. 1413–1423, Jun.

Bell, T.C., J.G. Cleary and I.H. Witten (1990), *Text Compression*, Prentice-Hall, Englewood Cliffs, NJ.

Bell, A.J. and T.J. Sejnowski (1995), "An information-maximization approach to blind separation and blind deconvolution," *Neural Comput.*, vol. 7, pp. 1129–1160.

Benediktsson, J.A., J.A. Palmason and J.R. Sveinson (2005), "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geosci. Rem. Sens.*, vol. 43, pp. 480–491, Mar.

Berman, M., H. Kiiveri, R. Lagerstrom, A. Ernst, R. Dunne and J.F. Huntington (2004), "ICE: a statistical approach to identifying endmembers in hyperspectral images," *IEEE Trans. Geosci. Rem. Sens.*, vol. 42, no. 10, pp. 2085–2095, Oct.

Bezdek, J.C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York.

Bezdek, J.C., R. Ehrlich and W. Full (1984), "FCM: the fuzzy *c*-means clustering algorithm," *Comp. Geosci.*, vol. 10, no. 2-3 pp. 191–203.

Bioucas-Dias, J.M. and José Nascimento (2005), "Estimation of signal subspace on hyperspectral data," *Proc. SPIE*, vol. 5982, pp. 191–198, Sept.

Bioucas-Dias, J.M. and José Nascimento (2008), "Hyperspectral subspace identification," *IEEE Trans. Geosci. Rem. Sens.*, vol. 46, no. 8, pp. 2435–2445, Aug.

Bischop, C.M. (1995), *Neural Networks for Pattern Recognition*, Oxford University Press.

Bischop, C.M. (2006), *Pattern Recognition and Machine Learning*, Springer

Blasch, E.P. (2008), "User refinement in information fusion," Chapter 19, *Handbook of Multisensor Data Fusion*, 2nd ed., edited by D. Hall and J. Llinas, CRC Press, Boca Raton, FL.

Blasch, E.P., S. Alsing and R. Bauer (1999), "Comparison of bootstrap and prior probability synthetic data balancing method for SAR target recognition," *SPIE Int. Symp. Aerosp./Defense Sim. Control*, vol. 3721, pp. 740–747, Apr.

Blasch, E.P. and R. Broussard (2000), "Physiologically motivated computational visual target recognition beta selection," *SPIE Int. Symp. Aerosp./Defense Sim. Control, App. Sci. Comp. Intell.*, vol. 4055, Orlando, FL, pp. 442–451, 24–28 Apr.

Blasch, E.P., J. Hoffman, and J. Petty (2001), "Defining a fusion gain–system operation characteristic curve," *SPIE Int. Symp. Aerosp./Defense Simul. Control, Conference on Sensor Fusion X*, vol. 4380, pp. 397–405, Apr.

Blasch, E.P. and S. Plano (2003), "Level 5: user refinement to aid the fusion process," *SPIE 03*, vol. 5095, pp. 288–297, Apr.

Boardman, J.W. (1989), "Inversion of imaging spectrometry data using singular value decomposition," *Proc. IEEE Sym. Geosci. Rem. Sens.*, pp. 2069–2072.

Boardman, J.W. (1990), "Inversion of high spectral resolution data," *Proc. SPIE*, vol. 1298, pp. 222–233.

Boardman, J.W. (1993), "Automated spectral unmixing of AVIRIS data using convex geometry concepts," *Summaries, Fourth JPL Airborne Geosci. Workshop*, JPL Publication 93-26, Pasadena, CA, vol. 1, pp. 11–14.

Boardman, J.W. (1994), "Geometric mixture analysis of imaging spectrometry data," *Int. Geosci. Rem. Sens. Symp.*, vol. 4, pp. 2369–2371.

Boardman, J.W. (1998), "Leveraging the high dimensionality of AVIRIS data for improved sub-pixel target unmixing and rejection of false positive: mixture tuned matched filtering," *Summaries of Seventh Annual JPL Earth Science Workshop*, JPL Publication 98-94, Pasadena, CA, vol. 1.

Boardman, J.W., F.A. Kruse and R.O. Green (1995), "Mapping target signatures via partial unmixing of AVIRIS data," *Summaries, Fifth JPL Airborne Geosci. Workshop*, JPL Publication 23-26, Pasadena, CA.

Bowles, J.H. and D.B. Gilles (2007), "An optical real-time adaptive spectral identification system," Chapter 4, *Hyperspectral Data Exploitation*, edited by C.-I Chang, pp. 77–106.

Bowles, J., P. Palmadesso, J. Antoniades and M. Baumback (1995), "Use of filter vectors in hyperspectral data analysis," *SPIE*, vol. 2553, pp. 148–157.

Bro, R. and S.D. Jong (1997), "A fast non-negativity-constrained least squares algorithm," *J. Chemometrics*, vol. 11, pp. 393–401.

Broadwater, J. and A. Banerjee (2009), "A Neyman–Pearson approach to estimating the number of endmembers," *2009 IEEE IGARSS*, pp. IV-693–IV-696, Jul.

Broadwater, J., J.R. Chellappa, A. Banerjee and P. Burlina (2007), "Kernel fully constrained least squares abundance estimates," *2007 IEEE Int. Geosci. Rem. Sens. Symp. (IGARSS)*, pp. 4041–4044, Jul.

Bruce, L.M., C.H. Koger and J. Li (2002), "Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction," *IEEE Trans. Geosci. Rem. Sens.*, vol. 40, no. 10, pp. 2331–2338, Oct.

Brumbley, C. (1998), *Kalman Filtering and Subspace Projection Approaches to Multispectral and Hyperspectral Image Classification*, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD, May.

Brumbley, C. and C.-I Chang (1999), "An unsupervised vector quantization-based target signature subspace projection approach to classification and detection in unknown background," *Pattern Recognit.*, vol. 32, no. 7, pp. 1161–1174, Jul.

Bullock, M.E., T.J. Patterson and S.R. Fairchild (1994), "Design of optimal transformation for multispectral change detection using projection pursuit," *SPIE*, vol. 2231, pp. 91–102.

Burt, P.J. and E.H. Adelson (1983), "The Laplacian Pyramid as an image code," *IEEE Trans. Commun.*, vol. COM-31, no. 4, pp. 532–540, Apr.

Camps-Valls, G. and L. Bruzzone (2009), *Kernel Methods for Remote Sensing Data Analysis*, John Wiley & Sons, New York, p. 256.

Cardoso, J.-F. and B.H. Laheld (1996), “Equivalent adaptive source separation,” *IEEE Trans. Signal Process.*, vol. 44, no. 12, pp. 3017–3030, Dec.

Cawse, K., M. Sears, A. Robin, S.B. Damelin, K. Wessels, F. van den Bergh and R. Mathieu (2010), “Using random matrix theory to determine the number of endmembers in a hyperspectral image,” 2nd *IEEE GRSS Workshop on Hyperspectral Image and Signal Processing—Evolution in Remote Sensing*, pp. 1–4, Jun.

Chai, J.W., J. Wang and C.-I Chang (2007), “Mixed PCA/-ICA transform for hyperspectral image analysis,” *Opt. Eng.*, vol. 46, no. 7, pp. 077006-1–077006-13, Jul.

Chakravarty, S. and Chang, C.-I (2008a), “Band selection for hyperspectral signature coding,” *International Symposium Spectral Sensing Research* (*ISSSR*), Steven Institute of Technology, New Jersey, Jun.

Chakravarty, S. and Chang, C.-I (2008b), “Block truncation signature coding for hyperspectral analysis,” *ISSSR*, San Diego, CA, Aug.

Chan, T.-H., C.-Y. Chi, Y.-M. Huang and W.-K. Ma (2009a), “A convex analysis-based minimum-volume enclosing simplex algorithm for hyperspectral unmxing,” *IEEE Trans. Signal Process.*, vol. 57, no. 11, pp. 4418–4432, Nov.

Chan, T.H., W.-K. Ma, C.-Y. Chi, *et al.* (2009b), “Hyperspectral unmixing from a convex analysis and optimization perspective,” *First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, WHISPERS’09.

Chang, C.-I (1998), “Further results on relationship between spectral unmixing and subspace projection,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 36, no. 3, pp. 1030–1032, May.

Chang, C.-I (1999), “Least squares error theory for linear mixing problems with mixed pixel classification for hyperspectral imagery,” *Recent Research Developments in Optical Engineering*, edited by S.G. Pandalai, Research Signpost, Trivandrum, Kerala, India, vol. 2, pp. 241–268.

Chang, C.-I (2000), “An information theoretic-based approach to spectral variability, similarity and discriminability for hyperspectral image analysis,” *IEEE Trans. Inf. Theory*, vol. 46, no. 5, pp. 1927–1932, Aug.

Chang, C.-I (2002a), “Relationship among orthogonal subspace projection, constrained energy minimization and RX-algorithm,” *SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral and Ultraspectral Imagery VIII*, *SPIE* 4725, Orlando, FL, Apr.

Chang, C.-I (2002b), “Target signature-constrained mixed pixel classification for hyperspectral imagery,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 40, no. 2, pp. 1065–1081, May.

Chang, C.-I (2003a), *Hyperspectral Imaging: Techniques for Spectral Detection and Classification*, Kluwer Academic/Plenum Publishers, Dordrecht, the Netherlands.

Chang, C.-I (2003b), “How to effectively utilize information to design hyperspectral target detection and classification algorithms,” Workshop in honor of Professor David Landgrebe on Advances in Techniques for Analysis of Remotely Sensed Data, NASA Goddard Visitor Center, Washington, DC, Oct.

Chang, C.-I and Q. Du (2004), “Estimation of number of spectrally distinct signal sources in hyperspectral imagery,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 42, no. 3, pp. 608–619, Mar.

Chang, C.-I (2005), “Orthogonal subspace projection revisited: a comprehensive study and analysis,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 43, no. 3, pp. 502–518, Mar.

Chang, C.-I (2006a), “Hand held device detects chemical and biological warfare agents,” 10.1117/2.1200612.0507, ISSN 1818-2559, Dec. http://newsroom.spie.org.x5237.xml.

Chang, C.-I (2006b), “Exploration of virtual dimensionality in hyperspectral image analysis,” *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XII*, SPIE Defense and Security Symposium, Orlando, FL, Apr.

Chang, C.-I, ed., (2006c) *Recent Advances in Hyperspectral Signal and Image Processing*, edited by C.-I Chang, Research Signpost, Trivandrum, Kerala, India.

Chang, C.-I (2006d), “Utility of virtual dimensionality in hyperspectral signal/image processing,” Chapter 1, *Recent Advances in Hyperspectral Signal and Image Processing*, edited by C.-I Chang, Research Signpost, Trivandrum, Kerala, India.

Chang, C.-I, ed. (2007a), *Hyperspectral Data Exploitation: Theory and Applications*, John Wiley & Sons, New York.

Chang, C.-I (2007b), “Overview,” Chapter 1, *Hyperspectral Data Exploitation: Theory and Applications*, edited by C.-I Chang, John Wiley & Sons, New York, pp. 1–16.

Chang, C. -I (2007c), “Information-processed matched filters for hyperspectral target detection and classification,” Chapter 3, *Hyperspectral Data Exploitation: Theory and Applications*, edited by C.-I Chang, John Wiley & Sons, New York, pp. 47–74.

Chang, C.-I (2008a), “Hyperspectral imaging: an emerging technique in remote sensing,” *International Symposium Spectral Sensing Research (ISSSR)*, Steven Institute of Technology, New Jersey, Jun.

Chang, C.-I (2008b), “Unsupervised linear hyperspectral unmixing,” *International Symposium Spectral Sensing Research (ISSSR)*, Steven Institute of Technology, New Jersey, Jun.

Chang, C.-I (2008c), “Three dimensional receiver operating characteristic (3D ROC) analysis for hyperspectral signal detection and estimation,” *International Symposium Spectral Sensing Research (ISSSR)*, Steven Institute of Technology, New Jersey, Jun.

Chang, C.-I (2009a), “Virtual dimensionality for hyperspectral imagery,” Sept, SPIE Newsroom [DOI: 10.1117/2.1200909.1749].

Chang, C.-I (2009b), “Further results on relationship between spectral unmixing and subspace projection,” *IEEE Trans. Geosci. Rem. Sen.*, vol. 47, no. 11, pp. 4418–4432, Nov.

Chang, C.-I (2010), “Multiple-parameter receiver operating characteristic analysis for signal detection and classification,” *IEEE Sensors J.*, vol. 10, no. 3, pp. 423–442, Mar (invited paper).

Chang, C.-I (2013), *Real Time Hyperspectral Image Processing: Algorithm Architecture and Implementation*, Springer-Verlag, Berlin.

Chang, C.-I and C. Brumbley (1997), “An orthogonalization target signature space projection approach to image classification in unknown background,” *31st Conference Information Sciences and Systems*, The Johns Hopkins University, Baltimore, MD, pp. 174–178, Mar.

Chang, C.-I and C. Brumbley (1999a), “A Kalman filtering approach to multispectral image classification and detection of changes in signature abundance,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 37, no. 1, pp. 257–268, Jan.

Chang, C.-I and C. Brumbley (1999b), “Linear unmixing Kalman filtering approach to signature abundance detection, signature estimation and subpixel classification for remotely sensed images,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 37, no. 1, pp. 319–330, Jan.

Chang, C.-I, S. Chakravarty, H. Chen and Y.C. Ouyang (2009), “Spectral derivative feature coding for hyperspectral signature,” *Pattern Recognit.*, vol. 42, no. 3, pp. 395–408, Mar.

Chang, C.-I, S. Chakravarty and C.-S. Lo (2010), “Spectral feature probabilistic coding for hyperspectral signatures,” *IEEE Sensors J.*, vol. 10, no. 3, pp. 395–409, Mar.

Chang, C.-J., C.-I Chang and M.-L. Chang (1993), “Subband multistage predictive coding,” *Proceedings of the International Conference on Signal Processing’93/ Beijing*, Beijing, China, pp. 783–787, Oct.

Chang, C.-I, Y. Cheng and M.L.G. Althouse (1992), “Chemical vapor detection using multistage predictive coding,” *Proc. Scientific Conference on Chemical Defense Research*, CRDEC, Aberdeen Proving Ground, MD, pp. 909–915, Nov.

Chang, C.-I, Y. Cheng, M.L.G. Althouse, L. Zhang and J. Wang (1992), “Multistage image coding: a top-down gray-level triangle method,” *Proc. International Symposium on Spectral Sensing Research (ISSSR)*, Kauai, Hawaii, pp. 497–511, Sept.

Chang, C.-I and S.-S. Chiang (2002), “Anomaly detection and classification for hyperspectral imagery,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 40, no. 2, pp. 1314–1325, Feb.

Chang, C.-I, S.-S. Chiang and I.W. Ginsberg (2001b), “Anomaly detection in hyperspectral imagery,” *SPIE Conference on Geo-Spatial Image and Data Exploitation II*, Orlando, FL, pp. 43-50, Apr.

Chang, C.-I, S.S. Chiang, J.A. Smith and I.W. Ginsberg (2002), “Linear spectral random mixture analysis for hyperspectral imagery,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 40, no. 2, pp. 375–392, Feb.

Chang, C.-I and Q. Du (1999a), “A noise subspace projection approach to determination of intrinsic dimensionality for hyperspectral imagery,” *EOS/SPIE Symp. Rem. Sens., Conference on Image Signal Process. Rem. Sens. V*, *SPIE* vol. 3871, pp. 34–44, Sept.

Chang, C.-I and Q. Du (1999b), “Interference and noise adjusted principal components analysis,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 37, no. 5, pp. 2387–2396, Sept.

Chang, C.-I and Q. Du (2004), “Estimation of number of spectrally distinct signal sources in hyperspectral imagery,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 42, no. 3, pp. 608–619, Mar.

Chang, C.-I, Q. Du, S.-S. Chiang, D. Heinz and I.W. Ginsberg (2001c), “Unsupervised subpixel target detection in hyperspectral imagery,” *SPIE Conference on Algorithms for Multispectral, Hyperspectral and Ultraspectral Imagery VII*, Orlando, FL, pp. 370–379, Apr.

Chang, C.-I, Q. Du, T.S. Sun and M.L.G. Althouse (1999), “A joint band prioritization and band decorrelation approach to band selection for hyperspectral image classification,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 37, no. 6, pp. 2631–2641, Nov.

Chang, C.-I, Y. Du, J. Wang, S.-M. Guo and P. Thouin (2006), “A survey and comparative study of entropic and relative entropic thresholding techniques,” *IEE Proc., Vision, Image Signal Process.*, vol. 153, no. 6, Dec.

Chang, C.-I and D. Heinz (2000a), “Constrained subpixel target detection for hyperspectral imagery,” *SPIE Conference on Signal and Data Process. Small Targets 2000*, Orlando, FL, pp. 35–45, Apr.

Chang, C.-I and D. Heinz (2000b), “Constrained subpixel detection for remotely sensed images,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 38, no. 3, pp. 1144–1159, May.

Chang, C.-I and M. Hsueh (2006), “Characterization of anomaly detection for hyperspectral imagery,” *Sensor Rev.*, vol. 26, no. 2, pp. 137–146.

Chang, C.-I and B. Ji (2006a), “Weighted least squares error approaches to abundance-constrained linear spectral mixture analysis,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 44, no. 2, pp. 378–388, Feb. 2006.

Chang, C.-I and B. Ji (2006b), “Fisher’s linear spectral mixture analysis,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 44, no. 8, pp. 2292–2304, Aug.

Chang, C.-I, X. Jiao, Y. Du and M.-L. Chang (2010), “Unsupervised hyperspectral target analysis,” *EURASIP J. Adv. Signal Process.*, vol. 2010, no. 485151, p. 20.

Chang, C.-I, X. Jiao, Y. Du and H.M. Chen (2011), “Component-based unsupervised linear spectral mixture analysis for hyperspectral imagery,” *IEEE Trans. Geosc. and Rem. Sens.,* vol. 49, no. 11, pp. 4123–4137, Nov.

Chang, C.-I, W. Liu and C.-C. Chang (2004), “Discrimination and identification for subpixel targets in hyperspectral imagery," *IEEE Int. Conf. on Image Process.*, Singapore, Oct.

Chang, C.-I, J.-M. Liu, B.-C. Chieu, C.-M. Wang, C. S. Lo, P.-C. Chung, H. Ren, C.-W. Yang and D.-J. Ma (2000), "A generalized constrained energy minimization approach to subpixel target detection for multispectral imagery," *Opt. Eng.*, vol. 39, no. 5, pp. 1275–1281, May.

Chang, C.-I and A. Plaza (2006), "Fast iterative algorithm for implementation of pixel purity index," *IEEE Trans. Geosci. Rem. Sens. Lett.*, vol. 3, no. 1, pp. 63–67, Jan.

Chang, C.-I, B. Ramakrishna, J. Wang and A. Plaza, "Exploitation-based hyperspectral image compression," *J. Appl. Rem. Sens.*, vol. 4, 041760 (Dec 03, 2010); doi:10.1117/1.3530429.

Chang, C.-I and H. Ren (1999a), *Computer-Assisted Target Detection and Classification for Hyperspectral Imagery*, Jun.

Chang, C.-I and H. Ren (1999b), "Linearly constrained minimum variance beamforming for target detection and classification in hyperspectral imagery," *IEEE 1999 Int. Geosci. Rem. Sens. Symp.*, Hamburg, Germany, pp. 1241–1243, 28 Jun.–2 Jul.

Chang, C.-I and H. Ren (2000a), "An experiment-based quantitative and comparative analysis of hyperspectral target detection and image classification algorithms," *IEEE Trans Geosci. Rem. Sens.*, vol. 38, no. 2, pp. 1044–1063, Mar.

Chang, C.-I and H. Ren (2000b), *Linearly Constrained Minimum Variance Beamforming Methods for Remote Sensing Image Analysis*, Disclosure of Invention, University of Maryland, Baltimore County, May.

Chang, C.-I, H. Ren, C.-C. Chang, J.O. Jensen and F. D'Amico (2004), "Estimation of subpixel target size for remotely sensed imagery," *IEEE Trans. Geosci. Rem. Sens.*, vol. 42, no. 6, pp. 1309–1320, Jun.

Chang, Y.C., H. Ren, C.-I Chang and B. Rand (2008), "How to design synthetic images to validate and evaluate hyperspectral imaging algorithms," *SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIV*, Orlando, FL, Mar.

Chang, C.-I, H. Ren and S.S. Chiang (2001a), "Real-time processing algorithms for target detection and classification in hyperspectral imagery," *IEEE Trans. Geosci. Rem. Sens.*, vol. 39, no. 4, pp. 760–768, Apr.

Chang, C.-I, H. Ren, Q. Du, S.-S. Chiang and A. Ifarragurri (2001b), "An ROC analysis for subpixel detection," *IEEE Int. Geosci. Remote Sens. Symp.*, Sydney, Australia, Jul.

Chang, C.-I and H. Safavi (2011), "Progressive dimensionality reduction by transform for hyperspectral image analysis," *Pattern Recognit.*, Apr., online access: doi: 10.1016/j.patcog.2011.03.030.

Chang, C.-I, T.-L.E. Sun and M.L.G. Althouse (1998a), "An unsupervised interference rejection approach to target detection and classification for hyperspectral imagery," *Opt. Eng.*, vol. 37, no. 3, pp. 735–743, Mar.

Chang, C.-I and S. Wang (2006), "Constrained band selectionfor hyperspectral imagery," *IEEE Trans. Geosci. Rem. Sens.*, vol. 44, no. 6, pp. 1575–1585, Jun.

Chang, C.-I and Jian Wang (2008), *Real-Time Implementation of Field Programmable Gate Arrays (FPGA) Design in Hyperspectral Imagery*, US Patent, number 7,366,326, Apr.

Chang, C.-I, Jing Wang, C.-C. Chang and C. Lin (2006), "Progressive coding for hyperspectral signature characterization," *Opt. Eng.*, vol. 45, no. 9, pp. 097002-1–097002-15, Sept.

Chang, C.-I, Jing Wang, F. D'Amico and J.O. Jensen (2003), "Multistage pulse code modulation for progressive spectral signature coding," *Chem. Biol. Standoff Detect.–Opt. Technol. Ind. Environ. Sensing Symp.–PhotonicsWest* 2003, pp. 252–261, Oct.

Chang, C.-I, S. Wang, K.H. Liu and C. Lin (2011a), "Progressive band dimensionality expansion and reduction for hyperspectral imagery," *IEEE J. Sel. Top. Appl. Earth Observ. Rem. Sens.*, vol. 4, no. 3, pp. 591–614, Sep.

Chang, C.-I, C.-C. Wu and H.M. Chen (2010), "Random pixel purity index algorithm," *IEEE Trans. Geosci. Rem. Sens. Lett.*, vol. 7, no. 2, pp. 324–328, Apr.

Chang, C.-I, C.C. Wu, W. Liu and Y.C. Ouyang (2006), "A growing method for simplex-based endmember extraction algorithms," *IEEE Trans. Geosci. Rem. Sens.*, vol. 44, no. 10, pp. 2804–2819, Oct.

Chang, C.-I, C.C. Wu, C.-S. Lo and M.-L. Chang (2010), "Real-time simplex growing algorithms for hyperspecral endmember extraction," *IEEE Trans. Geosc. Rem. Sens.*, vol. 40, no. 4, pp. 1834–1850, April.

Chang, C.-I, C.-C. Wu and C.-T. Tsai (2011b), "Random N-finder algorithm," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 641–656, Mar.

Chang, C.-I and W. Xiong (2010), "High order statistics Harsanyi–Farrand–Chang method for estimation of virtual dimensionality," *SPIE*, vol. 7810, San Diego, CA, Aug.

Chang, C.-I, W. Xiong, H.M. Chen and J.W. Chai (2011c), "Maximum orthogonal subspace projection to estimating number of spectral signal sources for hyperspectral images," *IEEE J. Sel. Top. Signal Processing*, vol. 5, no. 3, pp. 504–520, June.

Chang, C.-I, X. Zhao, M.L.G. Althouse and J.-J. Pan (1998b), "Least squares subspace projection approach to mixed pixel classification in hyperspectral images," *IEEE Trans. Geosci. Rem. Sens.*, vol. 36, no. 3, pp. 898–912, May.

Chaudhry, F., C. Wu, W. Liu, C.-I Chang and A. Plaza (2006), "Pixel purity index-based algorithms for endmember extraction from hyperspectral imagery," Chapter 2, *Recent Advances in Hyperspectral Signal and Image Processing*, edited by C.-I Chang, Research Signpost, Trivandrum, Kerala, India, pp. 29–61.

Chen, C.-T. (1999), *Linear System Theory and Design*, 3rd ed., Oxford University Press, New York.

Chen, S, C.F.N. Cowan and P.M. Grant (1991), "Orthogonal least squares learning algorithm for radial basis function networks," *IEEE Trans. Neural Netw.*, vol. 2, pp. 302–309.

Chen, C.-C., C.-M. Wang, C.-I Chang, C.-W. Yang, J. W. Chai, Y.-N. Chung and P.-C. Chung (2005), Techniques in detection of spectral signatures in MR images and their applications, *Medical Imaging Systems: Technology & Applications*, World Scientific Publishing theme volumes, Singapore.

Cheng, Y. (1993), *Multistage Pulse Code Modulation (MPCM)*, M.S. thesis, Department of Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD.

Cheng, Y. and C.-I Chang (1992), "Multistage DPCM," *Proc. IEEE Workshop on Visual Signal Processing and Communications*, Raleigh, North Carolina, pp. 188–193, Sep. 2–3.

Chiang, S.-S. and C.-I Chang (1999), "Target subpixel detection for hyperspectral imagery using projection pursuit," *EOS/SPIE Symp. Rem. Sens., Conference on Image Signal Process. Rem. Sens. V, SPIE* vol. 3871, pp. 107–115, Sept.

Chiang, S.-S. and C.-I Chang (2001), "Discrimination measures for target classification," *IEEE 2001 Int. Geosci. Remote Sens. Symp.*, Sydney, Australia, Jul.

Chiang, S.-S., C.-I Chang and I.W. Ginsberg (2000), "Unsupervised hyperspectral image analysis using independent components analysis," *IEEE 2000 Int. Geosci. Remote Sens. Symp.*, Hawaii, USA, Jul.

Chiang, S.-S., C.-I Chang and I.W. Ginsberg (2001), "Unsupervised subpixel target detection for hyperspectral images using projection pursuit," *IEEE Trans. Geosci. Rem. Sens.*, vol. 39, no. 7, pp. 1380–1391, Jul.

Choi, H.S., D.R. Haynor and Y. Kim (1991), "Partial volume tissue classification of multichannel magnetic resonance images–a mixel model," *IEEE Trans. Med. Imag.*, vol. 12, no. 3, pp. 566–574, Sept.

Chowdhury, A. and M. S. Alam (2007), "Fast implementation of N-FINDR algorithm for endmember determination in hyperspectral imagery," *Proc. SPIE*, vol. 6565, pp. 656526-1-656526-7.

Christophe, E., D. Leger and C. Mailhes (2005), "Quality criteria benchmark for hyperspectral imagery," *IEEE Trans. Geosci. Rem. Sens.*, vol. 43, no. 9, pp. 2103–2114.

Cichocki, A. and R. Unbehauen (1993), *Neural Networks for Optimization and Signal Processing*, Wiley, New York, pp. 364–367.

Comon, P. (1994), "Independent component analysis, A new concept?," *Signal Process.*, vol. 36, pp. 287–314.

Conese, C. and F. Maselli (1993), "Selection of optimum bands from TM scenes through mutual information analysis," *ISPRS J. Photogramm. Rem. Sens.*, vol. 48, no. 3, pp. 2–11.

Cover, T. and J. Thomas (1991), *Elements of Information Theory*, John Wiley & Sons, New York.

Craig, M.D. (1994), "Minimum-volume transforms for remotely sensed data," *IEEE Trans. Geosci. Rem. Sens.*, vol. 32, no. 3, pp. 542–552.

Crippen, R.E. (1988), "The dangers of understanding the importance of data adjustments in band ratioing," *Int. J. Rem. Sens.*, vol. 9, pp. 767–776.

DARPA Spectral Exploitation Workshop (1996), The Defense Advanced Research Projects Agency, Annapolis, MD, Jul.

Dasgupta, D. and Z. Michalewicz (1997), *Evolutionary Algorithms in Engineering Applications*, Springer-Verlag, Berlin.

Daubechies, I. (1992), *Ten Lectures on Wavelets*, SIAM, Philadelphia, PA.

Dempster, A.P., N.M. Laird and D.B. Rubin (1977), "Maximum likelihood from incomplete data via the EM algorithm," *J. R. Stat. Soc. B*, vol. 39, no. 1, pp. 1–38.

Dennison, P.E. and D.A. Roberts (2003), "Endmember selection for multiple endmember spectral mixture analysis using endmember average RMSE," *Rem. Sens. Environ.*, vol. 87, pp. 123–135.

Dowler, A. and M. Andrews (2011), "On the convergence of N-FINDR and related algorithms: to iterate or not to iterate?," *Geosci.Rem. Sens. Lett.*, vol. 8, no. 1.

Du, Q. (2000), *Topics in Hyperspectral Image Analysis*, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD, May.

Du, Q. and C.-I Chang (1998), "Radial basis function neural networks approach to hyperspectral image classification," *1998 Conference on Information Science and Systems*, Princeton University, Princeton, NJ, pp. 721–726, Mar.

Du, Q. and C. -I Chang (1999), "An interference rejection-based radial basis function neural network approach to hyperspectral image classification," *Int. Joint Conference on Neural Network*, Washington, DC, pp. 2698–2703, Jul.

Du, Q. and C.-I Chang (2000), "A hidden Markov model-based spectral measure for hyperspectral image analysis," *SPIE Conference on Algorithms for Multispectral, Hyperspectral, and Ultraspectral Imagery VI*, Orlando, FL, pp. 375–385, Apr.

Du, Q. and C.-I Chang (2001a), "A linear constrained distance-based discriminant analysis for hyperspectral image classification," *Pattern Recognit.*, vol. 34, no. 2.

Du, Q. and C.-I Chang (2001b), "An interference subspace projection approach to subpixel target detection," *SPIE Conference on Algorithms for Multispectral, Hyperspectral and Ultraspectral Imagery VII*, Orlando, FL, pp. 570–577, Apr.

Du, Q. and C.-I Chang (2004a), "Linear mixture analysis-based compression for hyperspectral image analysis," *IEEE Trans. Geosci. Rem. Sens.*, vol. 42, no. 4, pp. 875–891, Apr.

Du, Q. and C.-I Chang (2004b), "A signal-decomposed and interference-annihilated approach to hyperspectral target detection," *IEEE Trans. Geosci. Rem. Sens.* vol. 42, no. 4, pp. 892–906, Apr.

Du, Y. and C.-I Chang (2007), "Rethinking the effective assessment of biometric systems," 10.1117/2.1200711.0815, http://newsroom.spie.org/x17545.xml.

Du, Y. and C.-I Chang (2008), "3D combination curves for accuracy and performance analysis of positive biometrics identification," *Opt. Lasers Eng.*, vol. 46, no. 6, pp. 477–490, Jun.

Du, Y., C.-I Chang, C.-C. Chang, F. D'Amico and J.O. Jensen (2004), "A new hyperspectral measure for material discrimination and identification," *Opt. Eng.*, vol. 43, no. 8, Aug.

Du, Q., C.-I Chang, D.C. Heinz, M.L.G. Althouse and I.W. Ginsberg (2000), "Hyperspectral image compression for target detection and classification," *IEEE 2000 Int. Geosci. Rem. Sens. Symp.*, Hawaii, USA, July.

Du, Q. and J.E. Fowler (2007), "Hyperspectral image compression using JPEG2000 and principal component analysis," *IEEE Geosci. Rem. Sens. Lett.*, vol. 4, pp. 201–205.

Du, Y., C.-I Chang and P. Thouin (2003), "An automatic system for text detection in single video images," *J. Electron. Imaging*, vol. 12, no. 3, pp. 410–422, Jul.

Du, Y. C.-I Chang and P. Thouin (2004) "Unsupervised thresholding of video images," *Opt. Eng.*, vol. 43, no. 2, pp. 282–289, Feb.

Du, Y., C.-I Chang, C.-C. Chang, F. D'Amico and J.O. Jensen (2004), "A new hyperspectral measure for material discrimination and identification," *Optical Engineering*, vol. 43, no. 8, pp. 1777–1786, August.

Du, Z., M.K. Jeong and S.G. Kong (2007), "Band selection of hyperspectral images for automatic detection of poultry skin tumors," *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 3, pp. 332–339, Jul.

Du, Q., N. Raksuntorn and N.H. Younan (2008a), "Variants of N-FINDR algorithm for endmember extraction," *Proc. SPIE*, vol. 7109, pp. 71090G-1–71090G-8, Sept.

Du, Q., N. Raksuntorn, N.H. Younan and R. King (2008b), "Endmember extraction algorithms for hyperspectral image analysis," *Appl. Opt.*, vol. 47, no. 28, pp. F77–F84, Oct.

Du, Q., H. Ren and C.-I Chang (2003), "A comparative study for orthogonal subspace projection and constrained energy minimization," *IEEE Trans. Geosci. Rem. Sens.*, vol. 41, no. 6, pp. 1525–1529, Jun.

Du, Q., W. Zhu and J. Fowler (2004), "Anomaly-based JPEG2000 compression of hyperspectral imagery," *IEEE Trans. Geosci. Rem. Sens. Lett.*, vol. 5, no. 4, pp. 696–700.

Duda, R.O. and P.E. Hart (1973), *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York.

Eches, O., N. Dobigeon and J.-Y. Tourneret (2010), "Estimating the number of endmembers in hyperspectral images using the normal compositional model and a hierarchical Bayesian algorithm," *IEEE J. Sel. Top. Signal Process.*, vol. 4, no. 3, pp. 582–591, Jun.

Epp, S.S. (1995), *Discrete Mathematics with Applications*, 2nd ed., Brooks/Cole, Pacific Grove, CA.

Fano, R.M. (1961), *Transmission of Information: A Statistical Theory of Communication*, John Wiley & Sons, New York.

Farrand, W. and J.C. Harsanyi (1997), "Mapping the distribution of mine tailing in the coeur d'Alene river valley, Idaho, through the use of constrained energy minimization technique," *Rem. Sens. Environ.*, vol. 59, pp. 64–76.

Filippi, A.M. and R. Archibald (2009), "Support vector machine-based endmember extraction," *IEEE Trans. Geosci. Rem. Sens.*, vol. 47, no. 3, pp. 771–791, Mar.

Fisher, K. and C.-I Chang (2011), "Progressive band selection for satellite hyperspectral data compression and transmission," *J. Appl. Rem. Sens.*

Fowler, J.E. (2000), "QccPack: an open-source software library for quantization, compression, and coding," *Appl. Digit. Image Process. XXIII*, vol. 4115, pp. 249–301.

Friedman, J.H. (1987), "Exploratory projection pursuit," *J. Am. Stat. Assoc.*, vol. 82, pp. 249–266.

Friedman, J.H. and J.W. Tukey (1974), "A projection pursuit algorithm for exploratory data analysis," *IEEE Trans. Comput.*, vol. c-23, no. 9, pp. 881–889.

Frost III, O.L. (1972), "An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, pp. 926–935.

Fukunaga, K. (1982), "Intrinsic dimensionality extraction," *Classification, Pattern Recognition and Reduction of Dimensionality*, Handbook of Statistics, vol. 2, edited by P.R. Krishnaiah and L.N. Kanal, North-Holland Publishing Company, Amsterdam, pp. 347–360.

Fukunaga, K (1990), *Statistical Pattern Recognition*, 2nd ed., Academic Press, New York.

Funk, C.C., J. Theiler, D.A. Roberts and C.C. Borel, (2001), "Clustering to improve matched filter detection of weak gas plumes in hyperspectral thermal imagery," *IEEE Trans. Geosci. Rem. Sens.*, vol. 39, no. 7, pp. 1410–1420, Jul.

Gelb, A.ed. (1974), *Applied Optimal Estimation*, MIT Press, Cambridge, MA.

Gersho, A. and R.M. Gray (1992), *Vector Quantization and Signal Compression*, Kluwer Academics Publishers, New York.

Gillespie, A.R., M.O. Smith, J.B. Adams, S.C. Willis, A.F. Fischer, III and D.E. Sabol (1990), "Interpretation of residual images: spectral mixture analysis of AVIRIS images, Owens valley, California," *Proc. 2nd AVIRIS Workshop*, pp. 243–270.

Girolami, M. (2000), *Self-Organising Neural Networks: Independent Component Analysis and Blind Source Separation*, Springer-Verlag, Berlin.

Goetz, A.F.H. and J.W. Boardman (1989), "Quantitative determination of imaging spectrometer specifications based on spectral mixing models," *Proc. IEEE Int. Geosci. Rem. Sens. Symp.'89*, pp. 1036–1039.

Golderg, D.E. (1989), *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

Golub, G.H. and G.F. Van Loan (1989), *Matrix Computations*, 2nd ed., John Hopkins University Press, Baltimore, MD.

Gonzalez, R.C. and R.E. Woods (2002), *Digital Image Processing*, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ.

Gravel, P., G. Beaudoin and J.A. De Guise (2004), "A method for modeling noise in medical images," *IEEE Trans. Med. Imaging*, vol. 23, no. 10, pp. 1221–1232, Oct.

Green, A.A., M. Berman, P. Switzer and M.D. Craig (1988), "A transformation for ordering multispectral data in terms of image quality with implications for noise removal," *IEEE Trans. Geosci. Rem. Sens.*, vol. 26, no. 1 pp. 65–74, Jan.

Greg, I (2010), "An evaluation of three endmember extraction algorithms, ATGP, ICA-EEA and VCA," 2nd *IEEE GRSS Workshop on Hyperspectral Image and Signal Processing—Evolution in Remote Sensing* (WHISPERS).

Guilfoyle, K. (2003), *Application of Linear and Nonlinear Mixture Models to Hyperspectral Imagery Analysis Using Radial Basis Function Neural Networks*, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, May.

Guilfoyle, K., M.L.G. Althouse and C.-I Chang (2001), "A quantitative and comparative analysis of linear and nonlinear spectral mixture models using radial basis function neural networks," *IEEE Trans. Geosci. Rem. Sens.*, vol. 39, no. 10, pp. 2314–2318, Oct.

Guilfoyle, K., M.L.G. Althouse and C.-I Chang (2002), "Further results on linear and nonlinear mixture models for analyzing hyperspectral imagery," *SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral and Ultraspectral Imagery VIII*, *SPIE* 4725, Orlando, FL, Apr.

Hahn, H.K. and H.O. Peitgen (2000), "The skull striping problem in MRI solved by a single 3D watershed transform," *Proc. MICCAL*, *LNCS 1935*, pp. 134–143.

Hamming, R.W. (1989), *Introduction to Applied Numerical Analysis*, Hemisphere Publishing Corporation, New York.

Hapke, B. (1981), "Bidirection reflectance spectroscopy. *I. theory," J. Geophys. Res.*, vol. 86, pp. 3039–3054.

Harris, G., N.C. Anderson, T. Cizadlo, J.M. Bailey, H.J. Bockholt, V.A. Magnotta and S. Arndt (1999), "Improving tissue classification in MRI: a three-dimensional multispectral discriminant analysis method with automated training class selection," *J. Comput. Assist. Tomogr.*, vol. 23, no. 1, pp. 144–154.

Harsanyi, J.C. (1993), *Detection and Classification of Subpixel Spectral Signatures in Hyperspectral Image Sequences*, Department of Electrical Engineering, University of Maryland, Baltimore County, MD, Aug.

Harsanyi, J.C. and C.-I Chang (1994), "Hyperspectral image classification and dimensionality reduction: an orthogonal subspace projection approach," *IEEE Trans. Geosci. Rem. Sens.*, vol. 32, no. 4, pp. 779–785, Jul.

Harsanyi, J.C., W. Farrand and C.-I Chang (1994a), "Detection of subpixel spectral signatures in hyperspectral image sequences," *Annual Meeting Proc. American Society of Photogrammetry & Remote Sensing*, Reno, NV, USA, pp. 236–247.

Harsanyi, J.C., W. Farrand, J. Hejl and C.-I Chang (1994b), "Automatic identification of spectral endmembers in hyperspectral image sequences," *International Symposium on Spectral Sensing Research'94* (ISSSR), San Diego, pp. 267–277, Jul.

Haskell, K.H. and R.J. Hanson (1981), "An algorithm for linear least squares problems with equality and nonnegativity constraints generalized," *Math. Programm.*, vol. 21, pp. 98–118.

Haykin, S. (1986), *Adaptive Filter Theory*, Chapter 10, Prentice-Hall, Englewood Cliffs, New Jersey.

Haykin, S. (1999), *Neural Networks: A Comprehensive Foundation*, Chapter 6, 2nd ed., Prentice-Hall, Englewood Cliffs, NJ.

Heinz, D. and C.-I Chang (2000), *Constraint Least Squares Linear Spectral Mixture Methods for Material Abundance Estimation, Discrimination, Detection, Classification, Identification, Recognition and Quantification, Data Compression and Noise Estimation in Hyperspectral and Multispectral Imagery*, Feb.

Heinz, D. and C.-I Chang (2001), "Fully constrained least squares linear mixture analysis for material quantification in hyperspectral imagery," *IEEE Trans. Geosci. Rem. Sens.*, vol. 39, no. 3, pp. 529–545, Mar.

Heinz, D., C.-I Chang and M.L.G. Althouse (1997), "Fully constrained least squares-based linear unmixing," *IEEE 1999 Int. Geosci. Rem. Sens. Symp*. pp. 1401–1403, 28 Jun.–2 Jul.

Heute, A.R. (1986), "Separation of soil-plant spectral mixtures by factor analysis," *Rem. Sens. Environ.*, vol. 19, pp. 237–251.

Hofmann, T., B. Scgikkopf and A.J. Smola (2008), "Kernel methods in machine learning," *Ann. Stat.*, vol. 36, no. 3, pp. 1171–1220.

Hong, M.-H., Y.N. Sung and X.Z. Lin (2002), "Texture feature coding method for classification of liver sonography," *Comput. Med. Imag. Graph.*, vol. 26, pp. 33–42.

Horng, M.-H. (2003), "Texture feature coding method for texture classification," *Opt. Eng.*, vol. 42, no. 1, pp. 228–238.

Hsueh, M. (2004), *Adaptive Causal Anomaly Detection*, M.S. thesis, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, Aug.

Hsueh, M. (2007), *Reconfigurable Computing for Algorithms in Hyperspectral Image Processing,* Ph.D. dissertation, Department of Computer Science and Electrical

Engineering, University of Maryland, Baltimore county, Baltimore, MD, Aug.

Hsueh, M. and C.-I Chang (2004), “Adaptive causal anomaly detection for hyperspectral imagery,” *IEEE Int. Geosci. Rem. Sens. Symp.*, Sept.

Hsueh, M. and C.-I Chang (2008) “Field programmable gate arrays for pixel purity index using blocks of skewers for endmember extraction in hyperspectral imagery,” *Int. J. High Perform. Comput. Appl.*, vol. 22, no. 4, pp. 408–423.

Huang, R. and M. He (2005), “Band selection based feature weighting for classification of hyperspectral data,” *IEEE Trans. Geosci. Rem. Sens. Lett.*, vol. 2, no. 2, pp. 156–159, Apr.

Huber, P.J. (1985), “Projection pursuit,” *Ann. Stat.*, vol. 13, no. 2, pp. 435–475.

Hugh, G.F. (1968), “On the mean accuracy of statistical pattern recognizers,” *IEEE Trans. Inf. Theory*, vol. IT-14 pp. 55–63.

HYMSMO (1998), HYperspectral MASINT Support to Military Operations Program.

Hyvarinen, A., J. Karhunen and E. Oja (2001), *Independent Component Analysis*, John Wiley & Sons, New York.

Hyvarinen, A. and E. Oja (1997), “A fast fixed-point for independent component analysis,” *Neural Comput.*, vol. 9, no. 7, pp. 1483–1492.

Hyvarinen, A. and E. Oja (2000), “Independent component analysis: algorithms and applications,” *Neural Netw.*, vol. 13, pp. 411–430.

Ifarraguerri, A. (2000), *Hyperspectral Image Analysis with Convex Cones and Projection Pursuit*, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD, May.

Ifarraguerri, A. and C. -I Chang (1999), “Hyperspectral image segmentation with convex cones,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 37, no. 2, pp. 756–770, Mar.

Ifarraguerri, A. and C.-I Chang (2000), “Multispectral and hyperspectral image analysis with projection pursuit,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 38, no. 6, pp. 2529–2538, Nov.

ISO (2000a), Information Technology–JPEG 2000 Image Coding System–Part 1: Core Coding System, ISO, Geneva, Switzerland.

ISO (2000b), Information Technology–JPEG 2000 Image Coding System–Part 2: Extensions; Final Committee Draft, ISO, Geneva, Switzerland.

Jensen, J. R (1996), *Introductory Digital Image Processing: A Remote Sensing Perspective*, 2nd ed., Prentice-Hall, Upper Saddle River, NJ.

Jenson, S.K. and F.A. Waltz (1979), “Principal components analysis and canonical analysis in remote sensing,” *Proc. Am. Soc. Photogramm. 45th Ann. Meeting*, pp. 337–348.

Ji, B. (2006), *Constrained Linear Spectral Mixture Analysis,* Ph.D. dissertation, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, May.

Ji, B. and C.-I Chang (2006), “Principal components analysis-based endmember extraction algorithms,” *Recent Advances in Hyperspectral Signal and Image Processing*, edited by C.-I Chang, Research Signpost, Trivandrum, Kerala, India, pp. 63–91.

Ji, B., C.-I Chang, J.O. Jensen and J.L. Jensen (2004), “Unsupervised constrained linear Fisher’s discriminant analysis for hyperspectral image classification,” *49th Annual Meeting, SPIE Int. Symp. Opt. Sci. Technol., Imag. Spectrom. IX* (AM105), vol. 5546, pp. 344–353, Aug.

Jia, X. and J.A. Richards (1994), “Efficient maximum likelihood classification for imaging spectrometer data sets,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 32, no. 2, pp. 274–281, Mar.

Jiao, X. (2010), *Unsupervised Hyperspectral Target Detection and Classification,* Ph.D. dissertation, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD, May.

Jimenez, L.O. and D.A. Landgrebe (1999), “Hyperspectral data analysis and supervised feature reduction via projection pursuit,” *IEEE Trans. Geosci. Rem. Sens.*, vol. 37, no. 6, pp. 2653–2667, Nov.

Johnson, P., M. Smith, S. Taylor-George and J. Adams (1983), “A semiempirical method for analysis of the reflectance spectra of binary mineral mixtures,” *J. Geophys. Res.*, vol. 88, pp. 3557–3561.

Johnson, W.R., D. W. Wilson, W. Fink, M. Humayun and G. Bearman (2007), “Snapshot hyperspectral imaging in ophthalmology,” *J. Biomed. Opt.*, vol. 12, no. 1, 014036 -1-014036-7, Jan./Feb.

Jones, M.C. and R. Sibson (1987), “What is projection pursuit,” *J. R. Stat. Soc. Assoc.*, vol. 150, part 1, pp. 1–36.

Juang, B.H. and S. Katagiri (1992), “Discriminative learning for minimum error classification,” *IEEE Trans. Signal Process.*, vol. 40, pp. 3043–3054.

Kaarna, A., P.J. Toivanen and P. Keranen (1980), “Compression of multispectral AVIRIS images,” *Proc. SPIE*, vol. 4725, pp. 588–599.

Kailath, T. (1980), *Linear Systems*, Prentice-Hall, Englewood Cliffs, NJ.

Kakadu software: (2012) A Comprehensive Framework for JPEG2000, www.kakadusoftware.com, Implementation of JPEG2000.

Karhunen, J., E. Oja, L. Wang, R. Vigário and J. Joutsensalo (1997), “A class of neural networks for independent component analysis,” *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 486–504, May.

Katsavounides, I., C.C.J. Kuo and Z. Zhang (1994), “A new initialization technique for generalization Lloyd iteration,” *IEEE Trans. Signal Process. Lett.*, vol. 1, pp. 144–146.

Kelly, E.J. (1986) “An adaptive detection algorithm,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. 22, pp. 115–127, Mar.

Kennedy, W.J., Jr. and J.E. Gentle (1980), *Statistical Computing*, Marcel-Dekker Inc., New York.

Keshava, N. and J.F. Mustard (2002), "Spectral unmixing," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 44–57, Jan.

Kim, B. and D.A. Landgrebe (1991), "Hierarchical classifier design in high-dimensional, numerous class cases," *IEEE Trans. Geosci. Rem. Sens.*, vol. 29, no. 4, pp. 792–800, Jul.

Kim, B.-J., Z. Xiong and W.A. Pearlman (2000), "Low bitrate scalable video coding with 3-D set partitioning in hierarchical trees (3-D SPIHT)," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 8, pp. 1374–1387, Dec.

Klauschen, F., A. Goldman, V. Barra, A. Meyer-Lindenberg and A. Lundervold (2009), "Evaluation of automatic brain MR image segmentation and volumetry methods," *Human Brain Mapping*, vol. 30, pp. 1310–1327.

Kosaka, N., K. Uto and Y. Kosugi (2005), "ICA-aided mixed-pixel analysis of hyperspectral data in agricultural land," *IEEE Geosci. and Rem. Sens. Lett.*, vol. 2, no. 2, pp. 220–224, April.

Kraut, S. and L. Scharf (1999), "The CFAR adaptive subspace detector is a scale-invariant GLRT," *IEEE Trans. Signal Processing*, vol. 47, no. 9, pp. 2538–2541.

Kraut, S., L. Scharf and L.T. McWhorter (2001), "Adaptive subspace detector," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 3005–3014.

Kullback, S. (1968), *Information Theory and Statistics*, John Wiley & Sons, New York.

Kuybeda, O., D. Malah and M. Barzohar (2007), "Rank estimation and redundancy reduction of high-dimensional noisy signals with preservation of rare vectors," *IEEE Trans. Signal Process.*, vol. 55, no. 12, pp. 5579–5592.

Kwan, C., B. Ayhan, G. Chen, J. Wang, B. Ji and C.-I Chang (2006), "A novel approach for spectral unmixing, classification, and concentration estimation of chemical and biological Agents," *IEEE Trans. Geosci. Rem. Sens.*, vol. 44, no. 2, pp. 409–419, Feb.

Kwon, H., S.Z. Der and N.M. Nasrabadi (2003), "Adaptive anomaly detection using subspace separation for hyperspectral imagery," *Opt. Eng.*, vol. 42, no. 11, pp. 3342–3351, Nov.

Kwon, H. and N.M. Nasrabadi (2005a), "Kernel RX-algorithm: a nonlinear anomaly detector for hyperspectral imagery," *IEEE Trans. Geosci. Rem. Sens.*, vol. 43, no. 2, pp. 388–397, Feb.

Kwon, H. and N.M. Nasrabadi (2005b), "Kernel orthogonal subspace projection for hyperspectral signal classification," *IEEE Trans. Geosci. Rem. Sens.*, vol. 43, no. 12, pp. 2952–2962, Dec.

Laidlaw, D., K.W. Fleischer and A.H. Barr (1998), "Partial-volume Bayesian classification of material mixtures in MR volume data using volex histograms," *IEEE Trans. Med. Imag.*, vol. 17, no. 1, pp. 74–86, Feb.

Landgrebe, D.A. (1998), *Multispectral Data Analysis: A Signal Theory Perspective*, http://www.ece.purdue.edu/∼biehl/MultiSpec/documentation.html.

Landgrebe, D.A. (2003), *Signal Theory Methods in Multispectral Remote Sens.*, John Wiley & Sons, New York.

Langdon, G.G. and J.J. Rissanen (1981), "Compression of black-white image with arithmetic coding," *IEEE Trans. Commun.*, vol. 29, no. 6, pp. 858–867.

Lavenier, D., J. Theiler, J. Szymanski, M. Gokhle and J. Frigo (2000). "FPGA implementation of pixel purity index method," *Proc. SPIE*, vol. 4212, pp. 30–41.

Lawson, C.L. and R.J. Hanson (1995), *Solving Least Squares Problems*, CAM, vol. 15, SIAM, Philadelphia, PA.

Leadbetter, M. (1987), *Extremes and Related Properties of Random Sequences and Processes*, Springer-Verlag, New York.

Lee, T.W. (1998), *Independent Component Analysis: Theory and Applications*, Kluwer Academic Publishers, Dordrecht, the Netherlands.

Lee, C. and D.A. Landgrebe (1993a), "Feature extraction based on decision boundaries," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 15, no. 4, pp. 388–400, Apr.

Lee, C. and D.A. Landgrebe (1993b), "Analyzing high-dimensional multispectral data," *IEEE Trans. Geosci. Rem. Sens.*, vol. 31, no. 4, pp. 792–800, Jul.

Lee, T.W., M.S. Lewicki and M. Girolami (1999), "Bind source separation of more sources than mixtures using overcomplete representations," *IEEE Trans. Signal Process.*, vol. 6, no. 4, pp. 87–90, Apr.

Lee, D.D. and N.S. Seung (1999), "Learning the parts of objects by non-negative matrix factorization," *Science*, vol. 401, no. 21, pp. 788–791, Oct.

Lee, J.B., A.S. Woodyatt and M. Berman (1990), "Enhancement of high spectral resolution remote sensing data by a noise-adjusted principal components transform," *IEEE Trans. Geosci. Rem. Sens.*, vol. 28, no. 3, pp. 295–304, May.

Leemput, K.V., F. Maes, D. Vandermeulen and P. Suetens (2003), "A unifying framework for partial volume segmentation of brain MR images," *IEEE Trans. Med. Imaging*, vol. 22, no. 1, pp. 105–119, Jan.

Liew, A.W.C. and H. Yan (2003), "An adaptive spatial fuzzy clustering algorithm for 3-D MR image segmentation," *IEEE Trans. Med. Imag.*, vol. 22, no. 9, pp. 1063–1075, Sept.

Linde, Y., A Buzo and R.M. Gray (1980), "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. 28, pp. 84–95.

Liu, K., E. Wong, C.-I Chang and Y. Du (2009), "Kernel based linear spectral mixture analysis for hyperspectral image classification," *1st IEEE GRSS Workshop on Hyperspectral Image and Signal Processing–Evolution in Remote Sens.*, Grenoble, France, Aug.

Liu, K. (2011), *Progressive Band Prioritization and Selection for Linear Spectral Mixture Analysis*, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, May.

Liu, K. and C.-I Chang (2011), "Dynamic band selection for hyperspectral imagery," *Int. Geosci. Remote Sens. Symposium*, Vancouver, Canada, Jul.

Liu, K.H., E. Wong, E.Y. Du, C.C.C. Chen and C.-I. Chang (2012), "Kernel-based linear spectral mixture analysis," *IEEE Trans. Geosci. Rem. Sens. Lett.*, vol. 9, no. 1, pp. 129–133, Jan.

Liu, W. (2005), *Supervised and Unsupervised Classification for Purude Indian Pines Test Site*, M.S. thesis, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD, Nov.

Liu, W. (2008), *Unsupervised Hyperspectral Target Recognition*, Ph.D. dissertation, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD, Jun.

Liu, W. and C.-I Chang (2004), "A nested spatial window-based approach to target detection for hyperspectral imagery," *IEEE Int. Geosci. Remote Sens. Symposium*, Alaska, Sept.

Liu, W. and C.-I Chang (2008), "Multiple window anomaly detection for hyperspectral imagery," *IEEE Int. Geosci. Rem. Sens. Symp.*, Boston, MA, Jul.

Liu, W., C.-I Chang, S. Wang, J. Jensen, H. Hnapp, R. Daniel and R. Yin (2005), "3D ROC analysis for detection software used in water monitoring," *OpticsEast, Chemical and Biological Standoff Detection III* (SA03), Boston, MA, Oct.

Liu, W., C.-C. Wu and C.-I Chang (2007), "An orthogonal subspace projection-based estimation of virtual dimensionality for hyperspectral data exploitation," *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIII, SPIE Defense and Security Symposium*, Orlando, FL, Apr.

Lloyd, S.P. (1982), "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, Mar.

Malinowski, E.R. (1977a), "Theory of error in factor analysis," *Anal. Chem.*, vol. 49, pp. 606–612.

Malinowski, E.R. (1977b), "Determination of the number of factors and experimental error in a data matrix," *Anal. Chem.*, vol. 49, pp. 612–617.

Mallat, S.G. (1989), "A theory for multiresolution signature decomposition: the wavelet representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 11, no. 7, pp. 674–693.

Mallat, S.G. (1999), *A Wavelet Tour of Signal Processing*, 2nd ed., Academic Press, New York.

Manolakis, D. and G. Shaw (2002), "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Process. Mag.*, pp. 29–43, Jan.

Manolakis, D., C. Siracusa and G. Shaw (2001), "Hyperspectral subpixel target detection using the linear mixture model," *IEEE Trans. Geosci. Rem. Sens.*, vol. 39, no. 7, pp. 1392–1409, Jul.

Mausel, P.W., W.J. Kramber and J.K. Lee (1990), "Optimum band selection for supervised classification of multispectral data," *Photogrammetric Eng. Rem. Sens.*, vol. 56, no. 1, pp. 55–60, Jan.

Mavrovouniotis, M.L., A.M. Harper and A. Ifarraguerri, (1994a), "Classification of pyrolysis mass spectra of biological agents using convex cones," *J. Chemometr.*, vol. 8, pp. 305–333.

Mavrovouniotis, M.L., A.M. Harper and A. Ifarraguerri (1994b), "Convex-cone analysis of the time profiles of pyrolysis mass spectra of biological agents," *U.S. Army Edgewood Research, Development and Engineering Center,* Technical Report ERDEC-CR-130, Jul.

Mavrovouniotis, M.L., A.M. Harper and A. Ifarraguerri (1996), "A method for extracting patterns from pyrolysis mass spectra," *Computer Assisted Analytical Spectroscopy*, Wiley, Chichester, UK, pp. 189–240.

Max, J. (1960), "Quantizing for minimum distortion," *IRE Trans. Inf. Theory*, vol. IT-6 pp. 7–12, Mar.

Mayer, R., F. Bucholtz, E. Allen, D. L. von Berg and M. Kruer (2005), "A metric for background candidate assessment for spectral target signature transforms," *IEEE Trans. Geosci. Rem. Sens. Lett.*, vol. 2, pp. 113–117.

Mayer, R., F. Bucholtz and D. Scribner (2003), "Object detection by using whitening/dewhitening to transform target signatures in hyper-and multi-spectral imagery," *IEEE Trans. Geosci. Rem. Sens.*, vol. 41, pp. 1136–1142.

Mayer, R., F. Bucholtz, D. Scribner and M. Kruer (2004), "A family of spectral target signature transforms: relationship to the past, new transforms and sensitivity tests," *IEEE Trans. Geosci. Rem. Sens. Lett.*, vol. 1, pp. 26–30.

Mazer, A.S., M. Martin, M. Lee and J.E. Solomon (1988), "Image processing software for imaging spectrometry data analysis," *Rem. Sens. Environ.*, vol. 24, pp. 201–210.

Metz, C.E. (1978), "ROC methodology in radiological imaging," *Invest. Radiol.*, vol. 21, pp. 720–723.

Miao, L. and H. Qi (2007), "Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization," *IEEE Trans. Geosci. Rem. Sens.*, vol. 45, no. 3, pp. 765–777, Mar.

Michalewicz, Z. (1996), *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd. rev. and extended, Springer-Verlag, Berlin.

Mika, S., G. Ratsch and J. Weston (1999), "Fisher discriminant analysis with kernels," *Neural Networks for Signal Processing IX, Proc. of 1999 IEEE Signal Processing Workshop*, pp. 41–48.

Miller, J.W.V., J.B. Farison and Y. Shin (1992), "Spatially invariant image sequences," *IEEE Trans. Image Process.*, vol. 1, pp. 148–161.

Moon, T.K. and W.C. Stirling (2000), *Mathematical Methods and Algorithms for Signal Processing*, Prentice-Hall.

Murata, N., S. Yoshizawa and S. Amari (1994), "Network information criterion-determining the number of hidden units for an artificial neural network model," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 865–871, Nov.

Mustard, J.F. and C.M. Pieters (1987), "Abundance and distribution of ultramafic microbreccia in moses rock dike: quantitative application of mapping spectroscopy," *J. Geophys. Res.*, vol. 92, no. B10, pp. 10376–10390.

Narozny, M., M. Barret and D.T. Pham (2008), "ICA based algorithms for computing optimal 1-D linear block transforms in variable high-rate source coding," *Signal Process.*, vol. 88, no. 2, pp. 268–283.

Nascimento, J.M.P. and J.M. Dias (2005), "Vertex component analysis: a fast algorithm to unmix hyperspectral data," *IEEE Trans. Geosci. Rem. Sens.*, vol. 43, no. 4, pp. 898–910, Apr.

Neville, R.A., K. Staenz, T. Szeredi, J. Lefebvre and P. Hauff (1999), "Automatic endmember extraction from hyperspectral data for mineral exploration," *Proc. 4th International Airborne Remote Sensing Conference and Exhibition/21st Canadian Symposium on Remote Sensing*, Ottawa, Ontario, Canada, pp. 21–24, Jun.

Oja, E. (1992), "Principal components, minor components, and linear neural networks," *Neural Netw.*, vol. 5, pp. 927–935.

Oja, E., J. Karhunen, L. Wang and R. Vigário (1995), "Principal and independent components in neural networks–recent developments," *Proc. VII Italian Workshop Neural Networks WIRN'95*, Vietrisul Mare, Italy, pp. 16–35, May.

Oja, E. and M. Plumbley (2004), "Blind separation of positive sources by globally convergent gradient search," *Neural Comput.*, vol. 16, pp. 1811–1825.

Otsu, N. (1979), "A threshold selection method from gray-level histograms," *IEEE Trans. System Man Cybernet.*, vol. SMC-9, no. 1, pp. 62–66.

Ouyang, Y.C., H.M. Chen, J.W. Chai, C.C.C. Chen, S.K. Poon, C.W. Yang, S.K. Lee and C.-I Chang (2008a), "Band expansion-based over-complete independent component analysis for magnetic resonance image analysis," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 6, pp. 1666–11677, Jun.

Ouyang, Y.C., H.M. Chen, J.W. Chai, C.C.C. Chen, S.K. Poon, C.W. Yang and S.K. Lee (2008b), "Independent component analysis for magnetic resonance image analysis," *EURASIP Advances in Signal Process.*

Pal, N.R. and S.K. Pal (1989), "Entropic thresholding," *Signal Process.*, vol. 16, pp. 97–108.

Palnitkar, S. (2003), *Verilog HDL: A Guide to Digital Design and Synthesis*, SunSoft Press, Upper Saddle River, NJ.

Parker, D.R., S. C. Gustafson and T.D. Ross (2005a), "Bayesian confidence intervals for ROC curves," *Electron. Lett.*, vol. 41, no. 5, Mar.

Parker, D.R., S.C. Gustafson and T.D. Ross (2005b), "Receiver operating characteristic and confidence error metrics for assessing the performance of automatic target recognition systems," *Opt. Eng.*, vol. 44, no. 9, pp. 097202-1–097202-9.

Pauca, V.P., J. Piper and R.J. Plemmons (2006), "Nonnegative matrix factorization for spectral data analysis," *Linear Alg. Appl.*, vol. 416, no. 1, pp. 29–47, Jul.

Penna, B., T. Tillo, E. Magli and G. Olmo (2006), "Progressive 3-D coding of hyperspectral images based on JPEG2000," *IEEE Trans. Geosci. Rem. Sens. Lett.*, vol. 3, no. 1, pp. 125–129.

Penna, B., T. Tillo, E. Magli and G. Olmo (2007), "Transform coding techniques for lossy hyperspectral data compression," *IEEE Trans. Geosci. Rem. Sens.*, vol. 45, no. 5, pp. 1408–1421.

Pesses, M.E. (1999), "A least squares-filter vector hybrid approach to hyperspectral subpixel demixing," *IEEE Trans. Geosci. Rem. Sens.*, vol. 37, no. 2, pp. 846–849, Mar.

Pham, D.L. and J.L. Prince (1999), "Adaptive fuzzy segmentation of magnetic resonance images," *IEEE Trans. Med. Imag.*, vol. 18, no. 9, pp. 737–752, Sept.

Pickering, M.R. and M.J. Ryan (2001), "Efficient spectral-spatial compression of hyperspectral data," *IEEE Trans. Geosci. Rem. Sens.*, vol. 39, no. 7, pp. 1536–1539.

Pinzon, J.E., S.L. Ustin, C.M. Castaneda and M.O. Smith (1998), "Investigation of leaf biochemistry by hierarchical foreground/background analysis," *IEEE Trans. Geosci. Rem. Sens.*, vol. 36, no. 6, 1913–1927.

Plano, S. and E. Blasch (2003), "User fusion to constrain SAR targeting for TSTs," *SPIE 03*, vol. 5095, Apr.

Plaza, A. and C.-I Chang (2005), "An improved N-FINDR algorithm in implementation," *Conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI, SPIE Symposium on Defense and Security*, SPIE vol. 5806, Orlando, FL, 28 Mar.–1 Apr.

Plaza, A. and C.-I Chang (2006), "Impact of initialization on design of endmember extraction algorithms," *IEEE Trans. Geosci. Rem. Sens.*, vol. 44, no. 11, pp. 3397–3407, Nov.

Plaza, A. and C.-I Chang, ed. (2007a), *High Performance Computing in Remote Sens.*, CRC Press, Boca Raton, FL.

Plaza, A. and C.-I Chang (2007b), "Specific issues about high-performance computing in remote sensing, non-literal analysis versus image-based processing," Chapter 1, *High-Performance Computing in Remote Sensing*, edited by A. Plaza and C.-I Chang, CRC Press, Boca Raton, FL.

Plaza, A. and C.-I Chang (2007c), "Clusters versus FPGAs for real-time processing of hyperspectral imagery," *Int. J. High Perform. Comput. Appl.*, Dec.

Plaza, A., P. Martínez, R. Pérez and J. Plaza (2004), "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Trans. Geosci. Rem. Sens.*, vol. 42, no. 3, pp. 650–663.

Plaza, A.J., D. Valencia, C.-I Chang and J. Plaza (2006), "Parallel implementation of endmember extraction algorithms from hyperspectral data," *IEEE Trans. Geosci. Rem. Sens. Lett.*, vol. 3, no. 7, pp. 334–338, Jul.

Poor, H.V. (1994), *An Introduction to Detection and Estimation Theory*, 2nd. ed., Springer-Verlag, New York.

Pudil, P., J. Novovicova and J. Kittler (1994), "Floating search methods in feature selection," *Pattern Recogn. Lett.*, vol. 15, pp. 1119–1125.

Qian, S.-E. (2004), "Hyperspectral data compression using fast vector quantization algorithm," *IEEE Trans. Geosci. Rem. Sens.*, vol. 42, no. 8, pp. 65–74.

Qian, E., A.B. Hollinger, D. Williams and D. Manak (1996), "Fast three-dimensional data compression of hyperspectral imagery using vector quantization with spectral-feature-based binary coding," *Opt. Eng.*, vol. 35, no. 11, pp. 3242–3249.

Rabiner, L. and B.-H. Juamg (1993), *Fundamentals of Speech Recognition*, Prentice-Hall, Englewood Cliffs, NJ.

Ramakrishna, B. (2004), *Principal Components Analysis (PCA)-Based Spectral/Spatial Hyperspectral Image Compression,* M.S. thesis, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD.

Ramakishna, B., A. Plaza, C.-I Chang, H. Ren, Q. Du and C.-C. Chang (2005b), *Spectral/Spatial Hyperspectral Image Compression, Hyperspectral Data Compression*, edited by G. Motta and J. Storer, Springer-Verlag, Berlin, pp. 309–347.

Ramakrishna, B., J. Wang, A. Plaza and C.-I Chang (2005a), "Spectral/spatial hyperspectral image compression in conjunction with virtual dimensionality," *Conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XI, SPIE Symposium on Defense and Security*, *SPIE* vol. 5806, Orlando, FL, Mar. 28-1 Apr.

Ramey, N.I. and M. Scoumekh (2006), "Hyperspectral anomaly detection within the signal subspace," *IEEE Trans. Geosci. Rem. Sens. Lett.*, vol. 3, no. 3, pp. 312–316, Jul.

Reed, I.S., J.D. Mallett and L.E. Brennan (1974), "Rapid convergence rate in adaptive arrays," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 10, pp. 853–863.

Reed, M. and B. Simon (1972), *Methods of Modern Mathematical Physics I: Functional Analysis*, Academic Press, New York.

Reed, I.S. and X. Yu (1990), "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution," *IEEE Trans. Acoustic, Speech Signal Process.*, vol. 38, no. 10, pp. 1760–1770, Oct.

Ren, H. (1998), *A Comparative Study of Mixed Pixel Classification Versus Pure Pixel Classification for Multi/Hyperspectral Imagery*, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD, May.

Ren, H. (2000), *Unsupervised and Generalized Orthogonal Subspace Projection and Constrained Energy Minimization for Target Detection and Classification in Remotely Sensed Imagery*, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD, May.

Ren, H. and C.-I Chang (1998), "A computer-aided detection and classification method for concealed targets in hyperspectral imagery," *IEEE 1998 Int. Geosci. Rem. Sens. Symp.*, Seattle, WA, pp. 1016–1018, Jul.

Ren, H. and C.-I Chang (1999), "A constrained least squares approach to hyperspectral image classification," *1999 Conference on Information Science and Systems*, Johns Hopkins University, Baltimore, MD, pp. 551-556, Mar.

Ren, H. and C.-I Chang (2000a), "A generalized orthogonal subspace projection approach to unsupervised multispectral image classification," *IEEE Trans. Geosci. Rem. Sens.*, vol. 38, no. 6, pp. 2515–2528, Nov.

Ren, H. and C.-I Chang (2000b), "Target-constrained interference-minimized approach to subpixel target detection for hyperspectral imagery," *Opt. Eng.*, vol. 39, no. 12, pp. 3138–3145, Dec.

Ren, H. and C.-I Chang (2003), "Automatic spectral target recognition in hyperspectral imagery," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1232–1249, Oct.

Ren, H., Q. Du, J. Wang, C.-I Chang and J. Jensen (2006), "Automatic target recognition hyperspectral imagery using high order statistics," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 42, no. 4, pp. 1372–1385, Oct.

Research Systems (2001), Inc., ENVI User's Guide. Research Systems, Inc, Boulder, CO.

Resmini, R.S., M.E. Kappus, W.S. Aldrich, J.C. Harsanyi and M. Anderson (1997), "Mineral mapping with HYperspectral Digital Imagery Collection Experiment (HYDICE) sensor data at Cuprite, Nevada, USA," *Int. J. Rem. Sens.*, vol. 18, no. 17, pp. 1553–1570.

Richards, J.A. and X. Jia (1999), *Remote Sensing Digital Image Analysis*, 2nd ed. Springer-Verlag, Berlin.

Rissanen, J.J. (1976) "Generalized Kraft inequality and arithmetic coding," *IBM J. Res. Dev.*, vol. 20, pp. 198–203, May.

Rissanen, J.J. (1978), "Modeling by shortest data description," *Automatica*, vol. 14, pp. 465–471.

Roberts, D.A., M. Gardner, R. Church, S.L. Ustin, G. Scheer and R.O. Green (1998), "Mapping chaparral in the Santa Monica mountains using multiple endmember spectral mixture model," *Rem. Sens. Environ.*, vol. 65, pp. 267–279.

Robinove, C.J. (1982), "Computation with physical values from Landsat digital data," *Photogramm. Eng. Rem. Sens.*, vol. 48, pp. 781–784.

Roger, R.E. (1994), "A fast way to compute the noise-adjusted principal components transform matrix," *IEEE Trans. Geosci. Rem. Sens.*, vol. 32, no. 1, pp. 1194–1196, Nov.

Roger, R.E. (1996), "Principal components transform with simple, automatic noise adjustment," *Int. J. Rem. Sens.*, vol. 17, no. 14, pp. 2719–2727.

Roger, R.E. and J.F. Arnold (1996), "Reliably estimating the noise in AVIRIS hyperspectral imagers," *Int. J. Rem. Sens.*, vol. 17, no. 10, pp. 1951–1962.

Roggea, D.M., B. Rivarda, J. Zhanga, A. Sancheza, J. Harrisb and J. Feng (2007), "Integration of spatial–spectral information for the improved extraction of endmembers," *Rem. Sens. Environ.*, vol. 110, no. 3, pp. 287–303, Oct.

Ruan, S., C. Jaggi, J. Xue, J. Fadili and D. Bloyet (2000), "Brain tissues classification of magnetic resonance images using partial volume modeling," *IEEE Trans. Med. Imag.*, vol. 19, no. 12, pp. 1179–1187, Dec.

Rucker, J.T., J.E. Fowler and N.H. Younan (2005) "JPEG2000 coding strategies for hyperspectral data," *Proc. Int. Geosci. Rem. Sens. Symp.*, vol. 1, pp. 128–131, Jul.

Sabol, D.E., J.B. Adams and M.O. Smith (1992), "Quantitative sub-pixel spectral detection of targets in multispectral images," *J. Geophys. Res.*, vol. 97, pp. 2659–2672.

Safavi, H. (2010), *Hyperspectral Data Dimensionality Reduction and Applications,* Ph.D. dissertation, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, MD, Dec.

Safavi, H. and C.-I Chang (2008), "Projection pursuit-based dimensionality reduction," *SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIV*, Orlando, FL, Mar.

Safavi, H., K. Liu and C.-I Chang (2011), "Dynamic dimensionality reduction for hyperspectral imagery," *SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral and Ultraspectral Imagery XVII*, Orlando, FL, Apr.

Sahoo, P.K., S. Soltani, A.K.C. Wong and Y.C. Chen (1988), "A survey of thresholding techniques," *Comput. Vis. Graph. Image Process. (CVGIP)*, vol. 41, pp. 233–260.

Santago, P. and H.D. Gage (1993), "Quantification of MR brain images by mixture density and partial volume modeling," *IEEE Trans. Med. Imag.*, vol. 10, no. 3, pp. 395–407, Sept.

Schalkoff, R. (1992), *Pattern Recognition: Statistical, Structure and Neural Network*, John Wiley & Sons, New York.

Scharf, L.L. (1991), *Statistical Signal Processing*, Addison-Wesley, Reading, MA.

Scharf, L.L. and B. Friedlander (1994), "Matched subspace detectors,“ *IEEE Trans. Signal Process.*, vol. 42, no. 8, pp. 2146–2157.

Scherrer, B., F. Forbes, C. Garbay and M. Dojat (2009), "Distributed local MRF models for tissue and structure brain segmentation," *IEEE Trans. Med. Imag.*, vol. 28, no. 9, pp. 1278–1295, Aug.

Scholkopf, N. and A.J. Smola (2002), *Learning with Kernels, Support Vector Learning*, MIT Press, Cambridge MA, pp. 327–352.

Scholkopf, N., A.J. Smola and K.-R. Miller (1999a), Kernel principal component analysis. *Advances in Kernel Methods*, edited by N. Scholkopf, C.J.C. Burges and A. J. Smola (1999), *Learning with Kernels, 2002*, MIT Press, Cambridge, MA.

Scholkopf, N., A.J. Smola and K.-R. Miller (1999b), "Kernel principal component analysis," *Neural Comput.*, no. 10, pp. 1299–1319.

Schowengerdt, R.A. (1997), *Remote Sensing: Models and Methods for Image Processing*, 2nd ed., Academic Press, New York.

Schwarz, G. (1978), "Estimating the dimension of a model," *Ann. Stat.*, vol. 6, pp. 461–464.

Sebastiani, G. and P. Barone (1991), "Mathematical principles of basic magnetic resonance image in medicine," *Signal Process.*, vol. 25, pp. 227–250.

Serpico, S.B. and L. Bruzzone (2001), "A new search algorithm for feature selection in hyperspectral remote sensing images," *IEEE Trans. Geosci. Rem. Sens.*, vol. 39, no. 7, pp. 1360–1367, Jul.

Settle, J.J. (1996), "On the relationship between spectral unmixing and subspace projection," *IEEE Trans. Geosci. Rem. Sens.*, vol. 34, no. 4, pp. 1045–1046, Jul.

Settle, J.J. and N.A. Drake (1993), "Linear mixing and estimation of ground cover proportions," *Int. J. Rem. Sens.*, vol. 14, no. 6, pp. 1159–1177.

Shahshahani, B.M. and D.A. Landgrebe (1994), "The effect of unlabeled samples in reducing the small sample size problem and mitigating the Hugh phenomenon," *IEEE Trans. Geosci. Rem. Sens.*, vol. 32, no. 5, pp. 1087–1095, Sept.

Shapiro, J.M. (1993), "Embedded image coding using zero trees of wavelet coefficients," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3445–3462.

Shattuck, D.W., S.R. Sandor-Leahy, K.A. Schaper, D.A. Rotterberg and R.M. Leahy (2001), "Magnetic resonance image tissue classification using a partial volume modeling," *NeuroImaging*, vol. 13, no. 12, pp. 856–876.

Shen, S.S. and B.S. Beard (1998), "Effects of hyperspectral compression on non-literal exploitation," *Proc. SPIE*, vol. 3438, pp. 191–199.

Shimabukuro, Y.E. (1987), *Shade Images Derived from Linear Mixing Models of Multispectral Measurements of Forested Areas,* Ph.D. dissertation, Department of Forest and Wood Science, Colorado State University, Fort Collins.

Shimabukuro, Y.E. and J.A. Smith (1991), "The least-squares mixing models to generate fraction images derived from remote sensing multispectral data," *IEEE Trans. Geosci. Rem. Sens.*, vol. 29, pp. 16–20.

Siadat, M. and H. Soltanian-Zadeh (2007), "Partial volume and distribution estimation from multispectral images using continuous representation," *J. Electron. Imag.*, vol. 16, no. 4, pp. 043001-1–043001-15, Oct.–Dec.

Singer, R.B. and T.B. McCord (1979), "Mars: large scale mixing of bright and dark surface materials and implications for analysis of spectral reflectance," *Proc. Lunar Planet. Sci. Conf. 10th*, pp. 1835–1848.

Singh, A. and A. Harison (1985), "Standardized principal components," *Int. J. Rem. Sens.*, vol. 6, no. 6, pp. 883–896.

Siyal, M.Y. and L. Yu (2005), "An intelligent modified fuzzy c-means based algorithm for bias estimation and segmentation of brain MRI," *Pattern Recognit. Lett.*, vol. 26, pp. 2052–2062.

Smith, M.O., J.B. Adams and D.E. Sabol (1994), "Spectral mixture analysis–new strategies for the analysis of multispectral data," *Image Spectroscopy–A Tool for Environmental Observations*, edited by J. Hill and J. Mergier, ECSC, EEC, EAEC, Brussels and Luxembourg, pp. 125–143.

Smith, M.O., P.E. Johnson and J.B. Adams (1985), "Quantitative determination of mineral types and abundances from reflectance spectra using principal components analysis," *J. Geophys. Res.*, vol. 90, pp. C797–C904.

Smith, M.O., D.A. Roberts, J. Hill, W. Mehl, B. Hosgood, J. Verdebout, G. Schmuck, C. Koechler and J.B. Adams (1994), "A new approach to quantifying abundances of materials in multispectral images," *Proc. IEEE Int. Geosci. Rem. Sens. Symp.'94*, Pasadena, CA, pp. 2372–2374.

Smith, S.M. (2000) "Fast Robust Automated Brain Extraction," *Human Brain Mapping*, vol. 17, pp. 143–155.

Smith, S.M., M., Jenkinson, M.W., Woolrich, C.F., Beckmann, T.E.J., Behrens, P. R., Johansen-Berg, M. D., Bannister, H., Luca, I., Drobnjak, D.E., Flitney, R.K., Niazy, J., Saunders, J., Vickers, Y., Zhang, N.D., Stefano, J.M., Brady and P.M. Matthews (2004), "Advances in functional and structural MR image analysis and implementation as FSL," *NeuroImage*, vol. 23: S208–S219.

Soltanian-Zadeh, H., J.P. Windham and D.J. Peck (1996), "Optimal linear transformation for MRI feature extraction," *IEEE Trans. Med. Imag.*, vol. 15, pp. 749–767.

Stark, H. and J. Woods (1994), *Probability, Random Processes, and Estimation for Engineers*, 3rd ed., Prentice-Hall, Englewood Cliffs, NJ.

Stearns, S.D., B.E. Wilson and J.R. Peterson (1993), "Dimensionality reduction by optimal band selection for pixel classification of hyperspectral imagery," *Appl. Digit. Image Process. XVI, SPIE*, vol. 2028, pp. 118–127.

Stein, D.W., S.G. Beaven, L.E. Hoff, E.M. Winter, A.P. Schaum and A.D. Stocker (2002), "Anomaly detection from hyperspectral imagery," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 58–69, Jan.

Stellman, C.M., G.G. Hazel, F. Bucholtz, J.V. Michalowicz, A. Stocker and W. Scaaf (2000), "Real-time hyperspectral detection and cuing," *Opt. Eng.*, vol. 39, no. 7, pp. 1928–1935, Jul.

Strang, G. and T. Nguyen (1996), *Wavelets and Filter Banks*, Wellesley-Cambridge Press, Wellesley, MA.

Suen, C.Y., M. Berthod and S. Mori (1980), "Automatic recognition of handprinted characters–the state-of-the-art," *Proc. IEEE*, vol. 68, no. 4, pp. 469–487, Apr.

Swets, J.A. and R.M. Pickett (1982), *Evaluation of Diagnostic Systems: Methods from Signal Detection Theory*, Academic Press, New York.

Szu, H. (1999), "Independent component analysis (ICA): an enabling technology for intelligent information/image technology (IIT)," *IEEE Circuit Syst. Mag.*, vol. 10, no. 4, pp. 14–37, Dec.

Tang, X. and W.A. Pearlman (2005), "Three-dimensional wavelet-based compression of hyperspectral images," *Hyperspectral Data Compression*, Kluwer Academic Publishers, Dordrecht, the Netherlands.

Taubman, D. (2000), "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, pp. 1158–1170, Jul.

Taubman, D.S. and M.W. Marcellin (2002), JPEG2000: Image Compression Fundamentals, Standard and Practice. Kluwer, Boston, MA.

Thai, B. and G. Healey (1999), "Invariant subpixel material identification in hyperspectral imagery," SPIE, vol. 3717.

Thai, B. and G. Healey (2002), "Invariant subpixel material detection in hyperspectral imagery," *IEEE Trans. Geosci. Rem. Sens.*, vol. 40, no. 3, pp. 599–608, Mar.

Theiler, J., J. Frigo, M. Gokhle and J.J. Szymanski (2000), "FPGA Implementation of Pixel Purity Index Method," *Proc. SPIE*, vol. 4132.

Theiler, J., D. Lavernier, N.R. Harvey, S.J. Perkins and J.J. Szymanski (2000), "Using blocks of skewers for faster computation of pixel purity index," *Proc. SPIE*, vol. 4132, pp. 61–71.

Therrien, C.W. (1992), *Discrete Random Signals and Statistical Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ.

Tompkins, S., J.F. Mustarrd, C.M. Pieters and D.W. Forsyth (1997), "Optimization of targets for spectral mixture analysis," *Rem. Sens. Environ.*, vol. 59, pp. 472–489.

Tou, J.T. and R.C. Gonzalez (1974), *Pattern Recognition Principles*, Addison-Wesley, Reading, MA, pp. 92–94.

Tu, T.M. (2000), "Unsupervised signature extraction and separation in hyperspectral images: a noise-adjusted fast independent component analysis approach," *Opt. Eng.*, vol. 39, no. 4, pp. 897–906.

Tu, T.M., C.-H. Chen and C.-I Chang (1997), "*A posteriori* least squares orthogonal subspace projection approach to weak signature extraction and detection," *IEEE Trans. Geosci. Rem. Sens.*, vol. 35, no. 1, pp. 127–139, Jan.

Tu, T.M., C.-H. Chen, J-L. Wu and C.-I Chang (1998), "A fast two-stage classification method for high dimensional remote sensing data," *IEEE Trans. Geosci. Rem. Sens.*, vol. 36, no. 1, pp. 182–191, January.

Tu, T.-M., C.H. Lee, C.S. Chiang and C.P. Chang (2001), "A visual disk approach for determining data dimensionality in hyperspectral imagery," *Proc. Natl. Sci. Counc.*, vol. 25, no. 4, pp. 219–231.

Tu, T.M., H.C. Shy, C.-H. Lee and C.-I Chang (1999), "An oblique subspace projection to mixed pixel classification in hyperspectral images," *Pattern Recognit.*, vol. 32, no. 8, pp. 1399–1408, Aug.

Tzou, K.H. (1987), "Progressive image transmission: a review and comparison of techniques," *Opt. Eng.*, vol. 26, no. 7, pp. 581–589.

Valencia, D., A. Plaza, M.A. Vega-Rodriguez and R.M. Perez (2005), "FPGA design and implementation of a fast pixel purity index algorithm for endmember extraction in hyperspectral imagery," *SPIE East*, Boston.

Van Veen, B.D. and K.M. Buckley (1988), "Beamforming: a versatile approach to spatial filtering," *IEEE ASSP Mag.*, pp. 4–24, Apr.

Vannier, M.W., R.L. Butterfield, D. Jordan, W.A. Murphy, R.G. Levitt, and M. Gado (1985), "Multispectral analysis of magnetic resonance images." *Radiology*, vol. 154, pp. 221–224, Jan.

Vapnik, V.N. (1998), *Statistical Learning Theory*, John Wiley & Sons, New York.

Vetterli, M. and J. Kovacevic (1995), *Wavelets and Subband Coding*, Prentice-Hall, Englewood Cliffs, NJ.

Volder, J.E. (1959), "The CORDIC trigonometric computing technique," *IEEE Trans. Electron. Comput.* pp. 330–334.

Wang, C.-M (2002), *Applications of Multispectral Imaging Processing Techniques to Brain Magnetic Resonance Image Classification,* Ph.D. dissertation, Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, Jun.

Wang, J. (2003), *Field Programmable Gate Arrays* (*FPGA*) *Design for Real-time Implementation of Hyperspectral Detection and Classification Algorithms*, Ph.D. dissertation, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, May.

Wang, Jing (2006), *Applications of Independent Component Analysis to Hyperspectral Data Exploitation*, Ph.D. dissertation, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, May.

Wang, S. (2006), *Statistical Signal Processing Applications to Hyperspectral Signature Characterization*, Ph.D. dissertation, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, May.

Wang, Jian and C.-I Chang (2002), "Unsupervised Kalman filter approach to signature estimation for remotely sensed imagery," *Int. Geosci. Rem. Sens. Symp.*, Toronto, Canada, Jun.

Wang, Jian and C.-I Chang (2003), "FPGA design for constrained energy minimization," *Chemical and Biological Standoff Detection–Optical Technologies for Industrial and Environmental Sensing Symposia–Photonics West* 2003, pp. 262–273, Oct.

Wang, Jing and C.-I Chang (2004), "A uniform projection-based unsupervised detection and classification for hyperspectral imagery," *IEEE Int. Geosci. Rem. Sens. Symp.*, Alaska, Sept.

Wang, Jing and C.-I Chang, (2005) "Dimensionality reduction by independent component analysis for hyperspectral image analysis," *IEEE Int. Geosci. Rem. Sens. Symp.*, Seoul, Korea, Jul.

Wang, Jing and C.-I Chang (2006a), "Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis," *IEEE Trans. Geosci. Rem. Sens.*, vol. 44, no. 6, pp. 1586–1600, Jun.

Wang, Jing and C.-I Chang (2006b), "Applications of independent component analysis in endmember extraction and abundance quantification for hyperspectral imagery," *IEEE Trans. Geosci. Rem. Sens.*, vol. 44, no. 9, pp. 2601–2616, Sept.

Wang, J. and C.-I Chang (2007), "FPGA design for real-time implementation of hyperspectral target detection and classification algorithms," *Chapter 16, High-Performance Computing in Remote Sensing, edited by A. Plaza and C. -I Chang*, CRC Press, Boca Raton, FL.

Wang, S. and C. *-I. Chang* (2007), "Variable-number variable-band selection for feature characterization in hyperspectral signatures," *IEEE Trans. Geosci. Rem. Sens.*, vol. 45, no. 9, pp. 2979–2992, Sept.

Wang, Jing, C.-I Chang, C.-C. Chang and C. Lin (2004a), "Binary coding for remotely sensed imagery," 49th *Annual Meeting, SPIE International Symposium on Optical Science and Technology, Imaging Spectrometry IX (AM105)*, Denver, CO, Aug.

Wang, Jing, C.-I Chang, H.-M. Chen, C.C.C. Chen, J.W. Chai and Y.C. Ouyang (2005), "3D ROC analysis for medical diagnosis evaluation," *27th Annual International Conference of IEEE Engineering in Medicine and Biology Society (EMBS)*, Shanghai, China, Sept.

Wang, S., C.-I Chang, J.L. Jensen and J.O. Jensen (2004b), "Spectral abundance fraction estimation of materials using Kalman filters," *Optics East, Chemical and Biological Standoff Detection II* (OE120), Philadelphia, PA, Oct.

Wang C.M., C.C. Chen, Y.-N. Chung, S.-C. Yang, P.C. Chung, C.W. Yang and C.-I Chang (2003), "Detection of spectral signatures in MR images for classification," *IEEE Trans. Med. Imaging*, vol. TMI-22, no. 1, pp. 50–61, Jan.

Wang, C.-M., C.C. Chen, S.-C. Yang, Y.-N. Chung, P.C. Chung, C.W. Yang and C.-I Chang (2002), "An unsupervised orthogonal subspace projection approach to MR image classification MR images for classification," *Opt. Eng.*, vol. 41, no. 7, pp. 1546–1557, Jul.

Wang, L. and M. Goldberg (1988), "Progressive image transmission by transform coefficient residual error quantization," *IEEE Trans. Commun.*, vol. 36, no. 1, pp. 75–87.

Wang, L. and M. Goldberg (1989), "Progressive image transmission using vector quantization on images in pyramid form," *IEEE Trans. Commun.*, vol. 37, no. 12, pp. 1339–1349.

Wang, Y., L. Guo, and N. Liang (2009), "Using a new search strategy to improve the performance of N-FINDR algorithm for endmember determination," *2nd International Congress on Signal and Image Processing*, Tianjin, China.

Wang, S., C.-M. Wang, M.-L. Chang and C.-I Chang (2010), "New applications of Kalman filtering approach to hyperspectral signature estimation, identification and abundance quantification," *IEEE Sens. J.*, vol. 10, no. 3, pp. 547–563, Mar.

Wang, C.-M., S.-C. Yang, P.C. Chung, C.C. Chen, Y.-N. Chung, C.W. Yang and C.-I Chang (2003), "Detection of spectral signatures in MR images for classification," *IEEE Trans. Med. Imag.*, vol. TMI-22, no. 1, pp. 50–61, Jan.

Wang, C.-M., S-C. Yang, P.-C. Chung, C.S. Lo, C.-I Chang, C.C. Chen, C.-W. Yang and C.H. Wen (2001), "Orthogonal subspace projection-based approaches to classification of MR image sequences," *Comput. Med. Imag. Graph.*, vol. 25, no. 6, pp. 465–476, Oct.

Ward, C.R., P.J. Hargrave and J.G. McWhirter (1986), "A novel algorithm and architecture for adaptive digital beamforming," *IEEE Trans. Antennas Propag.*, vol. AP-34, no. 3, pp. 338–346, Mar.

Wax, M. and T. Kailath (1985), "Detection of signals by information criteria," *IEEE Trans. Acoustic Speech Signal Process.*, vol. ASSP-33, no. 2, pp. 387–392, Apr.

Welch, T.A. (1984), "A technique for high-performance data compression," *IEEE Computer*, vol. 17, no. 6, pp. 8–19, Jun.

Wells, W.M. III, W.E.L. Grimson, R. Kikinis and F.A. Jolesz (1996), "Adaptive segmentation of MRI data," *IEEE Trans. Med. Imag.*, vol. 15, no. 4, pp. 429–442, Aug.

Windham, J.P., M.A. Abd-Allah, D.A. Reimann, J.W. Froehich and A.M. Haggar (1988), "Eigneimage filtering in MR imaging,“ *J. Comput. Assist. Tomogr.*, vol. 12, no. 1, pp. 1–9, Jan./Feb.

Winter, M.E. (1999a), "Fast autonomous spectral endmember determination in hyperspectral data," *Proc. of 13th International Conference on Applied Geologic Remote Sensing*, Vancouver, B.C., Canada, vol. II, pp. 337–344.

Winter, M.E. (1999b), "N-finder: an algorithm for fast autonomous spectral endmember determination in hyperspectral data," *Image Spectrometry V*, *Proc. SPIE,* vol. 3753, pp. 266–277.

Winter, M.E. (2004), "A proof of the N-FINDR algorithm for the automated detection of endmembers in a hyperspectral image," *Proc. SPIE*, vol. 5425, pp. 31–41, Apr.

Winter, M.E. and E. Winter (2000), "Comparison of approaches for determining endmember in hyperspectral data," *Proc. IEEE Int. Aerosp. Conf.*, pp. 305–313.

Witten, I.H., R. Neal and J.G. Cleary (1987), "Arithmetic coding for data compression," *Commun. Assoc. Comput. Machinery*, vol. 30, no. 6, pp. 520–540.

Wong, E. (2008), *Linear Spectral Unmixing Approaches to Magnetic Resonance Image Classification,* M.S. thesis, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, Oct.

Wong, E. (2011), *Partial Volume Estimation of Magnetic Resonance Image Using Linear Spectral Mixing Analysis,* Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, May.

Wong, E. and C.-I Chang (2008), "Linear spectral unmixing approaches to magnetic resonance image analysis," *SPIE Conference on Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIV*, Orlando, FL, Mar.

Wright, G.A. (1997) "Magnetic resonance image," *IEEE Signal Process. Mag.*, pp. 56–66, Jan.

Wu, C.-C. (2006), *Exploration of Methods of Estimation on Number of Endmember,* M.S. thesis, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, May.

Wu, C.-C. (2009), *Design and Analysis of Maximum Simplex Volume-Based Endmember Extraction Algorithms,* Ph.D. dissertation, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, May.

Wu, C.-C. and C.-I Chang (2006), "Automatic algorithms for endmember extraction," *SPIE Conf. Imaging Spectrom. XI, SPIE Symp. Opt. Photonics*, vol. 6302, pp. 13-17, Aug.

Wu, C.-C., H.M. Chen and C.-I Chang (2010), "Real-time N-finder processing algorithms," *J. Real-Time Image Processing*, vol. 7, no. 2, pp. 105–129 [DOI: 10.1007/s11554-010-0151-z].

Wu, C.C., S. Chu and C.-I Chang (2008), "Sequential N-FINDR algorithm," *SPIE Conference on Imaging Spectrometry XIII*, San Diego, Aug.

Wu, C.-C., C.S. Lo and C.-I Chang (2009), "Improved process for use of a simplex growing algorithm for endmember extraction," *IEEE Trans. Geosci. Rem. Sens. Lett.*, vol. 6, no. 3, pp. 523–527, Jul.

Wu, H.T., J.F. Yang and F.K. Chen (1995), "Source number estimators using transformed Gerschgorin radii," *IEEE Trans. Signal Process.*, vol. 43, no. 6, pp. 1325–1333, Jun.

XESS Corp., http://www.xess.com.

Xilinx, Inc., http://www.xilinx.com.

Xiong, W. (2011), *Estimation of Effective Spectral Dimensionality for Hyperspectral Imagery*, Department of Computer Science and Electrical Engineering, University of Maryland, Baltimore County, Baltimore, MD, May.

Xiong, W., C.T. Tsai, C.W. Yang and C.-I Chang (2010), "Convex cone-based endmember extraction for hyperspectral imagery," *SPIE*, vol. 7812, San Diego, CA, Aug.

Xiong, W., C.-C. Wu, C.-I Chang, K. Kapalkis and H.M. Chen (2011), "Fast algorithms to implement N-FINDR for hyperspectral endmember extraction," *IEEE J. Sel. Top. Appl. Earth Observ. Rem. Sens.*, vol. 4, no. 3, pp. 545–564, Sept.

Yang, S, J. Wang, C.-I Chang, J.L. Jensen and J.O. Jensen (2004), "Unsupervised image classification for remotely sensed imagery," *49th Annual Meeting, SPIE International Symposium on Optical Science and Technology, Imaging Spectrometry IX* (AM105), Denver, CO, Aug.

Yu, X, I.S. Reed and A.D. Stocker (1993), "Comparative performance analysis of adaptive multispectral detectors," *IEEE Trans. Signal Process.*, vol. 41, no. 8, pp. 2639–2656, Aug.

Zare A. and P. Gader (2007), "Sparsity promoting iterated constrained endmember detection in hyperspectral imagery," *IEEE Geosci. Rem. Sens. Lett.*, vol. 4, no. 3, pp. 446–450, Jul.

Zenzo, S.D., R. Bernstein, S.D. Degloria and H.C. Kolsky (1987a), "Gaussian maximum likelihood and contextual classification algorithms for multicrop classification experiments using thematic mapper and multispectral scanner sensor data," *IEEE Trans. Geosci. Rem. Sens.*, vol. GE-25, no. 6, pp. 815–824, Nov.

Zenzo, S.D., S.D. Degloria, R. Bernstein and H.C. Kolsky (1987b), "Gaussian maximum likelihood and contextual classification algorithms for multicrop classification," *IEEE Trans. Geosci. Rem. Sens.*, vol. GE-25, no. 6, pp. 805–814, Nov.

Zhang, Y. and M.C. Amin (2001), "Array processing for nonstationary interference suppression in DS/SS communications using subspace projection techniques," *IEEE Trans. Signal Process.*, vol. 49, no. 12, pp. 3005–3014.

Zhang, Y., M. Brady, and S. Smith (2001), "Segmentation of brain MR images through a hidden Markov random field model and the expectation maximization algorithm," *IEEE Trans. Med. Imag.*, vol. 20, no. 1, pp. 45–57, Jan.

Zhang, X. and C.H. Chen (2002), "New independent component analysis method using higher order statistics with applications to remote sensing images," *Opt. Eng.*, vol. 41, pp. 1717–1728, Jul.

Zhao, X. (1996), *Subspace Projection Approach to Multispectral/Hyperspectral Image Classification Using Linear Mixture Modeling,* Master thesis, Department of Computer Sciences and Electrical Engineering, University of Maryland Baltimore County, MD, May.

Zortea, M. and A. Plaza (2009), "A quantitative and comparative analysis of different implementations of N-FINDR: a fast endmember extraction algorithm," *IEEE Geosci. Rem. Sens. Lett.*, vol. 6, no. 4, pp. 787–791, Oct.

Ziv, J and A. Lempel (1977), "A universal algorithm for sequential compression," *IEEE Trans. Inform. Theory*, vol. 23, no. 3, pp. 337–343.

Index

Hyperspectral Data Processing: Algorithm Design and Analysis, First Edition. Chein-I Chang.