

HYPERSENSITIVE ENDMEMBER DETECTION AND BAND SELECTION USING
BAYESIAN METHODS

By
ALINA ZARE

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2008

© 2008 Alina Zare

To Mom and Dad

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Paul Gader, for all of his guidance, support and the numerous opportunities he provided me throughout my studies and research. I would also like to thank my committee members, Dr. Jeffery Ho, Dr. Gerhard Ritter, Dr. Clint Slatton, and Dr. Joseph Wilson, for all of their help and valuable suggestions.

Thank you to Miranda Schatten Silvius of NVESD, Dr. Russell Harmon of ARL/ARO, Dr. William Clark of ARO, and Dr. Michael Cathcart of GTRI for their support throughout this research.

Additionally, thank you to my many former and current labmates. I am particularly grateful to Jeremy Bolton and Xuping Zhang for the countless number of times they provided insight and discussion during my studies.

Thank you to my parents, Roobik and Emik Zare, my sister, Anahita Zare, my boyfriend, Michael Black, and all of my family for their never-ending love, support and understanding. Finally, many thanks to my godfather, Artoosh Avanesian, whose influence first led me to computer science.

TABLE OF CONTENTS

| | <u>page</u> |
|---|-------------|
| ACKNOWLEDGMENTS | 4 |
| LIST OF TABLES | 7 |
| LIST OF FIGURES | 8 |
| LIST OF SYMBOLS AND ABBREVIATIONS | 10 |
| ABSTRACT | 12 |
| CHAPTER | |
| 1 INTRODUCTION | 14 |
| 1.1 Hyperspectral Image Data and Analysis | 14 |
| 1.1.1 Endmember Detection | 15 |
| 1.1.2 Hyperspectral Band Selection | 17 |
| 1.2 Statement of Problem | 17 |
| 1.3 Overview of Research | 18 |
| 2 LITERATURE REVIEW | 20 |
| 2.1 Existing Endmember Detection Algorithms | 20 |
| 2.1.1 Pixel Purity | 20 |
| 2.1.2 Convex Hull | 23 |
| 2.1.3 Nonnegative Matrix Factorization | 25 |
| 2.1.4 Morphological Associative Memories | 30 |
| 2.1.5 Evolutionary Search | 35 |
| 2.1.6 Independent Components Analysis | 36 |
| 2.1.7 Estimating the Number of Endmembers | 38 |
| 2.2 Existing Hyperspectral Band Selection Algorithms | 44 |
| 2.3 Summary of Literature Review | 47 |
| 3 TECHNICAL APPROACH | 55 |
| 3.1 Review of Sparsity Promotion Techniques | 56 |
| 3.2 Review of the Iterated Constrained Endmembers Detection Algorithm | 57 |
| 3.3 New Endmember Detection Algorithm Using Sparsity Promoting Priors | 59 |
| 3.4 New Band Selection Algorithm Using Sparsity Promoting Priors | 61 |
| 3.5 New Endmember Distribution Detection Algorithm | 64 |
| 3.6 Review of Markov Chain Monte Carlo Sampling Algorithms | 68 |
| 3.7 Review of the Dirichlet Distribution and the Dirichlet Process | 70 |
| 3.7.1 Dirichlet Process Mixture Model | 72 |
| 3.7.2 Gibbs Sampling for the Dirichlet Process Mixture Model | 73 |

| | | |
|-------|---|-----|
| 3.8 | New Piece-Wise Convex Endmember Detection Algorithm using the Dirichlet Process | 75 |
| 4 | RESULTS | 81 |
| 4.1 | Sparsity Promoting Iterated Constrained Endmember (SPICE) Algorithm Results | 81 |
| 4.1.1 | The SPICE Two-Dimensional Example Results | 81 |
| 4.1.2 | The SPICE AVIRIS Cuprite Data Results | 82 |
| 4.1.3 | The SPICE AVIRIS Indian Pines Results | 84 |
| 4.1.4 | The SPICE AHI Vegetation Detection Results | 85 |
| 4.2 | Band Selecting SPICE (B-SPICE) Algorithm Results | 90 |
| 4.2.1 | The B-SPICE AVIRIS Cuprite Data Results | 90 |
| 4.2.2 | The B-SPICE AVIRIS Indian Pines Results | 91 |
| 4.2.3 | The B-SPICE AVIRIS Indian Pines Results using Sampled Parameter Values | 93 |
| 4.3 | Endmember Distribution (ED) Detection Results | 94 |
| 4.3.1 | Results on Two-Dimensional Data using ED | 94 |
| 4.3.2 | Results on AVIRIS Cuprite data using ED | 95 |
| 4.4 | Piece-wise Convex Endmember (PCE) Detection Results | 95 |
| 4.4.1 | Detection Results on Two-Dimensional Data using PCE | 95 |
| 4.4.2 | Detection Results on the AVIRIS Indian Pines Data using PCE | 96 |
| 5 | CONCLUSION | 129 |
| | REFERENCES | 131 |
| | BIOGRAPHICAL SKETCH | 139 |

LIST OF TABLES

| <u>Table</u> | <u>page</u> |
|--|-------------|
| 4-1 Number of endmembers found by SPICE and ICE on AVIRIS Cuprite data . . . | 101 |
| 4-2 Mine distributions in overlap regions of AHI and Lynx imagery | 101 |
| 4-3 False alarm rate reduction using blackbody mask in AHI image 1 | 101 |
| 4-4 False alarm rate reduction using blackbody mask in AHI image 2 | 101 |
| 4-5 False alarm rate reduction using blackbody mask in AHI image 3 | 102 |
| 4-6 Mean and standard deviation of the number of endmembers and bands retained using SPICE and B-SPICE on the simulated AVIRIS Cuprite data set | 102 |
| 4-7 Statistics of the averaged squared error per abundance value using SPICE and B-SPICE | 102 |
| 4-8 Indian Pines data set classification results using SPICE and B-SPICE | 103 |
| 4-9 The AVIRIS Indian Pines data B-SPICE results using sampled parameter values | 103 |
| 4-10 Parameter values used to generate ED results on two-dimensional data sets . . . | 103 |
| 4-11 Parameter values used to generate ED results on hyperspectral data sets | 103 |
| 4-12 Parameter values used to generate PCE results | 104 |

LIST OF FIGURES

| <u>Figure</u> | <u>page</u> |
|--|-------------|
| 2-1 Three-dimensional data points and endmember results using convex cone analysis | 48 |
| 2-2 Subset of AVIRIS Cuprite data scene | 48 |
| 2-3 The CCA endmember results on AVIRIS Cuprite data | 49 |
| 2-4 Morphological associative memories endmember detection results using the min memory on two-dimensional data | 49 |
| 2-5 Morphological associative memories endmember detection results using both memories on two-dimensional data | 50 |
| 2-6 Three dimensional data set generated from two endmembers with Gaussian noise. | 51 |
| 2-7 Normalized AVIRIS Cuprite spectra | 52 |
| 2-8 Data set generated from AVIRIS Cuprite endmembers with a small amount of Gaussian noise | 53 |
| 2-9 Data set generated from AVIRIS Cuprite endmembers with a large amount of Gaussian noise | 54 |
| 3-1 Endmember distribution algorithm's abundance prior | 79 |
| 3-2 Endmember distribution algorithm's abundance prior as a function of \mathbf{c} | 79 |
| 3-3 Data points generated from two endmember distributions | 80 |
| 4-1 Two-dimensional example data set | 104 |
| 4-2 Comparison of ICE and SPICE algorithm results on two-dimensional data | 104 |
| 4-3 The SPICE results on two-dimensional data | 105 |
| 4-4 Endmembers found using SPICE on AVIRIS Cuprite data | 105 |
| 4-5 Comparison of SPICE endmember and USGS Alunite spectrum | 106 |
| 4-6 Selected endmembers from AVIRIS Cuprite data | 106 |
| 4-7 Test pixels from AVIRIS Cuprite data | 107 |
| 4-8 Endmember SPICE results on AVIRIS Cuprite scene | 107 |
| 4-9 The AVIRIS Indian Pines data set and ground truth | 108 |
| 4-10 Abundance maps on AVIRIS Indian Pines data generated by SPICE | 109 |
| 4-11 Abundance maps on labeled AVIRIS Indian Pines data generated by SPICE . . . | 110 |

| | | |
|------|---|-----|
| 4-12 | Histograms of SPICE endmember results on labeled AVIRIS Indian Pines data . | 111 |
| 4-13 | Subset of AHI hyperspectral image | 112 |
| 4-14 | Vegetation masks for AHI hyperspectral images generated using SPICE | 113 |
| 4-15 | Selected AVIRIS Cuprite endmembers | 114 |
| 4-16 | The BSPICE AVIRIS Cuprite simulated data endmember results | 114 |
| 4-17 | Histograms of the number of endmembers and bands found using B-SPICE | 115 |
| 4-18 | Results on two-dimensional triangle data found using ED | 115 |
| 4-19 | Data points generated from three endmember distributions | 116 |
| 4-20 | Results on two dimensional data using ED | 116 |
| 4-21 | Results on two dimensional data generated from endmember distributions using SPICE | 117 |
| 4-22 | Results on simulated AVIRIS Cuprite data found using ED | 117 |
| 4-23 | Results on subset of AVIRIS Cuprite data found using ED | 118 |
| 4-24 | Results on subset of AVIRIS Cuprite data found using ED | 119 |
| 4-25 | Two-dimensional data generated from three sets of endmembers | 119 |
| 4-26 | Two-dimensional data results found using PCE | 120 |
| 4-27 | Abundance maps found using PCE on labeled PCA-reduced AVIRIS Indian Pines data | 121 |
| 4-28 | Histograms of PCE endmember results on labeled PCA-reduced AVIRIS Indian Pines data | 122 |
| 4-29 | Abundance maps found using SPICE on labeled PCA-reduced AVIRIS Indian Pines data | 123 |
| 4-30 | Histograms of SPICE endmember results on labeled PCA-reduced AVIRIS Indian Pines data | 124 |
| 4-31 | Abundance maps found using PCE on labeled AVIRIS Indian Pines data with hierarchical dimensionality reduction | 125 |
| 4-32 | Histograms of PCE endmember results on labeled AVIRIS Indian Pines data with hierarchical dimensionality reduction | 126 |
| 4-33 | Abundance maps found using PCE on labeled AVIRIS Indian Pines data | 127 |
| 4-34 | Histograms of PCE endmember results on labeled AVIRIS Indian Pines data . . | 128 |

LIST OF SYMBOLS AND ABBREVIATIONS

| | |
|--------------|---|
| AMEE | Automated morphological endmember extraction algorithm |
| AVIRIS | Airborne visible/infrared imaging spectrometer |
| BSPICE | Band selecting sparsity promoting iterated constrained endmember detection algorithm |
| CCA | Convex cone analysis |
| D | Dimensionality of the input hyperspectral data set |
| DECA | Dependent components analysis |
| DPEM | Dirichlet process endmember detection algorithm |
| DWT | Discrete wavelet transform |
| \mathbf{E} | Matrix of endmembers where the i^{th} column contains the spectral signature for the i^{th} endmember, \mathbf{e}_i |
| ED | Endmember distribution detection algorithm |
| EMD | Euclidean minimum distance |
| FCM | Fuzzy c-means clustering algorithm |
| HFC | Harsanyi-Farrand-Chang method |
| ICA | Independent components analysis |
| ICE | Iterated constrained endmembers algorithm |
| iid | Independently and identically distributed |
| M | True or estimated number of endmembers for a given hyperspectral scene |
| \mathbf{m} | Vector containing the mean of each band of a given hyperspectral image |
| MCMC | Markov chain monte carlo |
| MEI | Morphological eccentricity index |
| MNF | Maximum noise fraction transform |
| MVC-NMF | Minimum volume constrained non-negative matrix factorization algorithm |
| MVT | Minimum volume transform |
| NATGD | Noise-adjusted transformed Gerschgorin disk |
| NMF | Non-negative matrix factorization |

| | |
|----------|--|
| NSP | Noise subspace projection method |
| NWHFC | Noise whitened Harsanyi-Farrand-Chang method |
| P | Matrix of proportion (abundance) values where each element, p_{ik} , is the abundances of endmember k in pixel i |
| PCA | Principal components transform |
| PCE | Piece-wise convex endmember detection algorithm |
| PNACP | Partitioned noise-adjusted principal components algorithm |
| PPI | Pixel purity index |
| RVM | Relevance vector machine |
| SAM | Spectral angle mapper |
| SIE | Single individual evolutionary strategy |
| SPICE | Sparsity promoting iterated constrained endmembers algorithm |
| SSEE | Spatial-spectral endmember extraction algorithm |
| SVD | Singular value decomposition |
| TGD | Transformed Gerschgorin disk |
| VD | Virtual dimensionality |
| X | Input hyperspectral data set |

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

HYPERSPECTRAL ENDMEMBER DETECTION AND BAND SELECTION USING BAYESIAN METHODS

By

Alina Zare

December 2008

Chair: Paul Gader

Major: Computer Engineering

Four methods of endmember detection and spectral unmixing are described. The methods determine endmembers and perform spectral unmixing while simultaneously determining the number of endmembers, representing endmembers as distributions, partitioning the input data set into several convex regions, or performing hyperspectral band selection. Few endmember detection algorithms estimate the number of endmembers in addition to determining their spectral shape. Also, methods which treat endmembers as distributions or treat hyperspectral images as piece-wise convex data sets have not been previously developed.

A hyperspectral image is a three-dimensional data cube containing radiance values collected over an area (or scene) in a range of wavelengths. Endmember detection and spectral unmixing attempt to decompose a hyperspectral image into the pure - separate and individual - spectral signatures of the materials in a scene, and the proportions of each material at every pixel location. Each spectral pixel in the image can then be approximated by a convex combination of proportions and endmember spectra.

The first method, the Sparsity Promoting Iterated Constrained Endmembers (SPICE) algorithm, incorporates sparsity-promoting priors to estimate the number of endmembers. The algorithm is initialized with a large number of endmembers. The sparsity promotion process drives all proportions of some endmembers to zero. These endmembers can be removed by SPICE with no effect on the error incurred by representing the image

with endmembers. The second method, the Endmember Distribution detection (ED) algorithm, models each endmember as a distribution rather than a single spectrum incorporating an endmember’s inherent spectral variation or the variation due to differing environmental conditions. The third method, the Piece-wise Convex Endmember (PCE) detection algorithm, partitions the input hyperspectral data set into convex regions while simultaneously estimating endmember distributions for each partition and proportion values for each pixel in the image. The number of convex regions are determined autonomously using the Dirichlet process. The fourth method is known as the Band Selecting Sparsity Promoting Iterated Constrained Endmember (B-SPICE) algorithm and is an extension of SPICE that performs hyperspectral band selection in addition to all of SPICE’s endmember detection and spectral unmixing features. This method applies sparsity promoting priors to discard those hyperspectral bands which do not aid in distinguishing between endmembers in a data set. All of the presented algorithms are effective at handling highly-mixed hyperspectral images where all of the pixels in the scene contain mixtures of multiple endmembers. These methods are capable of extracting endmember spectra from a scene that does not contain pure pixels composed of only a single endmember’s material. Furthermore, the methods conform to the Convex Geometry Model for hyperspectral imagery. This model requires that the proportions associated with an image pixel be non-negative and sum to one.

Results indicate that SPICE and B-SPICE consistently produce the correct number of endmembers and the correct spectral shape for each endmember. The B-SPICE algorithm is shown to significantly decrease the number of hyperspectral bands while maintaining competitive classification accuracy for a data set. The ED algorithm results indicate that the algorithm produces accurate endmembers and can incorporate spectral variation into the endmember representation. The PCE algorithm results on hyperspectral data indicate that PCE produces endmember distributions which represent the true ground truth classes of the input data set.

CHAPTER 1 INTRODUCTION

1.1 Hyperspectral Image Data and Analysis

Hyperspectral imaging sensors capture both the spatial and spectral information of a scene. A hyperspectral sensor collects radiance data in hundreds of contiguous wavelengths. As a sensor collects data over a region, a three-dimensional data cube is generated. The data cube can be interpreted as a stack of two-dimensional images captured over a range of wavelengths. Each element of the three-dimensional data cube corresponds to the radiance measured in a particular wavelength at one ground location (Keshava and Mustard, 2002; Manolakis, Marden, and Shaw, 2003).

Radiance measured by a hyperspectral sensor is a combination of radiation that is reflected and/or emitted by materials on the ground (Manolakis et al., 2003). In passive systems, the reflected portion of the signal is the amount of radiation reflected from sunlight shining on ground materials (Keshava and Mustard, 2002). The atmosphere between the sensor and materials on the ground affects the radiance measurements. Water vapor and oxygen in the atmosphere cause the largest effect. In certain wavelengths, those known as *absorption bands*, water vapor and oxygen absorb a large portion of the signal, causing poor signal-to-noise ratios (Manolakis et al., 2003). In addition to absorption characteristics, the different wavelengths across which radiance can be measured have varying properties. For example, in the 0.4 to 2.5 μm range, sunlight or another active illumination source is needed since reflected radiance dominates this portion of the spectrum. In contrast, the thermal infrared region from 8 to 14 μm is dominated by emitted radiance and can, therefore, be measured during the night without an active illumination source (Manolakis et al., 2003).

The main appeal for hyperspectral imaging is the concept that different materials reflect and emit varying amounts of radiance across the electromagnetic spectrum. In other words, different materials generally have unique spectral signatures. It is for this

reason that hyperspectral sensors can be used to identify and distinguish between different materials in a scene (Manolakis et al., 2003).

Two important characteristics of a hyperspectral sensor are its spectral and spatial resolution. Spectral resolution of a sensor corresponds to the range of wavelengths over which radiance values are measured and combined to become a single band in a hyperspectral image. Spatial resolution corresponds to the size of the physical area on the ground from which radiance measurements are taken for a single image pixel. As the area corresponding to a pixel increases, the spatial resolution of the image decreases (Keshava and Mustard, 2002; Manolakis et al., 2003). For airborne systems, spatial resolution is generally constant across an image. However, for many forward-looking ground-based systems, the spatial resolution may vary within an image. The varying spatial resolution is a result of the angle from which a hyperspectral sensor images a region. Pixels closer to the sensor have higher spatial resolution than those farther away.

Spatial resolution is one of the causes of *mixed pixels* in a hyperspectral data set (Keshava and Mustard, 2002; Manolakis et al., 2003). A mixed pixel is a pixel which combines the radiance values of multiple materials. A *pure pixel* corresponds to a single material's radiance values. Mixed pixels can occur from low spatial resolution since, as a pixel's corresponding area on the ground increases, neighboring materials are likely to be combined into the image pixel. Mixed pixels also occur when the different materials are mixed on the ground. Beach sand is a common example for this type of mixed pixel since grains of different materials are intermingled (Keshava and Mustard, 2002).

1.1.1 Endmember Detection

Pure spectral signatures, or the *constituent spectra*, in an imaged scene are referred to as *endmembers* (Keshava and Mustard, 2002). Due to the presence of mixed pixels in a hyperspectral image, *spectral unmixing* is often performed to decompose mixed pixels into their respective endmembers and *abundances*. Abundances are the proportions of

the endmembers in each pixel in a hyperspectral image. Spectral unmixing relies on the definition of a mixing model.

Complex mixing models for hyperspectral imagery can be defined. These complex models can take into account the atmospheric effects and the orientation, size, and shape of objects in a scene. They can also consider the incident angles of sunlight and the sensor on ground materials. Despite the large number of variables that can be included into a mixing model, the most popular model is the *convex geometry model* (also known as the *linear mixing model*) (Keshava and Mustard, 2002; Nascimento and Bioucas-Dias, 2005a).

The convex geometry model assumes that every pixel is a convex combination of endmembers in the scene. This model can be written as shown in Equation 1-1 (Keshava and Mustard, 2002; Manolakis et al., 2003; Nascimento and Bioucas-Dias, 2005a),

$$\mathbf{x}_i = \sum_{k=1}^M p_{ik} \mathbf{e}_k + \epsilon_i \quad i = 1, \dots, N \quad (1-1)$$

where N is the number of pixels in the image, M is the number of endmembers, ϵ_i is an error term, p_{ik} is the proportion of endmember k in pixel i , and \mathbf{e}_k is the k^{th} endmember. The proportions of this model satisfy the constraints in Equation 1-2,

$$p_{ik} \geq 0 \quad \forall k = 1, \dots, M; \quad \sum_{k=1}^M p_{ik} = 1. \quad (1-2)$$

The convex geometry model has been found to effectively describe regions where the various pure materials are separated into regions dominated by a single endmember. Generally, mixed pixels in these types of regions are caused by a sensor's inadequate spatial resolution. In cases where materials are mixed on the ground, nonlinear mixing models have been found to be more effective (Keshava and Mustard, 2002).

The endmember detection problem is the task of determining the pure spectral signatures in a given hyperspectral scene. Endmember detection algorithms often assume the convex geometry model and perform spectral unmixing to return the endmembers and abundances in an image (Keshava and Mustard, 2002).

1.1.2 Hyperspectral Band Selection

Since, for each pixel, hyperspectral sensors measure radiance values at a very large number of wavelengths, this imagery contains an immense amount of data (Manolakis et al., 2003). Although the resolution provided allows for the extraction of material spectra, the volume of data poses many challenging problems such as data storage, computational efficiency, and the *curse of dimensionality* (Chang, Du, Sun, and Althouse, 1999; Huang and He, 2005). One method to overcome these challenges is the use of data reduction techniques (Huang and He, 2005). Hyperspectral band selection is one method of data reduction that also retains the physical meaning of the data set (Guo, Gunn, Damper, and Nelson, 2006). Hyperspectral band selection selects a set of bands from the input hyperspectral data set which retain the information needed for subsequent hyperspectral image spectroscopy.

1.2 Statement of Problem

Most endmember detection algorithms require the knowledge of the number of endmembers for a given hyperspectral scene. Also, many existing algorithms rely on the *pixel purity assumption* which assumes that pure pixels for each endmember exist in the data set. Some existing endmember detection algorithms do not unmix the data set and do not provide abundance values that conform to the non-negativity and sum-to-one constraints in Equation 1–2. Existing endmember detection algorithms represent endmembers as single spectral points, which does not incorporate the spectral variability that occurs due to differing environmental conditions. Furthermore, existing endmember detection algorithms generally assume that the hyperspectral data points lie in a single convex region with one set of endmembers. However, it may be the case that multiple sets of endmembers, defining several overlapping convex regions, can better describe the hyperspectral image.

Existing hyperspectral band selection algorithms often require an input of the number of hyperspectral bands to be retained. Furthermore, many hyperspectral data

reduction techniques perform a projection or merging of the bands which removes each band’s physical meaning. Many hyperspectral band selection algorithms are also tied to a classification problem which requires labeled training data to determine the bands which distinguish between the identified classes.

This study examines methods to tackle endmember detection and hyperspectral band selection. Algorithms that autonomously estimate the number of endmembers and hyperspectral bands while simultaneously estimating endmember spectral shapes, and which yield abundances which conform to the convex geometry model’s constraints, retain physically meaningful bands, and avoid reliance on the pixel purity assumption are investigated. Methods which determine endmember distributions and autonomously learn the number of convex regions needed to describe an input hyperspectral scene are presented.

1.3 Overview of Research

The conducted research involves the development and analysis of three novel endmember detection algorithms. These methods either determine the number of endmembers required for a scene, learn endmember distributions, or determine the number of convex regions needed to describe a hyperspectral image while simultaneously estimating the endmembers’ spectral distributions. Furthermore, all of the presented methods also simultaneously determine appropriate abundance values for every pixel and do not rely on the pixel purity assumption. Additionally, a novel hyperspectral band selection algorithm is developed that: determines the needed number of hyperspectral bands to distinguish between endmembers for a given scene, performs unsupervised band selection, and retains the physical meaning of the hyperspectral bands.

The endmember detection algorithms determine the spectral shape of each endmember and the proportion of each endmember in each pixel. These algorithms are based on the convex geometry model in Equation 1–1 and thus constrain the proportion values to be non-negative and sum to one. The general approach involves integrating

state-of-the-art machine learning approaches based on Bayesian methods into the framework of hyperspectral image spectroscopy. The Sparsity Promoting Iterated Constrained Endmembers (SPICE) algorithm determines the number of endmembers by beginning with a large number of initial endmembers and removing endmembers as they become superfluous. The number of endmembers are determined by applying a sparsity promoting prior to the proportions for each endmember. The Endmember Distributions (ED) algorithm estimates the distribution of each endmember for an input data set rather than single spectra. The Piece-wise Convex Endmember (PCE) detection algorithm uses the Dirichlet process to determine the number of convex regions needed to describe an input hyperspectral image while simultaneously performing spectral unmixing and determining endmember distributions for each convex region. The sparsity promoting, endmember distributions, and Dirichlet process techniques utilize Bayesian machine learning approaches to estimate the number of endmembers, learn endmember distributions, or partition the data set into convex regions while estimating proportion values and values for the endmembers themselves.

In addition to the endmember detection and spectral unmixing algorithms, a simultaneous band selection and endmember detection algorithm is developed. This method, the Band Selecting Sparsity Promoting Iterated Constrained Endmember (B-SPICE) algorithm, determines the number of required bands for a data set needed to distinguish between the endmembers in a scene. This is in contrast to previous approaches which incorporate separate metrics into the objective function to find bands that aid in discriminating between labeled regions in a scene. In addition to performing band selection and determining the number of bands needed, this algorithm performs endmember determination and spectral unmixing. The unnecessary bands and endmembers in this method are removed using sparsity promoting priors.

CHAPTER 2

LITERATURE REVIEW

This chapter provides a review of existing hyperspectral endmember detection algorithms followed by a summary of existing hyperspectral band selection algorithms.

2.1 Existing Endmember Detection Algorithms

Many endmember detection algorithms are described in the literature. The majority of these algorithms rely on the convex geometry model described in Equation 1-1 (Keshava and Mustard, 2002). Most existing algorithms require advance knowledge of the number of endmembers in a given scene. However, this value is often unknown for a given data set. Several methods make the *pixel purity assumption* and assume that pure pixels exist in the input data set for every endmember in the scene. This assumption causes algorithms to be inaccurate for highly-mixed data sets where pure pixels for each material cannot be found in the imagery. Additionally, some methods do not encompass all of the data points and, therefore, either prevent spectral unmixing with abundance values that conform to the constraints in Equation 1-2 or have large reconstruction errors using the estimated endmember and abundance matrices. The existing methods generally represent each endmember as a single spectrum which does not account for the spectral variation that may occur due to varying environmental conditions. The majority of these methods also assume that the hyperspectral data points lie in a single convex region and can be described by a single set of endmembers which encompass the data set. In this chapter, a summary of many of these existing endmember detection algorithms is provided.

2.1.1 Pixel Purity

Many endmember detection algorithms rely on the assumption that the spectral signature for each endmember can be found without performing spectral unmixing on the data set. This assumes that there exists at least one pixel for each endmember which consists of only that endmember's material. Furthermore, the hyperspectral imaging

device must be operating at a spatial resolution that does not combine endmember spectra with the spectra of neighboring materials. Algorithms relying on the pixel purity assumption include the NFindr algorithm (Winter, 1999) and the Pixel Purity Index algorithm (Boardman, Kruse, and Green, 1995) both of which are described in detail below.

Additionally, the Automated Morphological Endmember Extraction (AMEE) algorithm defines multispectral dilation and erosion operators used to compute the *morphological eccentricity index* (MEI) (Plaza, Martinez, Perez, and Plazas, 2002). The MEI is used to identify spectrally pure pixels in the image which are returned as endmembers (Plaza et al., 2002). The Spatial-Spectral Endmember Extraction (SSEE) algorithm projects the image onto eigenvectors computed from the Singular Value Decomposition (SVD) of subsets in the input data set (Rogge, Rivard, Zhang, Sanchez, Harris, and Feng, 2007). SSEE identifies candidate endmembers as those that fall on the extreme ends of the projection and returns either the pixels or the mean of pixels that are spatially close and spectrally similar. The method based on Morphological Associative Memories described by Grana, Sussner, and Ritter (2003) also depends on the pixel purity assumption for endmember extraction as described in Section 2.1.4. Vertex Component Analysis adds endmembers sequentially by selecting pixels which project farthest in a direction orthonormal to the space spanned by the current endmember set (Nascimento and Bioucas-Dias, 2005b). Thus, Vertex Component Analysis also relies on the pixel purity assumption (Nascimento and Bioucas-Dias, 2005b).

NFindr. The NFindr algorithm is a well-known, established method of endmember detection that searches for endmembers within an input hyperspectral data set (Winter, 1999). NFindr seeks the set of input pixels that encompass the largest volume (Winter, 1999).

The algorithm begins by randomly selecting a set of pixels from the image to be the initial endmember set. Then, each endmember is replaced, in succession, by all other

pixels in the image. After each replacement, the volume of the space defined by the current set of potential endmembers is computed. When a replacement increases the volume, the replacement is maintained. The algorithm cycles through image pixels and endmembers until no further replacements are made (Winter, 1999).

The volume enclosed by each set of potential endmembers is computed using Equation 2-1,

$$V(\mathbf{E}^*) = \frac{1}{(M-1)!} \text{abs}(|\mathbf{E}^*|) \quad (2-1)$$

where

$$\mathbf{E}^* = \begin{pmatrix} 1 & 1 & \dots & 1 \\ \mathbf{e}_1 & \mathbf{e}_2 & \dots & \mathbf{e}_M \end{pmatrix} \quad (2-2)$$

M is the number of endmembers and \mathbf{e}_i is a column vector containing an endmember. If $(M-1)$ is not the dimensionality of the data, then a dimensionality reduction method, such as Principal Components Analysis or Maximum Noise Fraction, must be employed (Green, Berman, Switzer, and Craig, 1988; Lee, Woodyatt, and Berman, 1990). The data dimensionality must be one less than the desired number of endmembers since the determinant of a non-square matrix is not defined (Winter, 1999).

This algorithm works by maximizing the volume by the endmembers inscribed within the hyperspectral data cloud. Since the endmembers are found within the data cloud, the endmembers may not enclose all the data points. In addition to assuming pure pixels can be found in the image, this algorithm requires knowledge of the number of endmembers in advance (Winter, 1999).

Pixel purity index. The Pixel Purity Index, PPI, is a commonly used algorithm for determining the purest pixels in an input image (Boardman et al., 1995). The PPI algorithm ranks image pixels based on their *pixel purity indices*. Then, the M pixels with the highest pixel purity values are returned as potential endmembers. The number of endmembers, M , is not determined by this algorithm. PPI is often used for generating candidate endmembers which are then used as inputs to other endmember extraction algorithms (Berman, Kiiveri, Lagerstrom, Ernst, Donne, and Huntington, 2004) or loaded

into a visualization tool for users to hand select endmembers from the candidates (Rogge et al., 2007).

The PPI algorithm assigns each pixel a pixel purity value by repeatedly projecting all of the pixels onto randomly directed vectors. The algorithm is initialized by assigning all pixels a pixel purity value of zero. The pixel purity values are updated following each random projection by adding one to the values of the pixels that fall near either end of every projection. Since PPI values are generated using random vectors, the results are dependent on the number of random projections and the threshold for determining if a pixel’s projection is considered near an end-point (Boardman et al., 1995).

2.1.2 Convex Hull

The convex geometry model defines endmembers to be the vertices of a simplex that surround the pixels in an image. The Minimum Volume Transform and Convex Cone Analysis methods are based on this model and search for points that lie at the corners of a simplex surrounding the data.

The Minimum Volume Transform (MVT) (Craig, 1994) finds the smallest simplex that circumscribes the hyperspectral data points. This is in contrast to the NFindr method which obtains the largest simplex inscribed within the input data set (Winter, 1999). MVT searches for hyperplanes that minimize their enclosed volume while encompassing all of the data. The algorithm then iteratively varies the hyperplanes using linear programming methods to provide a progressively tighter fit around the data. After minimizing the volume enclosed by the hyperplanes while encompassing all of the input data, the intersections of the planes are returned as endmembers. Although MVT does not require pure pixels to be in the data set, it does require the number of endmembers in advance. The method performs the Maximum Noise Fraction transform (Green et al., 1988) to reduce the data dimensionality to $(M - 1)$ where M is the number of endmembers.

Convex cone analysis. The convex cone analysis (CCA) method (Ifarraguerri and Chang, 1999) of endmember extraction also searches for the boundaries of a convex, non-negative region that enclose the input data points. This method relies on the fact that radiance values are non-negative and, therefore, can restrict the endmembers to be non-negative points. The method requires an input of the number of desired endmembers, M . Given M , the eigenvectors of the sample correlation matrix that correspond to the M largest eigenvalues are computed (Ifarraguerri and Chang, 1999).

$$\mathbf{X}_N^T \mathbf{X}_N = \mathbf{C} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T \quad (2-3)$$

where \mathbf{X}_N is the normalized input data matrix, \mathbf{C} is the sample correlation matrix, $\mathbf{\Lambda}$ is a diagonal matrix of the sample correlation matrix's sorted eigenvalues, and \mathbf{U} is the matrix of sorted eigenvectors. The data is normalized such that every data point has a constant L2-norm. This normalization retains the spectral shape of each data point while constraining each point to the same hyper-sphere. Let $\{\mathbf{u}_i\}_{i=1}^M$ be the set of the M most significant eigenvectors. Given the eigenvectors, the endmembers in this method are defined to be linear combinations of the eigenvectors that satisfy Equation 2-4,

$$\mathbf{e} = \mathbf{u}_1 + a_1 \mathbf{u}_2 + \cdots + a_{M-1} \mathbf{u}_M \geq \mathbf{0} \quad (2-4)$$

where $\mathbf{0}$ is the zero vector (Ifarraguerri and Chang, 1999). The first eigenvector of the sample correlation matrix, \mathbf{u}_1 , will point towards the data set. Equation 2-4 can be interpreted as perturbing the first eigenvector by a linear combination of the other orthogonal eigenvectors while constraining the endmembers to be non-negative.

Since each eigenvector is of dimension D , solving for the $(M - 1)$ a_j coefficients in Equation 2-4 is an over-determined problem. Because of this, the CCA method iterates through each subset of bands of size $(M - 1)$ and solves a set of $(M - 1)$ linear equations for the a_j coefficients,

$$\mathbf{u}_1(\gamma^i) + a_1 \mathbf{u}_2(\gamma^i) + \cdots + a_{M-1} \mathbf{u}_M(\gamma^i) = 0 \quad (2-5)$$

where γ^i represents the indices of the i^{th} set of $(M - 1)$ hyperspectral bands. After solving for the a_j coefficients, they are plugged back into Equation 2-4. The vectors computed by plugging back into Equation 2-4 will contain $(M - 1)$ zero values. The remaining band values are checked to ensure that they are non-negative. If a potential endmember is found to be non-negative, then it is kept as an endmember, otherwise, that vector is discarded (Ifarraguerri and Chang, 1999).

The CCA method searches through $\binom{D}{M-1}$ potential endmembers which can be prohibitive for data sets with a large number of hyperspectral bands. Furthermore, since $\binom{D}{M-1}$ may be greater than M , more endmembers than specified may be found. Ifarraguerri and Chang (1999) list potential methods for removing the additional endmembers such as removing endmembers that are collinear with other endmember spectra.

This algorithm does not provide endmembers which tightly surround the data points. This is an artifact from the $(M - 1)$ zeros in each endmember spectra as can be seen in Figure 2-1. Figure 2-1 shows the three-dimensional data set and the three endmembers found using CCA. Since each endmember has two zeros in their spectra, the endmember spectra lie along the x-, y- and z-axis rather than tightly surrounding the data set. This is further illustrated in Figures 2-2 and 2-3. The normalized data points in Figure 2-2 are the first twenty-five bands (approximately 1978 to 2228nm) from a subset of pixels in the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Cuprite “Scene 4” data set (AVIRIS). Figure 2-3 shows the nine endmembers determined using CCA with M set to three. The normalized data set values range from 0.138 to 0.244 while the range of endmember values is 0 to 0.904.

2.1.3 Nonnegative Matrix Factorization

Several methods for endmember extraction have been developed based on Non-Negative Matrix Factorization (NMF). Non-Negative Matrix Factorization searches for two non-negative matrices, $\mathbf{E} \in \mathbf{R}^{D \times M}$ and $\mathbf{P} \in \mathbf{R}^{M \times N}$ that approximate an input

non-negative matrix $\mathbf{X} \in \mathbf{R}^{D \times N}$ (Lee and Seung, 1999; Miao and Qi, 2007),

$$\mathbf{X} \approx \mathbf{E}\mathbf{P}. \quad (2-6)$$

The non-negative assumptions in this method are appropriate for endmember detection since hyperspectral radiance data is nonnegative (Miao and Qi, 2007; Pauca, Piper, and Plemmons, 2005). One NMF algorithm proposed by Lee and Seung (2000) minimizes the objective function in Equation 2-7 (Pauca et al., 2005),

$$f(\mathbf{E}, \mathbf{P}) = \frac{1}{2} \|\mathbf{X} - \mathbf{E}\mathbf{P}\|_F^2 = \sum_{i=1}^D \sum_{j=1}^N (\mathbf{X}_{ij} - (\mathbf{E}\mathbf{P})_{ij})^2. \quad (2-7)$$

The NMF update developed by Lee and Seung (2000) uses the *multiplicative update rules* in Equations 2-8 and 2-9,

$$p_{ij}^{k+1} = p_{ij}^k \frac{(\mathbf{E}^T \mathbf{X})_{ij}^k}{(\mathbf{E}^T \mathbf{E} \mathbf{P})_{ij}^k} \quad (2-8)$$

$$e_{ij}^{k+1} = e_{ij}^k \frac{(\mathbf{X}^T \mathbf{P})_{ij}^k}{(\mathbf{E} \mathbf{P} \mathbf{P}^T)_{ij}^k} \quad (2-9)$$

where k indicates the iteration. The elements of the \mathbf{P} and \mathbf{E} matrices are updated simultaneously by iterating between the elements of the two matrices (Pauca et al., 2005). Lee and Seung (2000) prove that the distance in Equation 2-7 does not increase when using the updates in Equations 2-8 and 2-9. As shown by Lee and Seung (2000), the multiplicative update rules are equivalent to standard gradient descent updates when the step size parameter is set to $\frac{p_{ij}^k}{(\mathbf{E}^T \mathbf{E} \mathbf{P})_{ij}^k}$ when updating proportion values and $\frac{e_{ij}^k}{(\mathbf{E} \mathbf{P} \mathbf{P}^T)_{ij}^k}$ when updating endmember values.

The basic NMF algorithm has been modified to include constraints and initialization methods for better performance in endmember detection. Three endmember extraction algorithms based on NMF are described.

Minimum volume constrained nonnegative matrix factorization. The Minimum Volume Constrained Nonnegative Matrix Factorization (MVC-NMF) algorithm

for endmember detection (Miao and Qi, 2007) attempts to minimize the objective function in Equation 2–10 solving for endmembers and the abundance values for each pixel (Miao and Qi, 2007).

$$\min f(\mathbf{E}, \mathbf{P}) = \frac{1}{2} \sum_{i=1}^D \sum_{j=1}^N (\mathbf{X}_{ij} - \mathbf{E} \mathbf{P}_{ij})^2 + \lambda \frac{1}{2(M-1)!} \left| \begin{pmatrix} \mathbf{1}_M^T \\ \tilde{\mathbf{E}} \end{pmatrix} \right|^2 \quad (2-10)$$

where

$$\tilde{\mathbf{E}} = \mathbf{U}^T (\mathbf{E} - \mu \mathbf{1}_M^T) \quad (2-11)$$

The \mathbf{U} matrix consists of the $(M - 1)$ most significant principal components of the input data, \mathbf{X} , and $\mathbf{1}_M^T$ is an M length vector of ones. The size of the endmember matrix is reduced using Equation 2–11 in order to be able to compute the determinant in Equation 2–10 (Miao and Qi, 2007).

The first term of the objective function is a squared error term. By minimizing the first term, the error between the input data set and the estimated pixels computed from the abundance values and endmembers are minimized. The second term of the objective function is the volume of the space defined by the endmembers. By minimizing the second term, the endmembers provide a tight fit around the data. These two terms can be seen as an “internal force” and an “external force” (Miao and Qi, 2007). The first term can be interpreted as an outward force that prefers endmembers which completely encompass the data and the second term is an inward force that wants to minimize the volume enclosed by the endmembers (Miao and Qi, 2007).

In MVC-NMF, the objective function in Equation 2–10 is minimized using gradient descent with clipping. The values for the endmembers and their proportions are updated in an alternating fashion. In other words, in each iteration of the algorithm, either the endmembers or the proportions are updated while the other is held constant (Miao and Qi, 2007).

In order to enforce the non-negativity constraint in Equation 1–2, after solving for either the endmember or the proportions, any negative values are set to zero.

$$\mathbf{E}^{k+1} = \max(\mathbf{0}, \mathbf{E}^k - \alpha^k \nabla_{\mathbf{E}} f(\mathbf{E}^k, \mathbf{P}^k)) \quad (2-12)$$

$$\mathbf{P}^{k+1} = \max(\mathbf{0}, \mathbf{P}^k - \beta^k \nabla_{\mathbf{P}} f(\mathbf{E}^k, \mathbf{P}^k)) \quad (2-13)$$

where α and β are the gradient descent learning rates (Miao and Qi, 2007).

To promote the sum-to-one constraint of the proportions, when updating proportion values, the endmember and data matrices are augmented by a row of constant positive values. The larger the constant, the more emphasis is placed on the sum-to-one constraint (Miao and Qi, 2007).

The algorithm seeks endmembers that minimize the squared reconstruction error. The algorithm also allows for some resilience to noise and selects endmembers that provide a tight fit around the data. Still, the MVC-NMF algorithm does have some drawbacks. The algorithm requires knowledge of the number of endmembers in advance and does not strictly enforce the sum-to-one constraint.

Constrained non-negative matrix factorization. Pauca, Piper, and Plemmons (2005) also develop a method of endmember extraction based on the NMF algorithm. Their constrained NMF algorithm incorporates smoothness constraints into the NMF objection function described in Equation 2–7. The resulting objective function is shown in Equation 2–14,

$$\min_{\mathbf{E}, \mathbf{P}} \{ \|\mathbf{X} - \mathbf{EP}\|_F^2 + a \|\mathbf{E}\|_F^2 + \beta \|\mathbf{P}\|_F^2 \} \quad (2-14)$$

where a and b are regularization parameters balancing the error and smoothness terms (Pauca et al., 2005). The smoothness terms encourage sparsity within the matrices and are equivalent to applying a Gaussian prior on the endmembers and abundances. This objective is minimized using gradient descent. Following Lee and Seung (2000), the step size parameters are set to $\frac{p_{ij}^k}{(\mathbf{E}^T \mathbf{EP})_{ij}^k}$ when updating proportion values and $\frac{e_{ij}^k}{(\mathbf{EPP}^T)_{ij}^k}$ when updating endmember values (Pauca et al., 2005).

[Pauca, Piper, and Plemmons \(2005\)](#), after computing endmembers using the Constrained NMF method, retain endmembers based on their similarity to laboratory measured spectra. Endmembers are compared to laboratory spectra using the *symmetric Kullback-Leibler Divergence*,

$$K_s(\mathbf{e}_1^N, \mathbf{e}_2^N) = K(\mathbf{e}_1^N || \mathbf{e}_2^N) + K(\mathbf{e}_2^N || \mathbf{e}_1^N) \quad (2-15)$$

where

$$K(\mathbf{x} || \mathbf{y}) = \sum_{i=1}^D \mathbf{x}(i) \log \left(\frac{\mathbf{x}(i)}{\mathbf{y}(i)} \right) \quad (2-16)$$

and \mathbf{e}^N are endmembers normalized to sum to one. The symmetric Kullback-Leibler divergence is computed between each endmember and every library spectra. An endmember is associated to the library spectrum for which the endmember has the smallest divergence value given that the divergence is below a threshold, τ . If all of the symmetric Kullback-Leibler divergences computed for an endmember are greater than τ , the endmember is pruned ([Pauca et al., 2005](#)). Like the previous NMF-based algorithms, this method required an estimate of the number of endmembers prior to running the NMF algorithm. However, this number may change based on the final pruning step which requires access to a spectral library containing signatures that can be found in the image.

Fuzzy c-means initialized non-negative matrix factorization. [Liou and Yang \(2005\)](#) also developed an endmember extraction method based on NMF. Their method relies on the basic NMF multiplicative update rule ([Lee and Seung, 1999](#)) but provides a method of initializing the two non-negative matrices. The \mathbf{E} and \mathbf{P} matrices are initialized using the Fuzzy C-Means (FCM) clustering method. The FCM clustering algorithm clusters the data into M clusters with each input point having varying degrees of membership in each cluster. The objective function for FCM is defined in Equation 2-17,

$$J = \sum_{k=1}^M \sum_{i=1}^N u_{ki}^m d_{ik} + \sum_{i=1}^N \lambda_i \left(\sum_{k=1}^M u_{ki} - 1 \right) \quad (2-17)$$

where M is the number of clusters (in this case, the number of endmembers), N is the number of data points, m is the fuzzifier, u_{ki} is the membership of the i^{th} points to the k^{th} cluster, d_{ik} is the squared distance between the i^{th} data point and the k^{th} cluster center and λ_i is the Lagrange multiplier (Liou and Yang, 2005; Theodoridis and Koutroumbas, 2003). FCM minimizes the objective function by updating membership values and cluster centers.

Since FCM provides both cluster centers and membership values for each data point, the matrices for the NMF algorithm are initialized using the cluster centers as the initial endmembers and the membership values as the initial abundance values. Since NMF is dependent on initialization, well chosen initial matrices can improve performance (Liou and Yang, 2005).

This method also requires advance knowledge of the number of endmembers to perform both the FCM and the NMF algorithms. Liou and Yang (2005) utilize the Partitioned Noise-Adjusted Principal Component Analysis method (Tu, Huang, and Chen, 2001) to try to estimate the number of endmembers prior to applying the endmember detection algorithm. This method of estimating the number of endmembers is described in Section 2.1.7.

2.1.4 Morphological Associative Memories

Many methods for endmember detection have been based on the use of Morphological Associative Memories (Ritter and Gader, 2006). There are two types of morphological memories that can be computed, the *min memory* and the *max memory*. Given a set of input vectors, $X = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ and associated desired outputs, $Y = \{\mathbf{y}_1, \dots, \mathbf{y}_N\}$. The min and max morphological associative memories, \mathbf{W}_{XY} and \mathbf{M}_{XY} , are computed using

Equations 2–18 and 2–19,

$$\mathbf{W}_{XY} = \bigwedge_{i=1}^N [\mathbf{y}_i + (\mathbf{x}_i)^*] \quad (2-18)$$

$$\mathbf{M}_{XY} = \bigvee_{i=1}^N [\mathbf{y}_i + (\mathbf{x}_i)^*] \quad (2-19)$$

where \mathbf{x}^* is the lattice conjugate transpose of \mathbf{x} which is defined to be $\mathbf{x}^* = (-\mathbf{x})^T$.

Auto-associative morphological memories are the morphological associative memories which associate a set X to itself. The min and max auto-associative memories are related to each other using the conjugate transpose operator, $\mathbf{W}_{XX} = \mathbf{M}_{XX}^*$.

Patterns are recalled using associative morphological associative memories through either the *max product* or the *min product*, $\mathbf{y} = \mathbf{W}_{XY} \boxtimes \mathbf{x}$ and $\mathbf{y} = \mathbf{M}_{XY} \boxtimes \mathbf{x}$. The max product between an $m \times p$ matrix \mathbf{A} and an $p \times n$ matrix \mathbf{B} is defined in Equation 2–20,

$$c_{ij} = \bigvee_{k=1}^N a_{ik} + b_{kj}. \quad (2-20)$$

The min product is similarly defined using a minimum operator.

Morphological associative memories have been used for endmember detection because they can be used to find affinely independent points in D -dimensional space or points that are *morphologically independent* (Grana, Sussner, and Ritter, 2003; Myers, 2005). The convex hull of endmembers that follow the convex geometry model in Equation 1–1 defines a volume in D -dimensional space which surround the hyperspectral data points in an image. The convex hull of $D + 1$ affinely independent points defines a simplex in D -dimensional space (Myers, 2005; Ritter and Urcid, 2008). Therefore, the motivation to find affinely independent points using morphological associative memories is that the simplex defined by these points bounds a volume in D -dimensional space which can be used to try to surround the hyperspectral image points (Myers, 2005). Morphological associative memories can also be used to determine whether points are morphologically independent (Grana et al., 2003). Some endmember detection algorithms

using morphological associative memories search for morphologically independent endmembers which surround the points of the hyperspectral image (Grana et al., 2003).

Morphological associative memory method 1. Grana, Sussner, and Ritter (2003) developed a method using Morphological Associative Memories to extract endmembers. The image pixels are all shifted by their mean, $X' = \{\mathbf{x}'_i | \mathbf{x}'_i = \mathbf{x}_i - \mu\}$. Then, the algorithm begins by randomly selecting a single input pixel to be the initial endmember. Using this initial endmember's binary representation, min and max auto-associative memories are created. The binary representation of pixel \mathbf{x} is defined to be $sgn(\mathbf{x})$ (Grana and Gallego, 2003; Grana et al., 2003).

After selecting the initial pixel, all other image pixels are sequentially considered to be endmembers. When being considered, a pixel is shifted and tested for *morphological independence* against all of the current endmembers' binary representations (Grana et al., 2003).

$$\mathbf{x}_i^+ = \mathbf{M}_{BB} \boxtimes (\mathbf{x}'_i - a\sigma) \quad (2-21)$$

$$\mathbf{x}_i^- = \mathbf{W}_{BB} \boxtimes (\mathbf{x}'_i + a\sigma) \quad (2-22)$$

where a is a constant value, \mathbf{B} matrix of the current endmembers' binary representations, and σ is the vector of variances of each band of the input image. A pixel is determined to be morphologically independent if $\mathbf{x}_i^+ \notin B$ and $\mathbf{x}_i^- \notin B$. If a shifted pixel is found to be morphologically independent, it is added to the set of endmembers and new auto-associative memories are computed. If it is not morphologically independent, then the pixel is compared against existing endmembers to see if it is more extreme than the current endmember. If the pixel is more extreme, then it replaces that endmember; otherwise, the pixel is discarded (Grana et al., 2003). Using D -length vectors, there are 2^D possible binary vectors. This algorithm will return one endmember for each set of shifted input data points with the same binary representations. Therefore, up to 2^D endmembers may be returned.

Although this algorithm does not require prior knowledge of the number of endmembers, the algorithm assumes that pure pixels exist in the input hyperspectral image and it does not compute abundances values. Like NFindr, this algorithm relies on the pixel purity assumption. This method differs from NFindr since the volume encompassed by the endmembers is not computed while performing endmember detection.

Morphological associative memory method 2. Grana, Hernandez, and d’Anjou (2005) developed an algorithm that combines an evolutionary search and endmember detection using Morphological Associative Memories. This algorithm uses an evolutionary search to find a set of morphologically independent endmembers that minimize the fitness function in Equation 2–25 (Grana et al., 2005). The algorithm proceeds by evolving a set of binary vectors using a mutation operator and *roulette wheel selection* based on the fitness function (Grana et al., 2005). Every mutation is tested for morphological independence. If the mutated set of binary vectors is not morphologically independent, it is rejected. Given a set of binary vectors, the corresponding endmembers are the extreme pixels in the direction identified by the binary vectors (Grana et al., 2005).

This algorithm provides both endmember spectra and abundance values. However, this algorithm requires prior knowledge of the desired number of endmembers and does not strictly enforce the non-negativity and sum-to-one constraints on the abundance values.

Morphological associative memory method 3. Ritter and Urcid (2008) developed a method of extracting endmembers that uses the columns of the min auto-associative memories in Equation 2–18. This method is similar to the one presented by Myers (2005) which also returns the *strong lattice independent* columns of the min or max auto-associative memories. The auto-associative memories in this method are created using the points of the input hyperspectral image.

After computing the auto-associative memories, any duplicate columns of the min memory are removed ensuring that the remaining column are linearly independent as

proven by [Ritter and Urcid \(2008\)](#). Linearly independent sets are created since linear independence implies affine independence. The members of the linearly independent sets are then shifted by the elements of the *bright point* (the component-wise maximum of all input points)

$$\mathbf{w}^j = \mathbf{W}_{XX}^j + \mathbf{u}^j \quad (2-23)$$

where \mathbf{W}_{XX}^j is the j^{th} column of the \mathbf{W}_{XX} memory and $\mathbf{u}^j = \bigvee_{i=1}^N \mathbf{x}_i^j$ is the maximum value in the j^{th} spectral band over all input data points ([Ritter and Urcid, 2008](#)). The elements along the diagonal of the memory are equal to zero. Therefore, after shifting, these values are set to the maximum value of the data points in the corresponding hyperspectral band. This provides physical meaning between the endmembers and the input data set ([Ritter and Urcid, 2008](#))

After shifting, the unique linearly independent vectors, $\mathbf{W} = \{\mathbf{w}^1, \dots, \mathbf{w}^2\}$, are returned as endmembers. Additionally, the *shade point* (the component-wise minimum of all input points) is returned as an endmember ([Myers, 2005](#); [Ritter and Urcid, 2008](#)). Using the min-memory and the shade point provides up to $D + 1$ endmembers.

This method is very efficient; it requires only a single pass through all of the input pixels ([Myers, 2005](#); [Ritter and Urcid, 2008](#)). However, this algorithm does not compute abundance values and it does not guarantee that all pixels will be encompassed by the selected endmembers. Figure 2-4 displays the endmembers determined using this method on two-dimensional data. The data set was generated from four endmembers ($[10, 30]$, $[13, 24]$, $[15, 31]$, $[22, 25]$). Also, Gaussian random noise was added to each coordinate of the data set. The min memory, prior to shifting, found endmembers $(0, 1.38)$ and $(-21.42, 0)$. These were shifted by (the 1^{st} and 2^{nd} coordinates of the *bright point*, the component-wise max of the data) 22.64 and 32.97, respectively, to obtain $(22.64, 24.02)$ and $(11.55, 32.97)$. As shown in the figure, all of the pixels are not encompassed by the endmembers.

Using the bright point, the shade point and the unique columns of both the min- and max-memories as endmembers guarantees that all input data points will be encompassed

(Ritter and Urcid, 2008). This is shown in Figure 2-5. However, this does not provide a tight fit around the data points and it will return up to $2D + 2$ endmembers. With high dimensional data sets, the method would return a very large number of endmembers. Methods to reduce the number of endmembers when using either both memories or only the min-memory are discussed by Ritter and Urcid (2008). For example, every other column of the memory may be discarded since contiguous columns are often highly correlated. Another method presented is to compute linear correlation coefficients between each of the endmembers and retain a subset of endmembers whose correlation coefficients fall below a set threshold (Ritter and Urcid, 2008).

2.1.5 Evolutionary Search

In addition to the Morphological Associative Memory method which incorporates evolutionary search strategies to perform hyperspectral endmember detection, the single individual evolutionary strategy (SIE) for endmember detection also uses an evolutionary algorithm to determine an endmember set for a given hyperspectral image (Grana, Hernandez, and Gallego, 2004). The SIE algorithm begins by sampling M endmembers from a Gaussian distribution centered at the mean of the hyperspectral data set

$$\mathbf{e}_i \sim \mathcal{N}(\mathbf{m}, \text{diag}(\sigma)) \quad (2-24)$$

where \mathbf{m} is vector containing the mean of each band of the input image and $\text{diag}(\sigma)$ is a diagonal covariance matrix with the elements equal to the variance of each band of the input image. The initial *mutation variances* are set to the variance of each band in the input data set. The mutation variances are used to create the distribution from which mutations for the evolutionary step of the algorithm are generated.

Given the initial set of endmembers, the *global population fitness* can be computed.

$$F(\mathbf{E}, \mathbf{X}) = \sum_{i=1}^N \|\mathbf{x}_i - \mathbf{E}^T \mathbf{p}_i\|^2 + \sum_{i=1}^N (1 - |\mathbf{p}_i|)^2 + \sum_{i=1}^N \sum_{\{k: p_{ik} < 0\}} |p_{ik}| \quad (2-25)$$

where $\mathbf{p}_i = (\mathbf{E}^T \mathbf{E})^{-1} \mathbf{E}^T \mathbf{x}_i$ is the vector of unconstrained abundance values for the pixel \mathbf{x}_i . In addition to the global fitness function, individual fitness functions, $F(\mathbf{e}_i, \mathbf{X})$, are computed for each endmember using only that endmember's abundance fractions.

The algorithm proceeds by iteratively selecting endmembers for mutation. An endmember is chosen for mutation based on its individual fitness function using *roulette wheel* selection (Grana et al., 2004; Whitley, 2001). Once chosen, an endmember is mutated by adding a random Gaussian perturbation.

$$\mathbf{e}_i^{j+1} = \mathbf{e}_i^j + \zeta_i \quad (2-26)$$

where

$$\zeta_i \sim \mathcal{N}(\mathbf{0}, \sigma_i^{j+1}) \quad (2-27)$$

$$\sigma_i^{j+1} = \sigma_i^j \exp(\tau \cdot \xi) \quad (2-28)$$

$$\xi \sim \mathcal{N}(0, 1) \quad (2-29)$$

τ is a step size constant. After mutating the endmember, the global fitness function in Equation 2-25 is recomputed. If the fitness function improves, then the mutated endmember replaces the original endmember. An endmember can be mutated up to λ times.

This algorithm searches for endmembers that minimize the squared error between the pixels and their estimation and minimize the amount abundance values are negative or do not prescribe to the sum-to-one constraint. Like many of other of existing endmember detection algorithms, this algorithm requires the input of the number of desired endmembers.

2.1.6 Independent Components Analysis

Several methods have been developed based on Independent Component Analysis (ICA) (Chiang, Chang, and Ginsberg, 2000; Tu, 2000; Tu, Huang, and Chen, 2001; Wang and Chang, 2006). Independent Component Analysis performs unsupervised separation of statistically independent sources in a data set (Nascimento and Bioucas-Dias, 2005a). The

data points, $\{\mathbf{x}_i\}_{i=1}^N$, are assumed to be linear mixtures of these independent components (Hyvarinen and Oja, 2000).

$$\mathbf{x}_i = \mathbf{E}\mathbf{p}_i \quad (2-30)$$

One method of determining components is by minimizing the mutual information between sources and ensuring sources effectively describe the data using $\mathbf{E}\mathbf{p}$ (Hyvarinen and Oja, 2000). Equivalently, ICA can be performed by searching for the components which match the data and are “non-Gaussian” (Hyvarinen and Oja, 2000). In ICA, the number of signal sources found is the same as the dimensionality of the data. Therefore, either D signal sources are found where D is the data dimensionality, or dimensionality reduction is used to find $M \leq D$ signals (Hyvarinen and Oja, 2000; Tu et al., 2001).

Chiang et al. (2000) directly apply ICA to the problem of determining the endmembers for a hyperspectral image where the abundance values are assumed to be statistically independent “random signal sources”. Tu (2000) also applied the ICA algorithm for endmember extraction. Prior to running ICA, Tu estimates the number of endmembers and whitens the data to reduce the data dimensionality using the Noise-Adjusted Transformed Gerschgorin disk (NATGD) (Tu, 2000). Similarly, the Spectral Data Explorer algorithm (SDE) uses the Partitioned Noise-Adjusted Principal Components Algorithm (PNAPCA) to whiten the data and determine the number of endmember after which ICA is performed (Tu et al., 2001). NATGD and PNAPCA methods for estimating the number of endmembers are described in Section 2.1.7. Wang and Chang (2006) apply ICA for endmember extraction where, prior to running ICA, the number of endmembers, M , is estimated using the Virtual Dimensionality (described in Section 2.1.7) of the data set. Given the estimated number of endmembers, the independent components determined using the ICA algorithm are prioritized using the 3rd and 4th order statistics of the component (Wang and Chang, 2006). For each of the M highest priority components, image pixels with the largest absolute value of the abundance are returned as endmembers (Wang and Chang, 2006).

Nascimento and Bioucas-Dias (2005a) argue that the use of Independent Components Analysis for endmember detection is not an accurate method since the sum-to-one constraint on the abundance values cause the sources in the corresponding ICA problem to be dependent. This dependency violates the basic ICA assumption of statistically independent sources (Nascimento and Bioucas-Dias, 2005a). Nascimento and Bioucas-Dias (2005a) provide results to argue that the some endmembers are incorrectly unmixed using ICA methods. As an alternative, the Dependent Component Analysis method (DECA) was developed which assumes the abundance values are drawn from a Dirichlet distribution (Nascimento and Bioucas-Dias, 2007a,b). The Dirichlet enforces the non-negativity and sum-to-one constraints on the abundance values. DECA determines abundances and endmember values using the Expectation-Maximization (EM) method (Nascimento and Bioucas-Dias, 2007a,b). However, like ICA-based methods, DECA also requires the number of endmembers to be known in advance.

2.1.7 Estimating the Number of Endmembers

The concept of *virtual dimensionality* is used by some methods, prior to endmember extraction, to estimate the number of endmembers for a given scene (Chang and Du, 2004; Wang and Chang, 2006). Also, Tu (2000) relies on the Transformed Gerschgorin Disk (TGD) and the Noise-Adjusted TGD method of estimating the number of endmembers. The Partitioned Noise-Adjusted Principal Components Analysis (PNAPCA) method of computing the number of endmembers is based on partitioning and transforming the noise-adjusted covariance matrix (Liou and Yang, 2005; Tu et al., 2001).

Virtual dimensionality. Virtual Dimensionality (VD) is defined as the “minimum number of spectrally distinct signal sources” in a hyperspectral data set (Chang and Du, 2004). VD is computed using the eigenvalues of the covariance and correlation matrices of the input data set. Let $\{\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \dots \geq \hat{\lambda}_d\}$ be the eigenvalues from the sample correlation matrix and let $\{\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d\}$ be the eigenvalues from the sample

covariance matrix. The VD is computed as the number of corresponding correlation and covariance eigenvalues that differ (Chang and Du, 2004).

$$\hat{\lambda}_r \geq \lambda_r, \quad \text{for } r = 1, \dots, VD \quad (2-31)$$

$$\hat{\lambda}_r = \lambda_r, \quad \text{for } r = VD + 1 \dots, d \quad (2-32)$$

The eigenvalues computed from the sample covariance matrix equal the variances of the transformed data (Theodoridis and Koutroumbas, 2003) and the eigenvalues of the sample correlation matrix are related to the variance of the data from the origin. Assuming that the noise has zero-mean and unit-variance and signals in the data have non-zero values, the eigenvalues of the sample correlation matrix corresponding to signals in the data will have a larger value than the corresponding eigenvalues from the sample covariance matrix. The eigenvalues of the sample correlation and covariance matrix corresponding to noise will be equal (Chang and Du, 2004),

$$\hat{\lambda}_r \geq \lambda_r \geq \sigma_{n_i}^2 \quad \text{for } r = 1, \dots, VD \quad (2-33)$$

$$\hat{\lambda}_r = \lambda_r = \sigma_{n_i}^2 \quad \text{for } r = VD + 1 \dots, d \quad (2-34)$$

where $\sigma_{n_i}^2$ is the noise variance.

An example of this concept is shown using the data in Figure 2-6. The three-dimensional data set was generated using two endmembers, [2, 5, 0] and [3, 6, 1]. Zero-mean Gaussian noise with a variance of 0.03 was added to each coordinate of the data. The eigenvalues of the covariance and correlation matrices were computed. The eigenvalues from the covariance matrix were found to be 0.1814, 0.0010, and 0.0008. The eigenvalues for the correlation matrix were found to be 36.4991, 0.0647, and 0.0008. Therefore, the virtual dimensionality correctly determines the number of endmembers to be two, since the third eigenvalue from both the covariance and correlation matrices are equal.

In order to determine that the eigenvalues differ, their differences are thresholded. Chang and Du (Chang and Du, 2004) describe three thresholding methods for determining the virtual dimensionality. These methods include the Harsanyi-Farrand-Chang (HFC)

method, the Noise Whitened HFC (NWHFC) method, and the Noise Subspace Projection (NSP) method.

The HFC method thresholds the differences of the covariance and correlation eigenvalues based on a Neyman-Pearson detector. The probability of detection is maximized while the probability of false alarm is held to a constant value, a .

$$P_F = \int_{\tau_l}^{\infty} \mathcal{N}(0, \sigma_{z_l}^2) dz = a \quad (2-35)$$

$$P_D = \int_{\tau_l}^{\infty} \mathcal{N}(\mu_l, \sigma_{z_l}^2) dz \quad (2-36)$$

Solving for τ_l gives the threshold for the eigenvalue differences. This thresholding method requires an estimate of the variance of the difference between the eigenvalues at each band, σ_{z_l} . The HFC method uses $\sigma_{z_l}^2 \approx \frac{2\lambda_l^2}{N} + \frac{2\lambda_l^2}{N}$ as the estimate of this variance.

The HFC method assumes that the data has white noise with zero mean. The Noise-Whitened HFC method (NWHFC) attempts to improve VD estimates by performing noise-whitening to the correlation and covariance matrices prior to computing the differences between their eigenvalues. The NWHFC method requires an estimate of the noise covariance. After estimating the noise covariance and whitening the correlation and covariance matrices, the thresholds are computed using the same method as HFC (Chang and Du, 2004).

The Noise Subspace Projection method is similar to the NWHFC. An estimate of the noise covariance for the data is used to whiten the covariance matrix. However, instead of computing differences. This method recognizes that the eigenvalues corresponding to noise should be equal to one. Therefore, the computed threshold is applied directly to the eigenvalues to determine if their difference from one is significant (Chang and Du, 2004).

This method is sensitive to the variance and covariance estimates used in determining the thresholds for each eigenvalue. Therefore, the VD estimate of the number of endmembers is sensitive to noise in the input data set. The VD was run on two data sets generated from the three AVIRIS Cuprite spectra shown in Figure 2-7. The a small

amount of Gaussian noise added to the first data set, shown in Figure 2-8. The second data set, shown in Figure 2-9, has more added noise. The three thresholding methods were applied to both sets of data. On the first data set, the NSP method correctly determined the number of endmembers by estimating 3 signals. The HFC and NWHFC methods incorrectly estimated the number of endmembers with 2 and 5, respectively. On the second data set with larger amounts of noise, none of the thresholding methods correctly estimated three endmembers. The HFC method estimated 2, the NWHFC method estimated 7, and the NSP method estimated 2 endmembers.

Maximum noise fraction. The Partitioned Noise Adjusted Principal Components Analysis (PNAPCA) method (Tu et al., 1999, 2001) of estimating the number of endmembers is based on the Maximum Noise Fraction (MNF) (also known as the Noise Adjusted Principal Components Analysis method) (Green et al., 1988; Lee et al., 1990).

The MNF transform uses an estimate of the noise covariance matrix to transform the data into components which are sorted based on their signal-to-noise ratio. Using the convex geometry model in Equation 1-1 and assuming the noise and signal components of the data are uncorrelated, the covariance matrix of the data can be written as (Lee et al., 1990; Tu et al., 1999)

$$\begin{aligned}\Sigma_X &= E \left[\sum_{i=1}^N \mathbf{x}'_i \mathbf{x}_i'^T \right] = E \left[\sum_{i=1}^N (\mathbf{E}' \mathbf{a}_i^T) (\mathbf{E}' \mathbf{a}_i^T)^T \right] + E \left[\sum_{i=1}^N \boldsymbol{\epsilon}_i \boldsymbol{\epsilon}_i^T \right] \\ &= \Sigma_S + \Sigma_N\end{aligned}\tag{2-37}$$

where $X' = \{\mathbf{x}'_i | \mathbf{x}'_i = \mathbf{x}_i - \mu\}$. The MNF determines a transformation which maximizes the signal-to-noise ratio (Lee et al., 1990; Tu et al., 1999),

$$\begin{aligned}\arg \max_{\mathbf{W}} SNR &= \arg \max_{\mathbf{W}} \frac{\mathbf{W}^T \Sigma_S \mathbf{W}}{\mathbf{W}^T \Sigma_N \mathbf{W}} \\ &= \arg \max_{\mathbf{W}} \frac{\mathbf{W}^T \Sigma_X \mathbf{W}}{\mathbf{W}^T \Sigma_N \mathbf{W}} - 1.\end{aligned}\tag{2-38}$$

The signal-to-noise ratio is maximized by assigning $\mathbf{W} = \Phi_N \Lambda_N^{-\frac{1}{2}} \Phi_A$ where Φ_N is the eigenvector matrix of the noise covariance matrix, Λ_N is the diagonal eigenvalue matrix of

the noise covariance matrix, and Φ_A is the eigenvector matrix of noise-adjusted covariance matrix, $\Sigma_A = \left(\Phi_N \Lambda_N^{-\frac{1}{2}}\right)^T \Phi_X \left(\Phi_N \Lambda_N^{-\frac{1}{2}}\right)$ (Lee et al., 1990; Tu et al., 1999). Using the matrix \mathbf{W} , the MNF transform simultaneously diagonalizes the data covariance matrix and whitens the noise covariance matrix.

The MNF transformation requires an estimate of the noise covariance matrix, Σ_N (Tu et al., 1999). As described by Tu et al. (1999, 2001), PNAPCA method partitions the noise adjusted covariance matrix found by MNF and diagonalizes the two partitions. Tu et al. (1999, 2001) claim that by examining the eigenvalues of the two partitions simultaneously, the effects of incorrectly estimating the noise covariance matrix are lessened.

Transformed Gerschgorin disk and the noise adjusted transformed Gerschgorin disk. Wu et al. (1995) and Tu (2000) developed methods of estimating the number of signals in a data set based on Gerschgorin's disk theorem (Horn and Johnson, 1985). Gerschgorin's disk theorem provides a method of estimating the locations of eigenvalues of a matrix. The theorem states that the eigenvalues of a matrix, \mathbf{A} , are located within the union of the disks defined by

$$G_i = \{z : |z - a_{ii}| \leq r_i\} \quad (2-39)$$

where

$$r_i = \sum_{j=1, j \neq i}^D |a_{ij}| \quad (2-40)$$

and $a_{ii}, i = 1, \dots, D$, the centers of the Gerschgorin disks, are the elements along the diagonal of matrix \mathbf{A} (Horn and Johnson, 1985). The theorem also states that if the union of k of the D disks form a connected region and if the connected region is disjoint from all of the remaining disks, then k eigenvalues are located within the region defined by the union of the k disks (Horn and Johnson, 1985).

The transformed Gerschgorin Disk method developed by Wu et al. (1995) defines a transformation on the covariance matrix of an input data set so that the Gerschgorin disks

associated with noise in the data set have small radii and are located far from the signal disks. In other words, the goal of this transformation is to cause the disks containing eigenvalues corresponding to signals in the data to have large radii. [Tu \(2000\)](#) applies the transformed Gerschgorin disk method to the noise-adjusted covariance matrix.

The transformation matrix used in the Transformed Gerschgorin Disk and the Noise-Adjusted Transformed Gerschgorin Disk methods is determined by diagonalizing the $D - 1 \times D - 1$ leading sub-matrix of the input covariance matrix,

$$\mathbf{\Sigma}_X = \begin{bmatrix} \mathbf{C}_1 & \mathbf{c} \\ \mathbf{c}^T & c_{DD} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_1^T \mathbf{V}_1 \mathbf{U}_1 & \mathbf{c} \\ \mathbf{c}^T & c_{DD} \end{bmatrix} \quad (2-41)$$

where \mathbf{C}_1 is the $D - 1 \times D - 1$ leading sub-matrix of the input covariance matrix, $\mathbf{c}^T = [c_{1D}, \dots, c_{(D-1)D}]$, \mathbf{U}_1 is the matrix of eigenvectors of \mathbf{C}_1 and \mathbf{V}_1 is the diagonal matrix of eigenvalues. Given the matrix of eigenvectors of the leading sub-matrix, \mathbf{U}_1 , the following transformation, \mathbf{U} , is defined and applied to the input covariance matrix,

$$\mathbf{U} = \begin{bmatrix} \mathbf{U}_1 & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \quad (2-42)$$

$$\mathbf{U}^T \mathbf{\Sigma}_X \mathbf{U} = \begin{bmatrix} \mathbf{U}^T \mathbf{C}_1 \mathbf{U} & \mathbf{U}^T \mathbf{c} \\ \mathbf{c}^T \mathbf{U} & c_{DD} \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & 0 & . & . & . & 0 & \rho_1 \\ 0 & \lambda_2 & 0 & . & . & . & 0 & \rho_2 \\ 0 & 0 & \lambda_3 & . & . & . & 0 & \rho_3 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ 0 & . & . & . & . & . & \lambda_{D-1} & \rho_{D-1} \\ \rho_1 & \rho_2 & \rho_3 & . & . & . & \rho_{D-1} & c_{DD} \end{bmatrix}. \quad (2-43)$$

Using the Gerschgorin disk theorem in Equation 2-39, the Gerschgorin disks of the transformed input covariance matrix in Equation 2-43 have radii equal to $|\rho_1|, |\rho_2|, \dots, |\rho_{D-1}|$ and centers at $\lambda_1, \lambda_2, \dots, \lambda_{D-1}$. Assuming that the noise in the

data is uncorrelated and has zero mean, the radii associated with noise will be equal to zero. Therefore, the Transformed Gerschgorin Disk method returns, as the number of endmembers, the number of large radii.

The Noise Adjusted Transformed Gerschgorin Disk method applies this method to the noise-adjusted covariance matrix. This requires an estimate of the noise covariance matrix which is used to whiten the noise in the input data set. Like the VD and NAPCA methods of estimating the number of endmembers, the Transformed Gerschgorin Disk methods are sensitive to noise.

2.2 Existing Hyperspectral Band Selection Algorithms

In addition to endmember extraction, a hyperspectral band selection method that determines the required number of bands, performs unsupervised band selection, and retains bands that help to distinguish between endmembers in a scene is proposed. The presented method performs these tasks while simultaneously determining endmembers and the number of endmembers needed.

Many data reduction techniques such as Principal Components Analysis (PCA) and Maximum Noise Fraction transform (MNF) ([Green et al., 1988](#); [Lee et al., 1990](#); [Theodoridis and Koutroumbas, 2003](#)) have been used to project the data into a lower dimensional space and thus reduce the dimensionality of the data. Although these methods are effective at data reduction, they do not retain physically meaningful bands that correspond to wavelengths in the original data set. [Harsanyi and Chang \(1994\)](#) provide an orthogonal subspace projections approach that also transforms the data. [Bruce et al. \(2002\)](#) conducts dimensionality reduction by extracting features that distinguish between labeled classes in a training set using the Discrete Wavelet Transform (DWT). This method extracts features that incorporate both frequency information and detailed localized features of the input hyperspectral signal. This feature set is further reduced using Fisher’s Linear Discriminant Analysis. Since this method using the DWT and Fisher’s Linear Discriminant, extracted features do not correspond to wavelengths in the

original data set. [DeBacker et al. \(2005\)](#) and [Kumar et al. \(2001\)](#) both present methods that merge many adjacent hyperspectral bands. [DeBacker et al. \(2005\)](#) merges bands into groups which optimize the Bhattacharya distance between labeled classes in a training set. The supervised band selection method presented by [Riedmann and Milton \(2003\)](#) merges neighboring bands to improve accuracy in a classification task. The hyperspectral dimensionality reduction method based on Localized Discriminant Bases ([Venkataraman et al., 2005](#)) also merges adjacent bands for feature extraction. [Martinez-Uso et al. \(2007\)](#) present a band merging algorithm using information measures and hierarchical clustering. A divergence measure between every pair of bands is computed and used to perform hierarchical clustering of the bands. A band representative is then computed for each cluster. [Lin and Bruce \(2004\)](#) use a Projection Pursuits methods to reduce the dimensionality of a hyperspectral data set by determining a projection matrix that aids in distinguishing between classes in the data set. Instead of merging bands or transforming the data, this method maintains only those bands that are useful for the hyperspectral image analysis task. The advantage of physically meaningful bands is to identify useful wavelengths for a particular classification task. Identifying important wavelengths can also be used in the design of hyperspectral sensors. By reducing the number of wavelengths that need to be collected, data collection will be performed faster and with less required storage space.

Additionally, most of the previously mentioned band selection algorithms ([DeBacker et al., 2005](#); [Green et al., 1988](#); [Harsanyi and Chang, 1994](#); [Lee et al., 1990](#); [Martinez-Uso et al., 2007](#)) require the knowledge of the desired number of bands. Serpico and colleagues' search method for band selection, Du and colleagues' method of band prioritization based on the Independent Component Analysis' weight matrix, Han and colleagues' eigenvalue weighted band prioritization method, and Guo and colleagues' ([Du et al., 2003](#); [Guo et al., 2006](#); [Han et al., 2004](#); [Serpico and Bruzzone, 2001](#)) mutual information based band selection method require the desired number of bands. [Petrie et al. \(1998\)](#) outlines

four band selection strategies based on maximizing spatial autocorrelation, maximizing a distance measure between targets, or merging neighboring bands. All of the methods described by [Petrie et al. \(1998\)](#) require the desired number of bands. The band selection method based on the NFindr algorithm ([Wang et al., 2006](#)) retains the bands which maximize the volume between the endmembers found using the NFindr algorithm ([Winter, 1999](#)). This method attempts to find bands which aid in spectral unmixing, however, the number of bands to retain must be known in advance. Often, the number of required bands is not known.

Keshava presented a method based on the Spectral Angle Mapper (SAM) distance or the Euclidean Minimum Distance (EMD) measures ([Keshava, 2001, 2004](#)). The algorithm incrementally adds bands that increase the SAM or EMD measure between two labeled classes until some stopping criterion is reached. Although this method does not require the number of bands in advance, the method is limited to distinguishing between two labeled classes in the data set. Similarly, the Sparse Linear Filters algorithm ([Theiler and Gloer, 2006](#)) develops sparse linear filters to distinguish between two labeled classes in the data. The filters use a sparse set of the hyperspectral bands by utilizing an L1-penalty term to select the bands and the number of bands ([Tibshirani, 1996](#)). Although the number of bands is estimated, the method requires two labeled classes ([Theiler and Gloer, 2006](#)). [Chang et al. \(1999\)](#) ranks all bands based on loading factors constructed using maximum-variance PCA, MNF, orthogonal subspace projection and minimum misclassification canonical analysis methods. Following ranking, [Chang et al. \(1999\)](#) remove correlated bands using a divergence measure. [Chang and Wang \(2006\)](#) employ a method based on constrained energy minimization. The method selects bands that have the minimal correlation with each other and uses the concept of virtual dimensionality to determine the number of chosen spectral bands.

2.3 Summary of Literature Review

The majority of endmember detection algorithms require the number of endmembers needed for a hyperspectral image prior to running the detection algorithm. Methods of estimating the number of endmembers in a data set, such as Virtual Dimensionality, are sensitive to noise in the data. Furthermore, many endmember detection algorithms assume that pure pixels for each endmember can be found in the data set. In highly mixed data sets, this assumption does not hold causing the algorithms to return mixed pixels as endmembers. Also, many existing endmember detection algorithms do not estimate abundance values that conform to the non-negativity and sum-to-one constraints in Equation 1–2. These algorithms also do not account for spectral variability in their endmember representations. Furthermore, the existing algorithms do not consider cases in which multiple convex regions and sets of endmembers may more accurately describe the data.

Hyperspectral band selection algorithms often require the number of needed hyperspectral bands prior to running the band selection algorithm. Data reduction techniques which perform projections are often used to reduce the dimensionality of a hyperspectral image. However, the projection methods lose the physical meaning associated with the hyperspectral bands. Hyperspectral band selection algorithms are also often tied to classification problems. In these cases, labeled training data is needed to determine the bands which distinguish between the classes.

Band selection and endmember detection methods presented autonomously determine the number of endmember and hyperspectral bands needed for an image. Methods are presented which account for an endmember’s spectral variability and can autonomously determine the number of convex regions needed to describe an input data set. Furthermore, all presented methods provide abundance values which conform to the constraints in Equation 1–2. The new algorithms are also capable of determining endmembers for highly mixed data sets since the pixel purity assumption is not employed. The presented hyperspectral band selection algorithm also retains physically meaningful

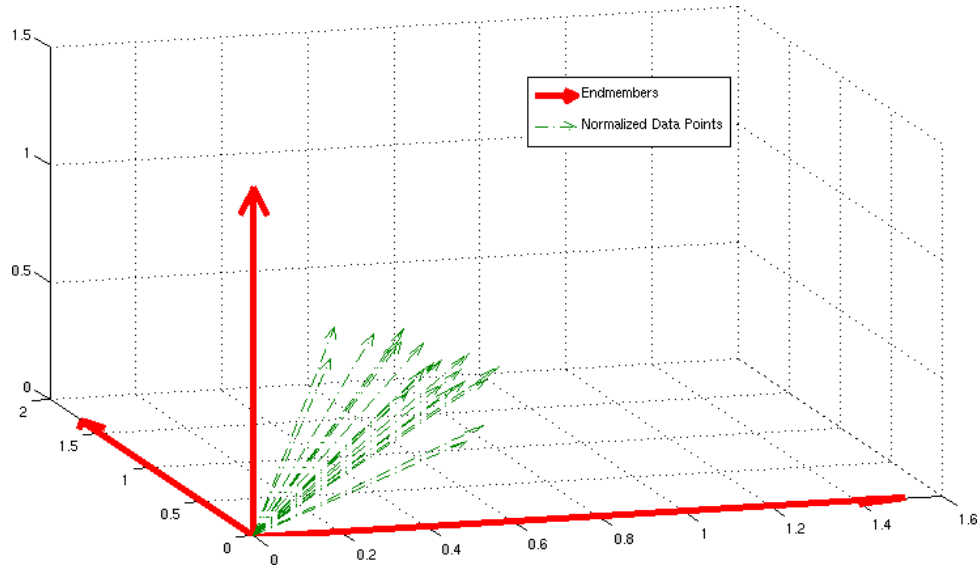


Figure 2-1. Three-dimensional data points and endmember results using convex cone analysis.

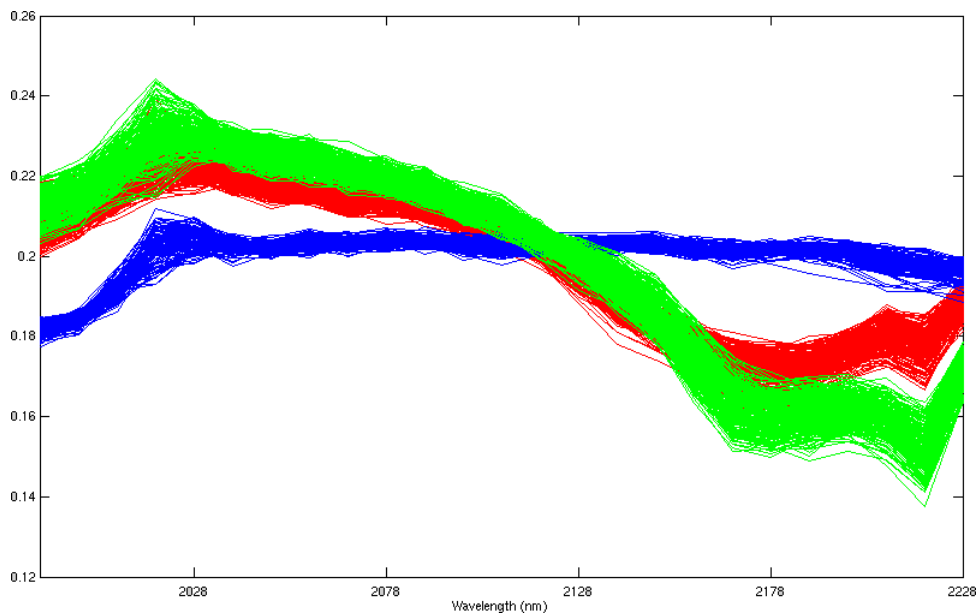


Figure 2-2. First 25 bands (1978 to 2228 nm) of a subset of normalized pixels from the AVIRIS cuprite “scene 4” data set

bands and does not require labeled training data. The method, instead, determines the hyperspectral bands which distinguish between the endmembers in a data set.

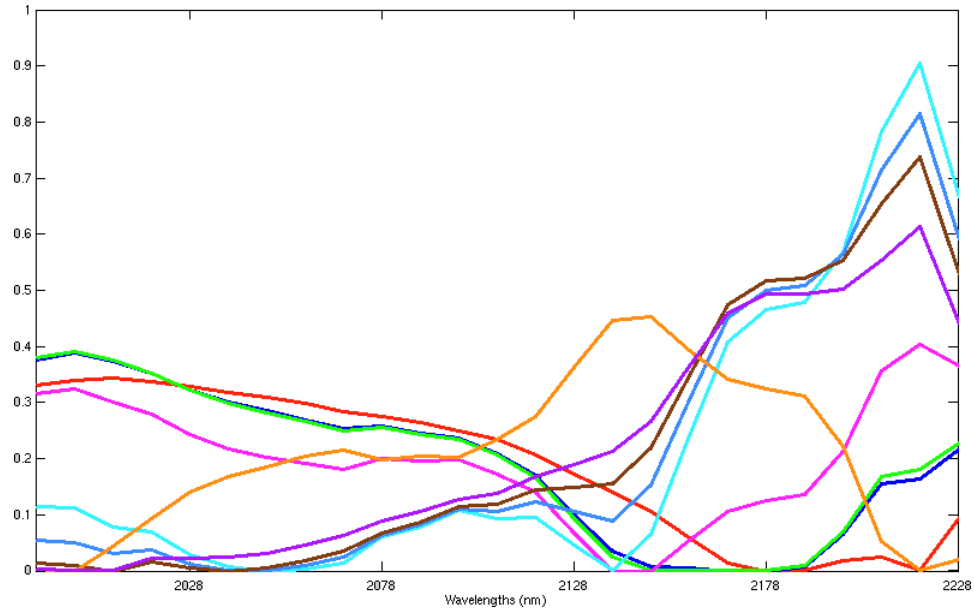


Figure 2-3. CCA endmember results on a subset of AVIRIS Cuprite data

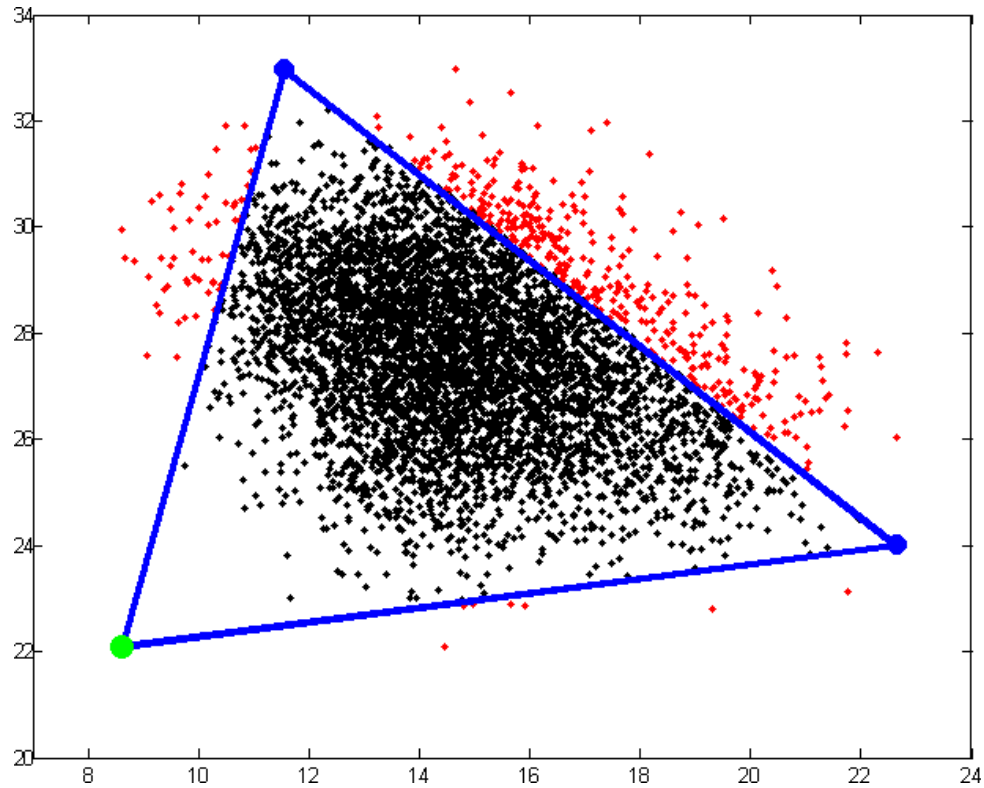


Figure 2-4. Morphological associative memories endmember results using the min memory on two-dimensional data. Endmembers found from the columns of the min memory are shown in blue. The shade point is green. Data points within the area defined by the endmembers are in black. Data points outside of the area defined by the endmembers are in red.

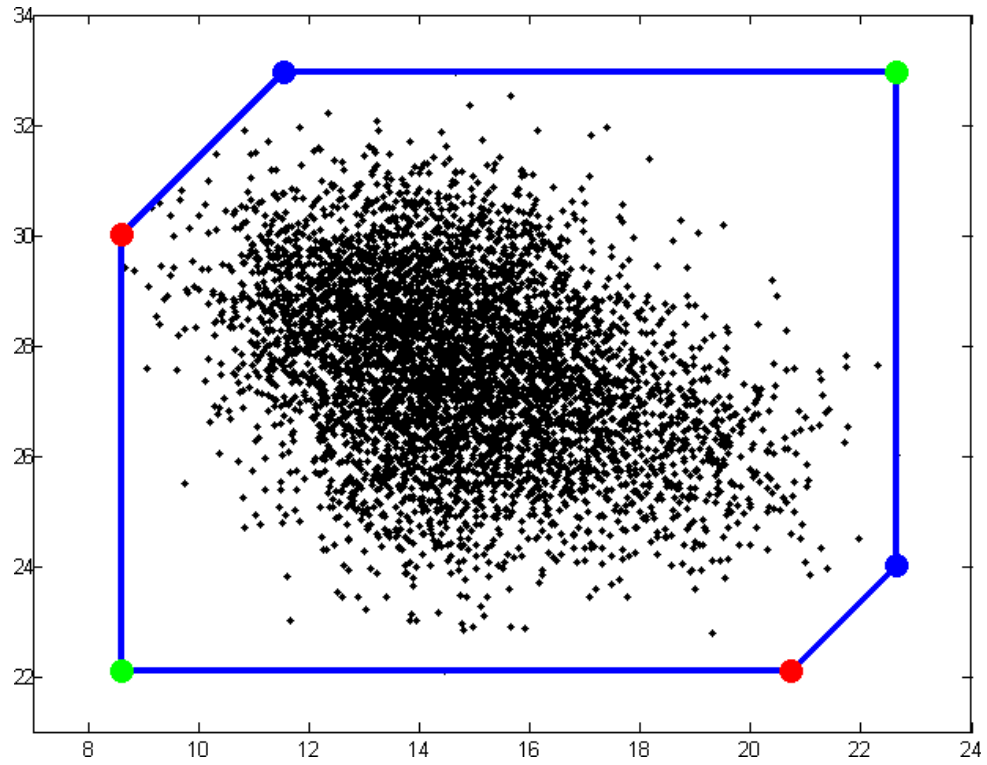


Figure 2-5. Morphological associative memories endmember results using both memories on two-dimensional data. Endmembers found from the columns of the max memory are shown in red. Endmembers from the columns of the min memory are shown in blue. The bright point and shade point are green.

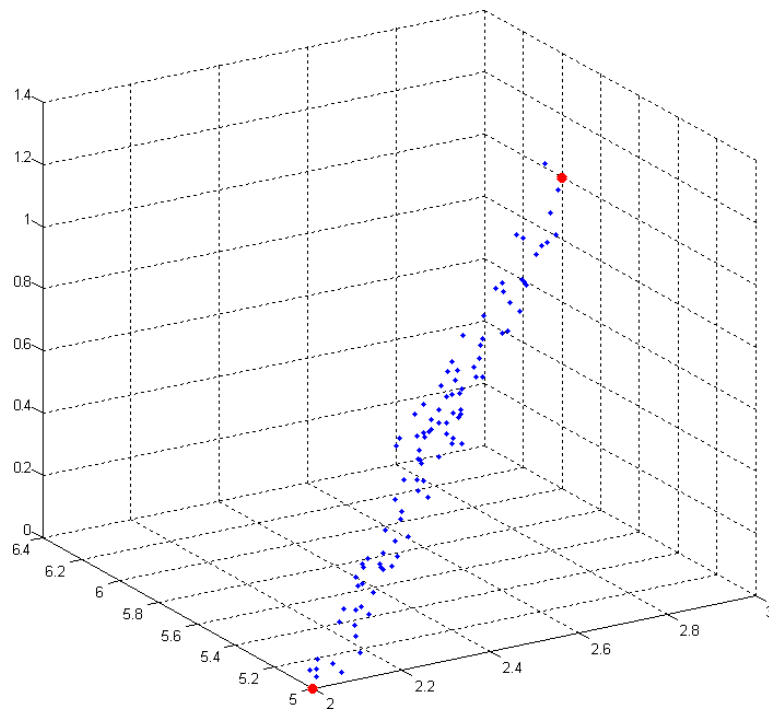


Figure 2-6. Three dimensional data set generated from two endmembers with Gaussian noise.

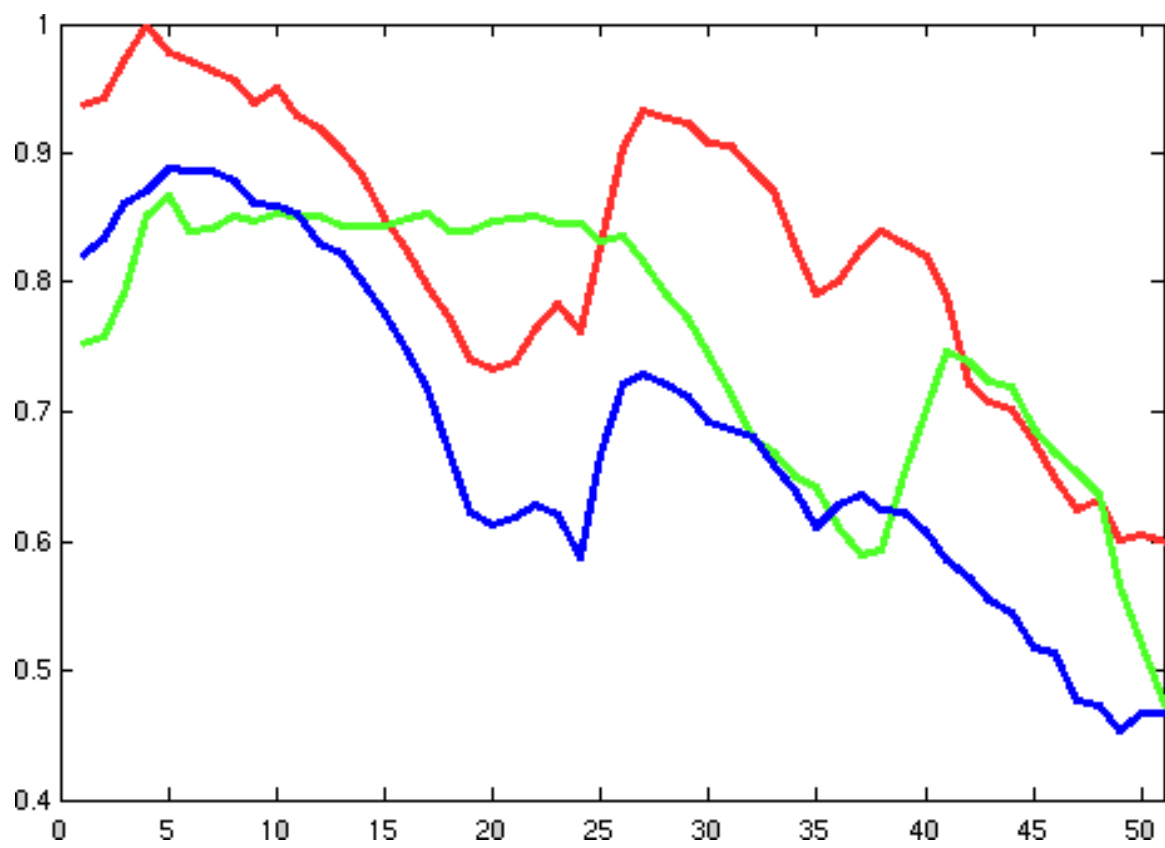


Figure 2-7. Normalized AVIRIS Cuprite spectra

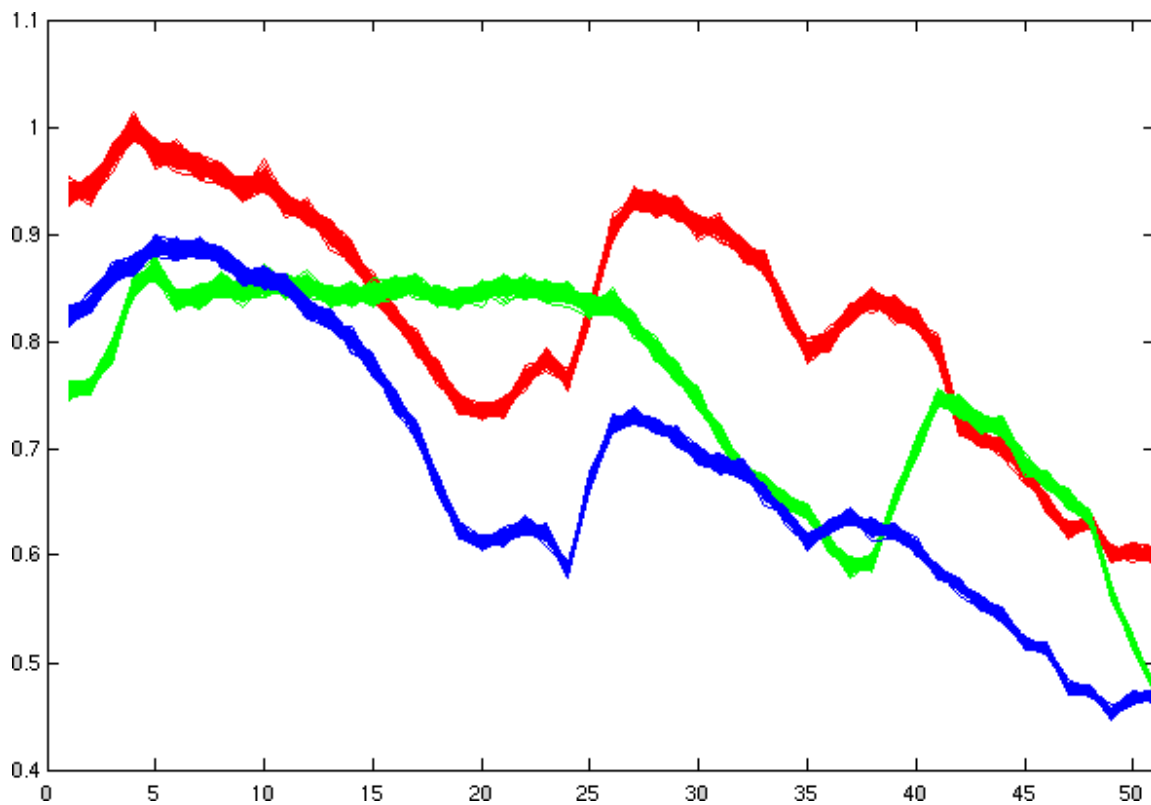


Figure 2-8. Data set generated from AVIRIS Cuprite endmembers with a small amount of Gaussian noise

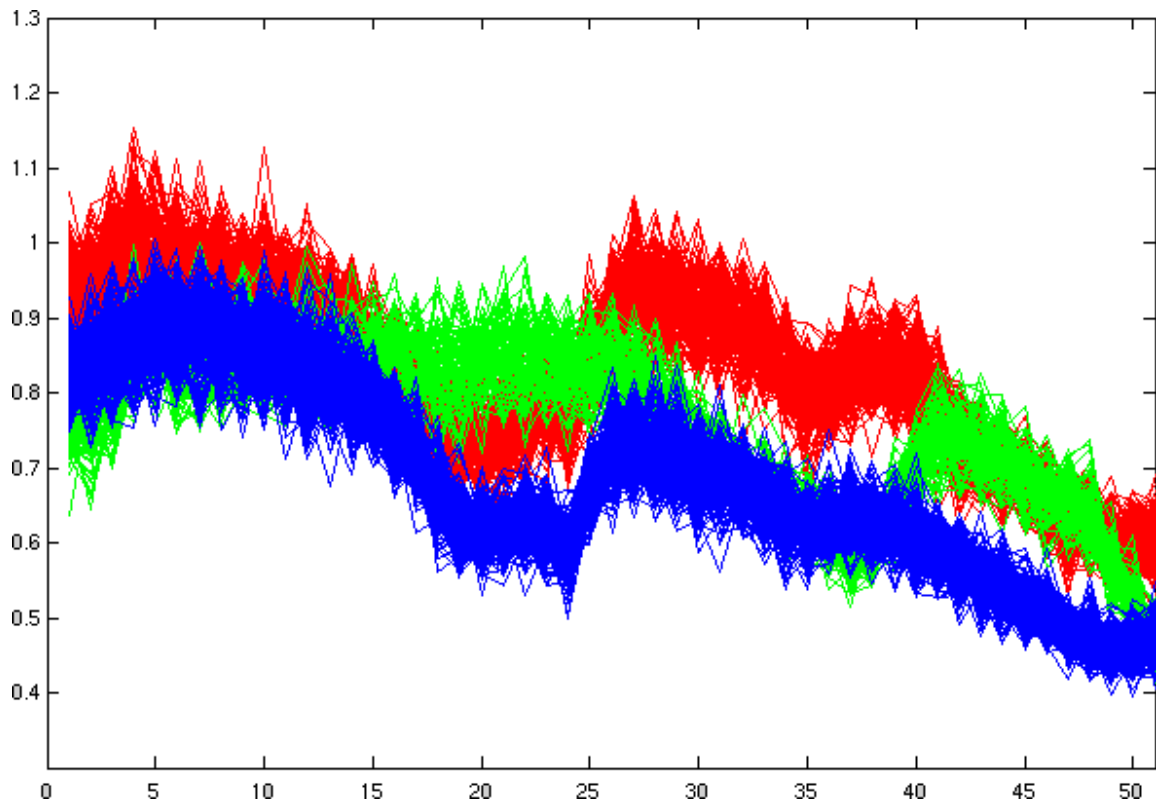


Figure 2-9. Data set generated from AVIRIS Cuprite endmembers with a large amount of Gaussian noise

CHAPTER 3

TECHNICAL APPROACH

The focus of the presented research is on a set of algorithms that utilize Bayesian methods to perform endmember detection and spectral unmixing simultaneously. To this end, four specific approaches are considered. These algorithms determine the number of endmembers, learn endmember distributions, determine the number of convex regions needed to describe the input hyperspectral data set, or determine the number of needed hyperspectral bands. Sparsity Promoting Iterated Constrained Endmembers (SPICE) is an endmember detection and spectral unmixing algorithm which uses sparsity promoting priors to remove unneeded endmembers. Band Selecting Sparsity Promoting Iterated Constrained Endmembers (B-SPICE) is an extension of the SPICE algorithm which incorporates band selection. The Endmember Distributions (ED) detection algorithm determines full endmember distributions for each endmember rather than single endmember spectra thus incorporating the spectral variation which occurs due to varying environmental conditions. The Piece-wise Convex Endmember (PCE) detection algorithm uses the Dirichlet process to autonomously determine the number of convex regions needed to describe a data set while simultaneously learning endmember distributions and abundances for each convex region. SPICE, B-SPICE, ED and PCE determine the spectral shape of the endmembers and compute abundance values which conform to the non-negative and sum-to-one constraints. All methods do not rely on the pixel purity assumption and are, therefore, capable of handling highly mixed data sets. Furthermore, the endmembers determined by these methods are capable of enveloping all of the hyperspectral data points while providing a tight fit around the data. The B-SPICE algorithm, in addition to determining the number of endmembers and their spectral shape, also performs band selection and determines the number of hyperspectral bands required. B-SPICE retains the physical meaning of each hyperspectral band and does not rely on a projection to perform data reduction. Also, B-SPICE does not require labeled data to

determine the useful hyperspectral bands but, instead, retains the bands which help to distinguish between the endmembers in the data set.

3.1 Review of Sparsity Promotion Techniques

The goal of sparsity promotion is to minimize the number of parameters. This is generally done by encouraging parameter values to be driven to zero and, thus, minimizing the number of non-zero parameter values. A common method to promote small parameter values is to add a weight decay term to the objective function (Williams, 1995). Weight decay terms has been previously applied in neural network applications to promote regularization (Haykin, 1999; Williams, 1995).

Consider a least squares objective with a weight matrix, \mathbf{P} . A weight decay term applied to the \mathbf{P} parameters attempts to prevent them from becoming large.

$$LSWD = \ln \exp \left\{ -\frac{1}{2} \sum_{i=1}^N \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right)^2 - \gamma \sum_{i=1}^N \sum_{k=1}^M p_{ik}^2 \right\} \quad (3-1)$$

$$= \ln \exp \left\{ -\frac{1}{2} \sum_{i=1}^N \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right)^2 \right\} \exp \left\{ -\gamma \sum_{i=1}^N \sum_{k=1}^M p_{ik}^2 \right\} \quad (3-2)$$

where $\gamma \geq 0$. Equation 3-2 can be interpreted in a probabilistic manner where the second exponential can be seen as a zero-mean Gaussian (Williams, 1995). Therefore, Equation 3-2 can be viewed as the log of the product in Equation 3-3 (Williams, 1995),

$$p(\mathbf{P}|\mathbf{X}) \propto p(\mathbf{X}|\mathbf{P}) p(\mathbf{P}) \quad (3-3)$$

where $p(\mathbf{X}|\mathbf{P})$ is the probability of the data given the parameters and $p(\mathbf{P})$ is the prior on the parameters.

Unfortunately, the Gaussian prior is not effective at sparsity promotion. The Gaussian does not prefer to set parameter values to zero which would promote sparsity. Instead, the Gaussian prefers several small valued non-zero parameters (Williams, 1995). Instead of using a Gaussian distribution for the parameters' prior, a zero-mean Laplacian distribution can be used which is more effective at sparsity promotion (Figueiredo, 2003; Williams,

1995),

$$LSSP = -\frac{1}{2} \sum_{i=1}^N \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right)^2 - \sum_{k=1}^M \gamma_k \sum_{i=1}^N |p_{ik}|. \quad (3-4)$$

The application of a Laplacian distribution is related the to *least absolute shrinkage and selection operator* (LASSO) (Tibshirani, 1996). As described by Tibshirani (1996), the LASSO applies the constraint that the sum of the absolute values of the weights, $\sum_{k=1}^M \sum_{i=1}^N |p_{ik}|$ must be less than a threshold. This is equivalent to applying a Laplacian prior (Tibshirani, 1996).

Often during optimization, the derivative of an objective function must be computed. In the case where the Laplacian prior has been used to penalize large parameter values, the derivative is not defined at zero due to the absolute value function (Williams, 1995). In these cases, the Laplacian prior on the parameters can be defined in a hierarchical fashion where the parameters are distributed according to a Gaussian distribution whose variance has an exponential hyper-prior (Figueiredo, 2003). By integrating over all possible values for the variance given the hyper-prior, the hierarchical expression is equivalent the to the Laplacian distribution (Figueiredo, 2003).

Sparsity promotion has been applied to a number of applications including neural networks (Williams, 1995), classification and regression using expectation-maximization (EM) (Figueiredo, 2003), feature selection and classification (Krishnapuram et al., 2004), classification and regression using the Choquet integral (Mendez-Vazquez and Gader, 2007) and others.

3.2 Review of the Iterated Constrained Endmembers Detection Algorithm

The presented methods for endmember detection and band selection using sparsity promoting priors are based on the Iterated Constrained Endmembers (ICE) algorithm. The ICE algorithm (Berman et al., 2004) performs a minimization of a residual sum of squares (RSS) term based on the convex geometry model in Equation 1-1. The error between the pixel spectra and the pixel estimate found by the ICE algorithm using the endmembers and their proportions is minimized when the residual sum of squares (RSS)

term is minimized (Berman et al., 2004),

$$RSS = \sum_{i=1}^N \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right)^T \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right). \quad (3-5)$$

As described by Berman et al. (2004), the minimizer for the RSS term is not unique. Therefore, the ICE algorithm adds a sum of squared distances (SSD) term to the objective function.

$$SSD = \sum_{k=1}^{M-1} \sum_{l=k+1}^M (\mathbf{E}_k - \mathbf{E}_l)^T (\mathbf{E}_k - \mathbf{E}_l) \quad (3-6)$$

This term is related to the volume bounded by the endmembers. Therefore, by adding this term to the objective function, the algorithm finds endmembers that provide a tight fit around the data. Berman et al. (2004) show that the SSD is equivalent to Equation 3-7,

$$SSD = M(M-1)V \quad (3-7)$$

where V is the sum of variances (over the bands) of the simplex vertices. ICE uses V in the objective function instead of $M(M-1)V$ in an effort to make this term independent of the number of endmembers, M (Berman et al., 2004).

The objective function used in the ICE algorithm is shown in Equation 3-8,

$$RSS_{reg} = (1 - \mu) \frac{RSS}{N} + \mu V \quad (3-8)$$

where μ is a regularization parameter that balances the RSS and SSD terms of the objective function.

The ICE algorithm minimizes this objective function iteratively. First, given endmember estimates, the proportions for each pixel are estimated. For the first iteration of the algorithm, endmember estimates may be set to randomly chosen pixels from the image. Estimating the proportions requires a least squares minimization of each term in Equation 3-5. Since each of these terms is quadratic and subjected to the linear constraints in Equation 1-2, the minimization is done using quadratic programming. After solving for the proportions, the endmembers are updated using the current proportion

estimates,

$$\mathbf{e}_j = \left\{ \mathbf{P}^T \mathbf{P} + \lambda \left(\mathbf{I}_M - \frac{\mathbf{1}\mathbf{1}^T}{M} \right) \right\}^{-1} \mathbf{P}^T \mathbf{x}_j \quad (3-9)$$

where \mathbf{P} is the $N \times M$ proportion matrix, \mathbf{e}_j is the vector of endmember values in the j^{th} band, \mathbf{x}_j is the vector of all the pixel values in the j^{th} band, \mathbf{I}_M is the $M \times M$ identity matrix, $\mathbf{1}$ is the M -vector of ones and $\lambda = N\mu/\{(M-1)(1-\mu)\}$. This iterative procedure is continued until the value of RSS_{reg} is smaller than a tolerance value. Although the ICE algorithm is an effective algorithm for finding endmembers when the number of endmembers is known, there is no automated mechanism in ICE to determine the correct number of endmembers.

3.3 New Endmember Detection Algorithm Using Sparsity Promoting Priors

The proposed Sparsity Promoting Iterated Constrained Endmember (SPICE) algorithm is based on the Iterated Constrained Endmembers algorithm by [Berman et al. \(2004\)](#) described in Section 3.2. The SPICE algorithm, which is a sparsity promoting extension of ICE, is developed in this section.

The RSS term of the ICE objective function is a least squares term whose minimization is equivalent to the maximization of Equation 3-10 ([Williams, 1995](#))

$$-\frac{1}{2} \sum_{i=1}^N \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right)^2 = \ln \exp \left\{ -\frac{1}{2} \sum_{i=1}^N \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right)^2 \right\}. \quad (3-10)$$

When examining the exponential in Equation 3-10, it can be seen that this is proportional to the Gaussian density with mean $\sum_{k=1}^M p_{ik} \mathbf{E}_k$ and variance 1,

$$\begin{aligned} \mathcal{N} \left(\sum_{k=1}^M p_{ik} \mathbf{E}_k, 1 \right) &= \frac{1}{\sqrt{2\pi}} \exp \left\{ -\frac{1}{2} \sum_{i=1}^N \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right)^2 \right\} \\ &\propto \exp \left\{ -\frac{1}{2} \sum_{i=1}^N \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right)^2 \right\}. \end{aligned} \quad (3-11)$$

Given Equation 3-4, the sparsity promoting term to be added to the ICE objective function should be of the form shown in Equation 3-12,

$$SPT = \sum_{k=1}^M \gamma_k \sum_{i=1}^N |p_{ik}| = \sum_{k=1}^M \gamma_k \sum_{i=1}^N p_{ik} \quad (3-12)$$

where the last equality follows due to the constraints in Equation 1-2. For this work, γ_k is set as shown in Equation 3-13,

$$\gamma_k = \frac{\Gamma}{\sum_{i=1}^N p_{ik}}. \quad (3-13)$$

Γ is a constant associated with the degree that the proportion values are driven to zero. The advantage of this expression for γ_k is that as the proportion values change during the minimization of the objective function, the weight associated with each endmember adjusts accordingly. If the sum of a particular endmember's proportion values becomes small, then the weight, γ_k , for that endmember becomes larger. This weight change accelerates the minimization of those proportion values. Furthermore, since the objective function is minimized in an iterative fashion, the change in the γ_k values does not disrupt the minimization.

Incorporating this sparsity promoting term into ICE's objective function yields the objective function for SPICE (Zare and Gader, 2007a),

$$RSS_{reg}^* = (1 - \mu) \frac{RSS}{N} + \mu V + SPT \quad (3-14)$$

$$= \frac{1 - \mu}{N} \sum_{i=1}^N \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right)^T \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right) + \mu V + \sum_{k=1}^M \gamma_k \sum_{i=1}^N p_{ik} \quad (3-15)$$

$$= \frac{1 - \mu}{N} \sum_{i=1}^N \left[\left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right)^T \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right) + \frac{N}{1 - \mu} \sum_{k=1}^M \gamma_k p_{ik} \right] + \mu V. \quad (3-16)$$

To minimize this new objective function, the iterative procedure in ICE can still be used. The endmembers are still found by solving Equation 3-9 since the SPT term does not depend on the endmembers. When solving for the proportion values given the

endmember estimates, each of the N terms of the sum in Equation 3-17 need to be minimized given the constraints in Equation 1-2 using quadratic programming.

$$RSS_{reg,term1}^* = \frac{1-\mu}{N} \sum_{i=1}^N \left[\left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right)^T \left(\mathbf{X}_i - \sum_{k=1}^M p_{ik} \mathbf{E}_k \right) + \sum_{k=1}^M \gamma_k^* p_{ik} \right] \quad (3-17)$$

where

$$\gamma_k^* = \frac{\Gamma^*}{\sum_{i=1}^N p_{ik}}, \quad \Gamma^* = \frac{N\Gamma}{1-\mu} \quad (3-18)$$

During the iterative minimization process, endmembers can be removed as their proportion values drop below a threshold. After every iteration of the minimization process, the maximum proportion values for every endmember can be calculated,

$$MAXP_k = \max_i \{p_{ik}\}. \quad (3-19)$$

If the maximum proportion for an endmember drops below a threshold, then the endmember can be removed from the endmember set.

3.4 New Band Selection Algorithm Using Sparsity Promoting Priors

The proposed Band Selecting Sparsity Promoting Iterated Constrained Endmembers (B-SPICE) performs band selection using sparsity promoting priors applied to band weights. This method is developed by extending the SPICE algorithm to perform simultaneous band selection. In order to perform simultaneous band selection, band weights and a band sparsity promoting term are added to the SPICE objective function in Equation 3-14 (Zare and Gader, 2008). Incorporating the band weights and the band sparsity promoting term yields Equation 3-20,

$$J = \eta \frac{RSS_B}{N} + \beta SSD_B + SPT + BST \quad (3-20)$$

where

$$RSS_B = \sum_{i=1}^N \left(\mathbf{W} \mathbf{x}_i - \sum_{k=1}^M p_{ik} \mathbf{W} \mathbf{e}_k \right)^T \left(\mathbf{W} \mathbf{x}_i - \sum_{k=1}^M p_{ik} \mathbf{W} \mathbf{e}_k \right) \quad (3-21)$$

$$SSD_B = \sum_{k=1}^{M-1} \sum_{l=k+1}^M (\mathbf{W}e_k - \mathbf{W}e_l)^T (\mathbf{W}e_k - \mathbf{W}e_l) \quad (3-22)$$

$$\mathbf{W} = \text{diag}(w_1, \dots, w_D) \quad (3-23)$$

w_i is the weight for the i^{th} band, D is the number of bands, η and β are the constant coefficient parameters for the RSS and SSD terms, and the BST term is the band sparsity promoting term defined in Equation 3-24.

The band sparsity promoting term (BST) is defined as a weighted sum of band weights with one term for each band,

$$BST = \sum_{j=1}^D \lambda_j |w_j| = \sum_{j=1}^D \lambda_j w_j \quad (3-24)$$

where

$$\lambda_j = \frac{\Lambda \left(\left(\sum_{k=1}^M \frac{1}{N} \sum_{i=1}^N p_{ik} (x_{ij} - e_{kj})^2 \right) + 1 \right)}{\left(\sum_{k=1}^M (e_{kj} - \mu_0)^2 \right) + 1} \quad (3-25)$$

Λ is a tunable parameter controlling the degree of sparsity among the band weights, μ_0 is the global data mean, x_{ij} is the j^{th} band of the i^{th} pixel, and e_{kj} is the j^{th} band of the k^{th} endmember.

The band weights are subject to the constraints in Equation 3-26,

$$w_j \geq 0, \quad j = 1 \dots d, \quad \sum_{j=1}^d w_j = d \quad (3-26)$$

where d is the number of bands. The non-negativity constraint in Equation 3-26 allows for the second equality in Equation 3-24.

The λ_j values are related to the method of ranking bands according to the Minimum Misclassification Canonical Analysis (MMCA) used by Chang et al. (1999). Chang:1999 rank bands according to the MMCA value which is derived from Fisher's discriminant function. Although the proposed method uses a weight that is related to the Fisher's discriminant value, this algorithm differs from the method used by Chang et al. (1999) by

performing simultaneous endmember detection and using sparsity promoting priors rather than a divergence threshold to determine the number of useful spectral bands.

Note that if a λ_j value is small, then the associated band weights can be large and still yield a small value in the objective function. Conversely, if a λ_j value is large, then the associated weight must be small. Hence, a large λ_j value for a particular band should lead to small weights for that band. The λ_j values are defined to depend on the ratio of the within class to between-class scatter. Each endmember has one class that consists of those points with high abundances with respect to the corresponding endmember. So, bands with small ratios separate the data and endmembers well and are therefore encouraged to have large weights. In contrast, bands with large ratios do not separate the data and endmembers well and are encouraged to be removed.

In order to minimize the new objective function in Equation 3–20, the iterative procedure used in SPICE can still be applied. The minimization process iterates between solving for the proportions, endmembers and band weights. The endmembers can be solved for directly as was done in Equation 3–9. When solving for the proportion values given endmember and band weight estimates, N quadratic programming steps, one for each data point, can be employed to minimize Equation 3–20 with respect to the constraints in Equation 1–2.

Similarly, when solving for the band weights given the proportion and endmember estimates, Equation 3–20 can be minimized using a single quadratic programming step given the constraints in Equation 3–26. After updating band weights, bands are removed from data points and endmembers when the corresponding band weight drops below a prescribed threshold.

Since the band weights and endmember values depend on each other, an optimization schedule needs to be employed. An estimate of the endmembers is needed before determining which bands are useful in distinguishing between the endmembers. Therefore, an update schedule allows the endmembers and proportions to settle before determining

band weights. The optimization schedule consists of a starting iteration for band selection, the frequency of band weight updates and a stopping criterion for band weight updates.

This iterative schedule is summarized in the following pseudo-code.

```

BSPICE(X)
1: iteration  $\leftarrow$  1
2: n  $\leftarrow$  iteration frequency of band updates
3: StartBandUpdate  $\leftarrow$  iteration to begin band weight updates
4: while (ObjValue – PreviousObjValue)2 > ChangeThreshold do
5:   Update Proportion Values
6:   Update Endmember Values
7:   BandUpdateFlag  $\leftarrow$  0
8:   if (iteration > StartBandUpdate) and (modn(iteration) = 0) and (BandUpdateFlag = 0) then
9:     Update Band Weights
10:    Remove Bands
11:    if norm(PrevBandWeight – CurrentBandWeight) < BandChangeThreshold then
12:      BandUpdateFlag  $\leftarrow$  1
13:    end if
14:  end if
15:  Update Objective Function Value, ObjValue
16:  iteration  $\leftarrow$  iteration + 1
17: end while

```

3.5 New Endmember Distribution Detection Algorithm

The new Endmember Distribution (ED) detection algorithm has the unique property of representing endmembers as random vectors, thereby calculating endmembers distributions rather than single spectra. Endmember distributions are found by assuming a model for each endmember and iteratively updating endmember distributions and proportion vectors for each pixel. ED was developed for use within the Piece-wise Convex Endmember detection algorithm in Section 3.8. However, since ED incorporates spectral variability when performing spectral unmixing and endmember determination, applications for ED may extend beyond use within the PCE algorithm.

Assuming the convex geometry model in Equation 1–1, each input hyperspectral pixel is a linear combination of the endmembers. In the following, all endmember distributions

are assumed to be Gaussian distributions with mean spectra, \mathbf{e}_k and known covariance matrices, \mathbf{V}_k . It follows that each pixel is a multivariate Gaussian random variable whose distribution is defined by the linear combination of the endmembers' Gaussian distributions,

$$P(\mathbf{x}_j|\mathbf{E}, \mathbf{p}_j) \propto \exp \left\{ -\frac{1}{2} \left(\mathbf{x}_j - \sum_{k=1}^M p_{jk} \mathbf{e}_k \right)^T \left(\sum_{k=1}^M p_{jk}^2 \mathbf{V}_k \right)^{-1} \left(\mathbf{x}_j - \sum_{k=1}^M p_{jk} \mathbf{e}_k \right) \right\} \quad (3-27)$$

where \mathbf{e}_k and \mathbf{V}_k are the mean spectrum and covariance matrix for the k^{th} endmember distribution, M is the number of endmember distributions being determined, and p_{jk} is the j^{th} data point's proportion value for the k^{th} endmember (Wackerly et al., 1996). The joint likelihood for all the hyperspectral pixels is assumed to be the product of the individual likelihoods,

$$P(\mathbf{X}|\mathbf{E}, \mathbf{P}) \propto \prod_{j=1}^N \exp \left\{ -\frac{1}{2} \left(\mathbf{x}_j - \sum_{k=1}^M p_{jk} \mathbf{e}_k \right)^T \left(\sum_{k=1}^M p_{jk}^2 \mathbf{V}_k \right)^{-1} \left(\mathbf{x}_j - \sum_{k=1}^M p_{jk} \mathbf{e}_k \right) \right\}. \quad (3-28)$$

Each hyperspectral data point has a unique abundance vector. Although all the data points share the same set of endmember distributions, their unique abundance vectors result in each data point having a unique Gaussian distribution. In Equation 3-28, the maximum likelihood value of the data point \mathbf{x}_j is $\mathbf{p}_j \mathbf{E}$.

In order to provide a tight fit around the input hyperspectral data set, the prior on the endmembers is defined using the sum of squared distances between the means of the endmember distributions. This is similar to the prior on the endmembers used by SPICE algorithm.

$$P(\mathbf{E}) = \frac{1}{(2\pi)^{\frac{D}{2}} |\mathbf{S}|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{4} \sum_{k=1}^M \sum_{l=1}^M (\mathbf{e}_k - \mathbf{e}_l)^T \mathbf{S}^{-1} (\mathbf{e}_k - \mathbf{e}_l) \right\} \quad (3-29)$$

Initially, the Dirichlet distribution was considered for the prior on the abundance values. However, since the Dirichlet distribution is not a conjugate prior to $P(\mathbf{X}|\mathbf{E}, \mathbf{P})$, a simple update formula cannot be used. Instead, constrained non-linear optimization is required when updating abundance values. As abundances approach zero (which

is very desirable and common), the log of the Dirichlet distribution is very steep and approaches $-\infty$ causing instability when using non-linear optimization techniques. Therefore, Equation 3–30 was developed for the prior on the abundance vectors.

$$P(\mathbf{p}_j) = \frac{1}{Z} \left(\sum_{k=1}^M b_k + 1 - \sum_{k=1}^M b_k (p_{jk} - c_k)^2 \right) \quad (3-30)$$

where Z is a normalization constant given by

$$Z = \frac{\sqrt{M} \left(\sum_{k=1}^M b_k + 1 \right)}{(M-1)!} - \sqrt{M} \sum_{k=1}^M \frac{b_k}{(M-1)!} \left(\left(c_k - \frac{1}{M} \right)^2 + \frac{M-1}{(M+1)M^2} \right).$$

The \mathbf{p} and \mathbf{c} vectors are constrained to be non-negative and sum-to-one,

$$\begin{aligned} p_{jk} &\geq 0 \quad \forall k = 1, \dots, M; \quad \sum_{k=1}^M p_{jk} = 1, \\ c_k &\geq 0 \quad \forall k = 1, \dots, M; \quad \sum_{k=1}^M c_k = 1. \end{aligned}$$

The vector \mathbf{c} is the maximum likelihood value for \mathbf{p} . The b_k terms control the steepness of the prior. This abundance prior prefers abundance vectors which are binary; that is, vectors with a single abundance with value 1 and the rest with value 0. This is a result of the normalization constant, Z .

The numerator of the abundance prior is maximized when \mathbf{c} is equal to \mathbf{p} . The normalization constant in the denominator is minimized when \mathbf{c} is binary. Thus, when both the \mathbf{p} and \mathbf{c} vectors are binary, the abundance prior is maximized. This property introduces sparsity within abundance vectors which, when combined with the flexibility achieved by representing endmembers by distributions, represents a major advance in automated determination of meaningful endmembers and abundances.

If several endmembers adequately describe a data point, the abundance prior will place all weight on one endmember rather than spreading the abundance across endmembers encouraging the method to use the minimum number of endmembers needed. Furthermore, many different points can be assigned abundance values of one with respect

to a given endmember because of the variance of the endmember distribution. Examples of this prior for abundance vectors of length two are shown in Figure 3-1. Also, plots showing the abundance prior as a function of \mathbf{c} are shown in Figure 3-2.

The algorithm proceeds by iteratively maximizing $P(\mathbf{X}|\mathbf{E}, \mathbf{P})P(\mathbf{E})P(\mathbf{P})$ where $P(\mathbf{P})$ is the joint likelihood of all the abundance vectors. Given initial estimates of the endmember distributions and \mathbf{c} from the abundance prior, abundance vectors are updated by maximizing the log of the product of Equations 3-28 and 3-30 with respect to \mathbf{P} . This is a constrained non-linear optimization problem. In the current Matlab implementation, this is maximized using Matlab's `fmincon` function in the optimization toolbox. Following an update of the abundance vectors, the product of Equations 3-28 and 3-29 are maximized with respect to means of the endmember distributions, \mathbf{e}_k for $k = 1, \dots, M$. This maximization is performed directly by taking the derivative of the log of the product and setting it equal to zero.

$$\mathbf{e}_k^T = \left(\sum_{j=1}^N \left(\mathbf{x}_j - \sum_{l \neq k} a_{jl} \mathbf{e}_l \right)^T \left(\sum_{l \neq k} a_{jl}^2 \mathbf{V}_l \right)^{-1} a_{jk} + \sum_{l \neq k} \mathbf{e}_l^T \mathbf{S}^{-1} \right) \left(\sum_{j=1}^N a_{jk}^2 \left(\sum_{l \neq k} a_{jl}^2 \mathbf{V}_l \right)^{-1} + (M-1) \mathbf{S}^{-1} \right)^{-1}. \quad (3-31)$$

The third step of the iteration updates the \mathbf{c} vector in the abundance prior given the abundance vectors for all the data points. The third step is also a non-linear optimization problem solved using Matlab's `fmincon` function.

Although the ED algorithm was developed for use within the PCE algorithm, applications of the ED algorithm may extend beyond this. This may occur since, using endmember distributions, the spectral variation which occurs due varying environmental conditions or inherent variability can be measured in controlled environments and then incorporated and utilized during endmember detection or spectral unmixing. For example, if endmember means and covariances are estimated from a spectral library, these can be

held constant during the ED algorithm while spectral unmixing is performed. Additional endmember distributions are learned if necessary.

The use of the endmember distribution model can represent a wide variety of data. For example, the data points in Figure 3-3 were generated using two endmember distributions. The standard model using convex combinations of single endmember spectra would require three endmembers to represent the data while maintaining a small reconstruction error.

3.6 Review of Markov Chain Monte Carlo Sampling Algorithms

The presented Piece-Wise Convex Endmember (PCE) detection algorithm uses the Dirichlet Process to sample the number convex regions needed to describe a data set. Before developing the new algorithm, a review of MCMC sampling methods is provided.

Markov Chain Monte Carlo (MCMC) sampling methods provide a means of generating samples from complicated *target distributions* without needing to enumerate every possible outcome and its probability (Chib and Greenberg, 1995; MacKay, 2003). Samples are produced in a sequence where each new sample is generated based on the previous one using a *transition kernel*. The transition kernel defines the conditional probability of moving to a particular sample (or any subset of samples) given the current sample value (Chib and Greenberg, 1995). Since each sample in the sequence is produced based on the previous one, consecutive samples generated using MCMC methods are not independent (MacKay, 2003).

One MCMC sampling method is the Metropolis-Hastings algorithm which uses a normalized *candidate-generating density* to provide potential samples (Chib and Greenberg, 1995). These candidate samples are then evaluated using an *acceptance ratio* which defines the probability of retaining or rejecting the candidate sample (Chib and Greenberg, 1995). The Metropolis-Hastings algorithm can be used when it is difficult to generate samples directly from the target distribution but samples can be easily evaluated in the distribution (MacKay, 2003).

The Metropolis-Hastings sampling method is initialized with an arbitrary starting point, s_0 . Then, a candidate, c_1 , is generated from the candidate-generating distribution, $q(s_0, \cdot)$. Given s_0 and c_1 , the acceptance ratio is computed according to Equation 3–32 (Chib and Greenberg, 1995),

$$a(s_{i-1}, c_i) = \begin{cases} \min \left[\frac{f(c_i)q(c_i, s_{i-1})}{f(s_{i-1})q(s_{i-1}, c_i)}, 1 \right] & \text{if } f(s_{i-1})q(s_{i-1}, c_i) > 0 \\ 1 & \text{otherwise} \end{cases} . \quad (3-32)$$

where q is the candidate-generating distribution which can rely on a previous sample and f is the target density from which the samples are desired (Chib and Greenberg, 1995).

The candidate sample is accepted with probability $a(s_{i-1}, c_i)$. If the sample is rejected, then $s_1 = s_0$ otherwise $s_1 = c_1$. Samples are generated in this sequential manner for a large number of iterations. In Metropolis-Hastings and all MCMC methods, samples generated during an initial period of running the algorithm are discarded. These samples generated during the *burn-in period* are discarded since convergence to the desired target distribution has not yet been reached and a bias based on the arbitrary starting point is present (Casella and George, 1992; Chib and Greenberg, 1995).

The number of samples that need to be discarded is difficult to determine. One technique to determine the length of the burn-in period is described by Chib and Greenberg (1995). This technique uses several Metropolis-Hastings generated sample sequences with varying initialization points. As samples are collected in each sequence, variances across samples are compared between the chains. The technique of using several chains is also described by Casella and George (1992) to generate independent samples from an MCMC method. Sequences with distinct starting points are generated for large number of iterations. The final samples from each chain are then used as independent and identically distributed (iid) samples from the target distribution (Casella and George, 1992). Another technique uses a single sequence and returns every k th position in the

sequence. When k is “large enough,” each retained sample can be regarded as an iid samples from the target density (Casella and George, 1992).

The popular Gibbs sampler is a significant special case of the Metropolis-Hastings algorithm (Casella and George, 1992; Chib and Greenberg, 1995). The Gibbs sampler produces samples from a multi-variate distribution by iteratively sampling from the conditional distribution of each variable given all the others (Casella and George, 1992). In this case, the candidate-generating distribution is the conditional distribution for the variable being sampled (Chib and Greenberg, 1995; MacKay, 2003). It can be shown that using the conditional distributions for producing candidate samples causes the acceptance ratio for every transition to be 1 (Chib and Greenberg, 1995).

Consider the multi-variate joint density of the random variables R , S and T , $f(r, s, t)$. The Gibbs sequence, $r_0, s_0, t_0, r_1, s_1, t_1, \dots$ is generated by iterating between the conditionals in Equation 3–33 given initial values for r_0 and s_0 (Casella and George, 1992),

$$\begin{aligned} t_j &\sim f(t|R = r_j, S = s_j) \\ r_{j+1} &\sim f(r|S = s_j, T = t_j) \\ s_{j+1} &\sim f(s|R = r_{j+1}, T = t_j). \end{aligned} \tag{3–33}$$

Furthermore, the acceptance ratio for this example can be computed and shown to be equal to 1,

$$\frac{f(r_j, s_j, t_j)q((r_j, s_j, t_j), (r_j, s_j, t_{j-1}))}{f(r_j, s_j, t_{j-1})q((r_j, s_j, t_{j-1}), (r_j, s_j, t_j))} = \frac{f(t_j|r_j, s_j)f(r_j, s_j)}{f(t_{j-1}|r_j, s_j)f(r_j, s_j)} \frac{f(t_{j-1}|r_j, s_j)}{f(t_j|r_j, s_j)} = 1. \tag{3–34}$$

3.7 Review of the Dirichlet Distribution and the Dirichlet Process

Markov Chain Monte Carlo (MCMC) sampling techniques have been applied with the Dirichlet Process to clustering problems and determining the required number of clusters (Neal, 1991; Teh et al., 2006; Xing et al., 2007). A review of the Dirichlet Process Mixture Model and its application to a data set for clustering is described in this section. First, the definitions for the Dirichlet distribution, the Dirichlet Process, and the Dirichlet Process

Mixture Model are provided. Then, a method of sampling from a Dirichlet Process Mixture Model using Gibbs sampling is developed.

Definition 1. *The Dirichlet distribution with a base distribution, $\mathbf{m} = \{m_1, m_2, \dots, m_n\}$, and a concentration parameter, α , on $\boldsymbol{\pi} = \{\pi_1, \pi_2, \dots, \pi_n\}$ is defined to be (Devroye, 1986)*

$$D(\boldsymbol{\pi}; \alpha \mathbf{m}) = \frac{\Gamma(\alpha)}{\prod_{i=1}^n \Gamma(\alpha m_i)} \prod_{i=1}^n \pi_i^{\alpha m_i - 1} \quad (3-35)$$

where $\sum_{i=1}^n \pi_i = 1$ and $\sum_{i=1}^n m_i = 1$.

The mean and covariance of the Dirichlet distribution are given by Equations 3-36, 3-37 and 3-38,

$$E[\pi_i] = \frac{\alpha m_i}{\sum_{j=1}^n \alpha m_j} = m_i \quad (3-36)$$

$$V[\pi_i] = \frac{m_i(1 - m_i)}{1 + \alpha} \quad (3-37)$$

$$C[\pi_i, \pi_j] = \frac{-m_i m_j}{1 + \alpha}. \quad (3-38)$$

By examining Equation 3-36, it can be seen that as α , the concentration parameter, is varied, the mean of the Dirichlet distribution does not change. In contrast, as α is increased, the covariance decreases.

Given the definition of a Dirichlet Distribution, the Dirichlet Process can be defined.

Definition 2. *Given a set π with a σ -algebra, \mathcal{B} , let α be a positive constant and let αG_0 be a finite, non-null, non-negative, finitely additive measure on (π, \mathcal{B}) . Then, the random probability measure G on (π, \mathcal{B}) is a Dirichlet Process on (π, \mathcal{B}) with parameters G_0 and α if for every measurable partition of the set, (B_1, B_2, \dots, B_r) , the joint distribution of $(G(B_1), G(B_2), \dots, G(B_r))$ is a Dirichlet distribution with parameters α and $(G_0(B_1), G_0(B_2), \dots, G_0(B_r))$. Therefore, $(G(B_1), G(B_2), \dots, G(B_r)) \sim \mathcal{D}(\alpha G_0)$ (Antoniak, 1974; Ferguson, 1973; Teh et al., 2004).*

3.7.1 Dirichlet Process Mixture Model

Mixture models are often used to describe data which is distributed according to some set of “underlying mechanisms” where each data point is assumed to be independently generated by only one of these underlying distributions (Neal, 1991). Finite mixture models can be expressed using Equation 3–39,

$$p(\mathbf{x}_i | \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{k=1}^M \pi_k p(\mathbf{x}_i | \boldsymbol{\theta}_k) \quad (3-39)$$

where $\boldsymbol{\pi} = \{\pi_1, \pi_2, \dots, \pi_M\}$ is the set of mixing proportions for component distributions such that $\sum_{k=1}^M \pi_k = 1$ and $\pi_k \geq 0$ and $\boldsymbol{\theta} = \{\theta_1, \theta_2, \dots, \theta_M\}$ where $\boldsymbol{\theta}_k$ is a vector of parameters for the k^{th} component distribution for $k = 1, \dots, M$.

The Dirichlet Process Mixture Model extends the basic mixture model by applying a Dirichlet Process prior to the mixing proportions. This extension allows for a countably infinite number of mixture components (Jain and Neal, 2000). Consider N data points, $\{x_1, \dots, x_N\}$ each of which are assumed to have been independently generated by some distribution $f_i(\cdot, \boldsymbol{\phi}_i)$ where $\boldsymbol{\phi}_i$ is the vector of parameters that defines the process generating observation \mathbf{x}_i . Under the Dirichlet Process Mixture Model, $\boldsymbol{\phi}_i$ is generated by some unknown distribution G (West et al., 1994). Then, G is distributed according to the Dirichlet process, $\mathcal{D}(\alpha G_0)$ where G_0 is the base distribution and α is the concentration parameter (Jain and Neal, 2000). Therefore, the complete model can be written as (Jain and Neal, 2000; Neal, 1998)

$$\begin{aligned} \mathbf{x}_i &\sim f(\cdot | \boldsymbol{\phi}_i) \\ \boldsymbol{\phi}_i &\sim G \\ G &\sim \mathcal{D}(\alpha G_0). \end{aligned} \quad (3-40)$$

Under this model, the values $\boldsymbol{\phi}_i$, $i = 1, \dots, N$, are generated from G are members of a set of $M \leq N$ distinct values denoted as $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_M\}$ corresponding to the parameters for each mixture components. More precisely, \mathbf{X} can be partitioned into M subsets, $\mathbf{X} = \mathbf{X}_1 \cup \mathbf{X}_2 \cup \dots \cup \mathbf{X}_M$ with the property that $\mathbf{x}_i \in \mathbf{X}_j$ if and only if $\boldsymbol{\phi}_i = \boldsymbol{\theta}_j$.

In other words, several data points can be generated from the same mixture component (West et al., 1994).

To simplify the model, G can be integrated out to express the prior of each ϕ_i in terms of the base distribution, G_0 , and all other parameter sets (Jain and Neal, 2000; Neal, 1998; Rasmussen, 2000; West et al., 1994),

$$\phi_i | \phi_{-i} \sim \frac{1}{\alpha + N - 1} \sum_{j=1, i \neq j}^N \delta(\phi_j) + \frac{\alpha}{\alpha + N - 1} G_0 \quad (3-41)$$

where ϕ_{-i} is the set of component distributions for all data points other than i , N is the number of data points, $\delta(\phi_i)$ is the distribution over parameters with all weight concentrated at parameter set ϕ_i , and G_0 is the prior distribution for the component parameters (Neal, 1998; Ranganathan, 2006).

3.7.2 Gibbs Sampling for the Dirichlet Process Mixture Model

As shown by Neal (1998), the likelihood of a data point given component parameters can be combined with the probability of a class label given all other labels in Equation 3-41. Then, the Gibbs sampler can be used to sample indicator variable values and component parameter values. The conditional probabilities for an indicator variable are defined in Equation 3-42,

$$\begin{aligned} P(c_i = c_j \text{ for some } j \neq i | \mathbf{c}_{-i}, \mathbf{x}_i, \theta) &= C \frac{n_{-i,j}}{\alpha + N - 1} f(\mathbf{x}_i | \theta_{c_j}) \\ P(c_i \neq c_j \forall j \neq i | \mathbf{c}_{-i}, \mathbf{x}_i) &= C \frac{\alpha}{\alpha + N - 1} \int f(\mathbf{x}_i | \theta) G_0(\theta) d\theta \end{aligned} \quad (3-42)$$

where C is a normalizing constant computed by Equation 3-43,

$$\frac{1}{\sum_j \left(\frac{n_{-i,j}}{\alpha + N - 1} f(\mathbf{x}_i | \theta_{c_j}) \right) + \frac{\alpha}{\alpha + N - 1} \int f(\mathbf{x}_i | \theta) G_0(\theta) d\theta}. \quad (3-43)$$

The Markov Chain for the Gibbs sampler using the conditionals in Equation 3-42 consists of all the indicator variables c and all component distribution parameters θ (Neal, 1998).

If G_0 is a conjugate prior to the likelihood distributions (component distributions) $f(\cdot | \theta)$, then the integral in Equation 3-42 can be analytically computed (Neal, 1998). Assuming

that the posterior can be integrated, when G_0 is a conjugate prior to the likelihood, integrating over the component parameters in both conditionals in Equation 3–42 requires that only the indicator variables of the observations need to be sampled. In this case, the conditional distributions are expressed as in Equation 3–44 (Jain and Neal, 2000),

$$\begin{aligned} P(c_i = c_j \text{ for some } j \neq i | \mathbf{c}_{-i}, \mathbf{x}_i) &= C \frac{n_{-i,j}}{\alpha + N - 1} \int f(\mathbf{x}_i | \boldsymbol{\theta}) H_{-i,c_j}(\boldsymbol{\theta}) d\boldsymbol{\theta} \\ P(c_i \neq c_j \forall j \neq i | \mathbf{c}_{-i}, \mathbf{x}_i) &= C \frac{\alpha}{\alpha + N - 1} \int f(\mathbf{x}_i | \boldsymbol{\theta}) G_0(\boldsymbol{\theta}) d\boldsymbol{\theta} \end{aligned} \quad (3-44)$$

where C is a normalizing constant and H_{-i,c_j} is the posterior distribution of the component parameters given prior G_0 and current indicator values \mathbf{c}_{-i} (Jain and Neal, 2000). These integrals remove the need to include component parameters in the Markov Chain which significantly reduces the search space for the Gibbs sampler (Jain and Neal, 2000).

For cases in which G_0 is not a conjugate prior to the likelihood functions, techniques have been developed to either estimate the integral values or use sampling techniques to avoid the need to compute the integral values. Some of these techniques are discussed by Neal (1998). One method uses the Metropolis-Hastings algorithm where the candidate distribution for a parameter set, $\boldsymbol{\theta}_i$, is $\frac{1}{N-1+\alpha} \sum_{j \neq i} \delta(\boldsymbol{\theta}_j) + \frac{\alpha}{N-1+\alpha} G_0$ and the acceptance probability is $a(\boldsymbol{\theta}_i^*, \boldsymbol{\theta}_i) = \min \left[1, \frac{f(\mathbf{x}_i, \boldsymbol{\theta}_i^*)}{f(\mathbf{x}_i, \boldsymbol{\theta}_i)} \right]$. Another method avoids the need to evaluate the integral by introducing temporary auxiliary variables into a Gibbs sampling scheme. In this method, the temporary auxiliary variables are parameter sets drawn independently from G_0 (Neal, 1998).

Specific cases of this Gibbs algorithm for the Dirichlet Process Mixture Model are derived in the literature. Rasmussen (2000) derives the algorithm where the component distributions and priors are all Gaussian. Neal (1991) derives the method for categorical data using a Bernoulli distributions for the component distributions and a Beta distributions for the priors.

As described by [Rasmussen \(2000\)](#) and [West et al. \(1994\)](#), this model can also be extended by adding hyper-priors for the α parameter and the parameters of the prior distribution G_0 .

3.8 New Piece-Wise Convex Endmember Detection Algorithm using the Dirichlet Process

In this section, a novel method for endmember detection using the Dirichlet process is presented. Existing endmember detection algorithms generally assume that all pixels in a hyperspectral image are convex combinations of a single set of endmembers. However, some hyperspectral images may be better represented using several sets of endmembers. The new algorithm partitions the input hyperspectral data set into convex regions each with its own set of endmember distributions. Using the Dirichlet process, the Piece-wise Convex Endmember (PCE) detection algorithm learns the number of convex regions needed to represent an input hyperspectral image while simultaneously learning endmember distributions and proportion values for each partition.

This method differs from the Dirichlet process mixture model since each convex region is represented with a set of endmember distributions for which each data point has a unique abundance vector. Thus, as previously shown in Equation 3-27, each data point is a random variable with a unique distribution. Each data point having a unique distribution contrasts with the DPMM approach where data points from each cluster are assumed to be identically distributed.

The PCE algorithm performs Gibbs sampling with Dirichlet process priors to sample the partition to which each data point belongs. The probability of sampling a partition is computed using the likelihood of a data point belonging to a convex combination of the

associated endmember distributions,

$$\begin{aligned}
P(r_i = r_j \quad j \neq i | \mathbf{r}_{-i}, \mathbf{x}_i) &= C \frac{n_{-i,j}}{\alpha + N - 1} \int f(\mathbf{x}_i | \mathbf{p}_i^{r_j}, \mathbf{V}^{r_j}, \mathbf{E}^{r_j}) H_{-i,r_j}(\mathbf{E}^{r_j}, \mathbf{V}^{r_j}, \mathbf{P}^{r_j}) d\mathbf{p}_i^{r_j} \mathbf{E}^{r_j} \\
&= \mathcal{N} \left(T(T + S)^{-1} \mathbf{p} \mathbf{E}^{r_j} + S(T + S)^{-1} \mathbf{c} \mathbf{E}^{r_j}, S + T(T + S)^{-1} S \right) \\
P(r_i \neq r_j \quad \forall j \neq i | \mathbf{r}_{-i}, \mathbf{x}_i) &= C \frac{\alpha}{\alpha + N - 1} \int f(\mathbf{x}_i | \mathbf{E}^*) G_0(\mathbf{E}^*) d\mathbf{E}^* \\
&= \mathcal{N} \left(\mathbf{V}_0(\mathbf{V}_0 + \mathbf{V})^{-1} \mathbf{x}_i + \mathbf{V}(\mathbf{V}_0 + \mathbf{V})^{-1} \mu_0, (\mathbf{V}_0^{-1} + \mathbf{V}^{-1})^{-1} + \mathbf{V} \right)
\end{aligned} \tag{3-45}$$

where r_i is the indicator variable for the current data point, \mathbf{x}_i , C is a normalization constant, $n_{-i,j}$ is the number of data points excluding \mathbf{x}_i in partition r_j , N is the total number of data points, and α is the innovation parameter for the Dirichlet process. The matrices, T and S , correspond to $\sum_k c_k^2 V_k$ and $\sum_k p_{ik}^2 V_k$, respectively. The matrices \mathbf{V} and \mathbf{V}^{r_j} are all the covariance matrices associated with new and existing endmember distributions. In the current implementation of this algorithm, all covariance matrices for endmember distributions are set to the same constant matrix value.

The prior distribution, G_0 , is Gaussian where the mean, μ_0 , is set to the mean of the input data set and the covariance, \mathbf{V}_0 , is constant,

$$G_0 = \mathcal{N} \left(\mu_0 = \frac{1}{N} \sum_{j=1}^N \mathbf{x}_j, \mathbf{V}_0 \right) \tag{3-46}$$

The prior distribution combined with α , the innovation parameter in the Dirichlet process prior, dictates the probability of generating a new partition. The covariance matrix, \mathbf{V}_0 , is set to a large value to approximate a broad uniform prior over the data set.

Assuming that each endmember distribution is Gaussian with a known covariance matrix, the likelihood for an existing partition, $f(\mathbf{x}_i | \mathbf{p}_i^{r_j}, \mathbf{V}^{r_j}, \mathbf{E}^{r_j})$, is determined by Equation 3-27. The vector, $\mathbf{p}_i^{r_j}$, contains the proportion values for the current data point in partition r_j . These proportion values are determined by maximizing the product of Equations 3-30 and 3-27 given the endmembers of the partition, \mathbf{E}^{r_j} . The likelihood

value, f , measures the ability of a set of endmembers to represent a data point by computing the distance between the data point and \mathbf{pE} .

The distribution, $H_{-i,r}(\mathbf{E}^r, \mathbf{V}^r, \mathbf{P}^r)$ is the prior distribution updated based on the data points assigned to the r^{th} partition,

$$H_{-i,r}(\mathbf{E}^r, \mathbf{P}^r) = \mathcal{N} \left(\mathbf{cE}^r, \sum_{k=1}^M c_k^2 \mathbf{V}_k^r \right) \quad (3-47)$$

where \mathbf{c} is the center from abundance prior determined by maximizing Equation 3-30 given \mathbf{P}^r and \mathbf{V}^r . By incorporating this updated prior, the likelihood depends not only on the distance to \mathbf{pE} but also to \mathbf{cE} . When the covariance matrices for endmember distributions are equal, the updated prior depends on the distance to a point on the line segment connecting \mathbf{pE} and \mathbf{cE} , namely, $w_1 \mathbf{pE} + w_2 \mathbf{cE}$ where $w_1 = \frac{\sum_{k=1}^M c_k^2}{\sum_{k=1}^M (c_k^2 + p_{jk}^2)}$ and $w_2 = \frac{\sum_{k=1}^M p_{jk}^2}{\sum_{k=1}^M (c_k^2 + p_{jk}^2)}$.

As stated above and shown in line 12 of the following pseudo-code, in each iteration of the algorithm a partition is sampled for the current data point. In this step, a partition is sampled by computing the likelihood of a data point belonging to each existing partition and the likelihood of a data point generating a new partition using Equation 3-45. The unit interval is then divided into regions whose lengths are equal to each partition's normalized likelihood value. A random value from the unit interval is then generated. The corresponding partition whose region includes the generated random value is the partition which is sampled for the current data point.

After a partition is sampled, the parameters of the sampled partition are updated. This is done by updating the prior on the abundances, Equation 3-30, with respect to \mathbf{c} for the given partition. After one or more iterations of the partition sampling scheme using the Dirichlet process, the endmember distributions and all proportion values are updated using a designated number of iterations of the ED algorithm.

Several items in the following PCE pseudo-code differ from the standard DPMM method. As stated in lines 10 and 13 of the pseudo-code, in PCE, a partition's parameters

are updated when a data point is removed or added to the partition by updating the partition's \mathbf{c} vector in the abundance prior. In contrast, for the standard Gaussian DPMM method, the mean of the Gaussian cluster would be updated instead. Lines 16 to 18 of the pseudo-code also differ from the standard DPMM method. After a set number of Gibbs sampling iterations in PCE, each partition's endmembers and proportion matrices are updated. In the standard DPMM, all values associated with each cluster are updated in each Gibbs iteration. PCE essentially performs a series of several Gibbs sampling runs each with a new set of endmembers.

PCE(X)

```

1: Initialize Partitions
2: for  $r \leftarrow 1$  to  $R_{initial}$  partitions do
3:   Initialize  $\mathbf{E}^r$  and  $\mathbf{P}^r$  using ED
4: end for
5: for  $k \leftarrow 1$  to number of total iterations do
6:   for  $i \leftarrow 1$  to number of Gibbs sampling iterations do
7:     Randomly reorder data points in  $X$ 
8:     for  $j \leftarrow 1$  to number of data points do
9:       Remove  $\mathbf{x}_j$  from its current partition
10:      Update the partition's  $\mathbf{c}$ 
11:      Compute Dirichlet process partition probabilities for  $\mathbf{x}_j$  using Equation 3–45.
12:      Sample a partition for  $\mathbf{x}_j$  based on the Dirichlet process partition probabilities
13:      Update new partition's  $\mathbf{c}$ 
14:    end for
15:   end for
16:   for  $r \leftarrow 1$  to  $R_k$  partitions do
17:     Update  $\mathbf{E}^r$  and  $\mathbf{P}^r$  using ED
18:   end for
19: end for
20:  $R_{final} = R_k$ 

```

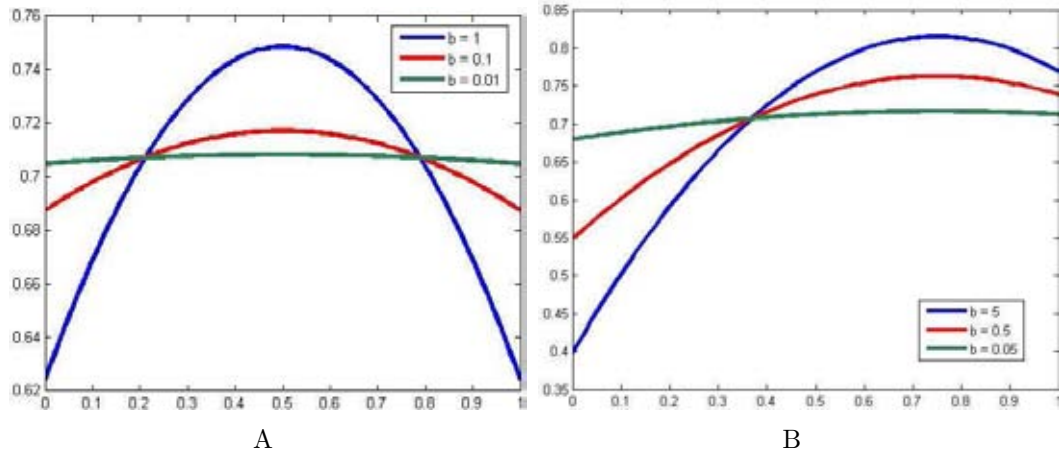


Figure 3-1. Plots of ED's abundance prior for $M = 2$ and various \mathbf{c} and \mathbf{b} values. The x -axis is the 1st abundance value. The y -axis is the prior probability value for the abundance vector. A) $\mathbf{c} = [.5, .5]$ B) $\mathbf{c} = [.75, .25]$.

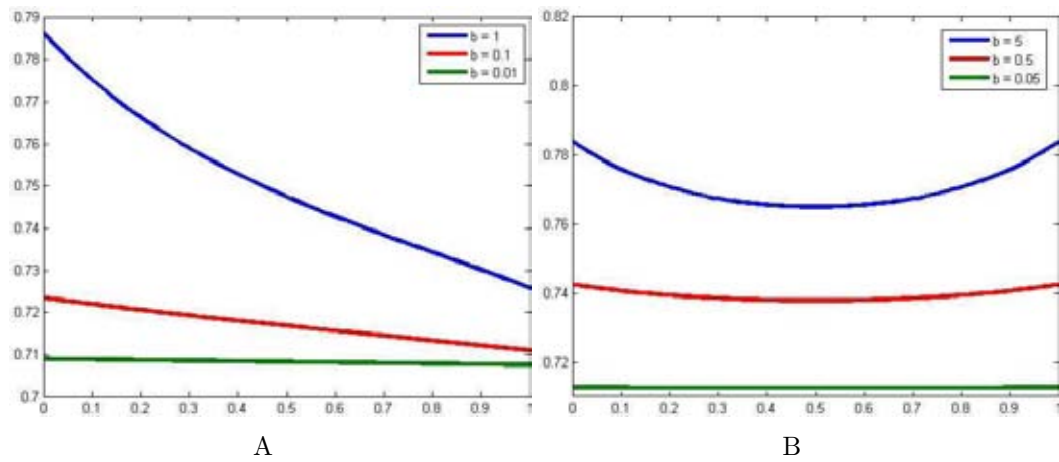


Figure 3-2. Plots of ED's abundance prior for $M = 2$ and various \mathbf{p} and \mathbf{b} values. The x -axis is the 1st \mathbf{c} value. The y -axis is the prior probability value for \mathbf{c} . A) $\mathbf{p} = [.45, .55]$ B) $\mathbf{p} = [.5, .5]$.

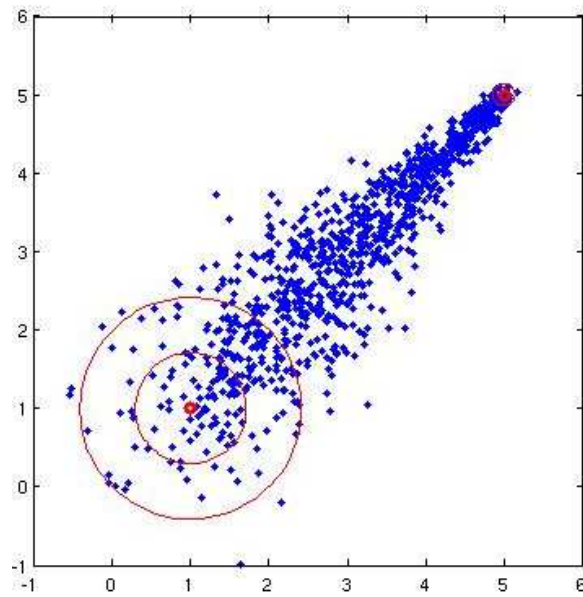


Figure 3-3. Data points generated from linear combinations of 2 endmember distributions. The endmember distribution centered at (5,5) has a diagonal covariance whose elements are all equal to 0.005. The endmember distribution centered at (1,1) has a diagonal covariance whose elements are all equal to 0.5. Data points are shown in blue. Mean spectra and standard deviation curves for the endmember distributions are shown in red.

CHAPTER 4 RESULTS

The presented algorithms were applied to a variety of data sets including two-dimensional data, simulated data, and hyperspectral imagery collected using AVIRIS, the Airborne Visible/Infrared Image Spectrometer, and AHI, the Airborne Hyperspectral Imager. Both the AVIRIS Cuprite and Indian Pines data sets were used.

4.1 Sparsity Promoting Iterated Constrained Endmember (SPICE) Algorithm Results

The SPICE algorithm has been applied to a variety of simulated and real hyperspectral data sets. SPICE results are shown and discussed in Sections 4.1.1 to 4.1.4.

4.1.1 The SPICE Two-Dimensional Example Results

A two-dimensional example was initially used for testing the SPICE algorithm. Figure 4-1 shows the data set and the endmembers from which the data were generated. These data points were generated in the same fashion as the two-dimensional example shown by Berman et al. (2004). The endmembers that were used to generate the 100 data points were $(-10\sqrt{2}, 0)$, $(10\sqrt{2}, 0)$ and $(0, 20)$. The maximum proportions of the bottom two endmembers were 0.80 and 0.60, respectively. Zero-mean independent Gaussian random noise with a variance of 1 was added to each coordinate of the data points.

When running ICE, Berman et al. (2004) assign the number of endmembers for this example to three. In SPICE, the number of endmembers does not need to be known in advance. Therefore, the algorithm can be initialized with a large number of endmembers. The results of three experiments comparing the ICE and SPICE algorithms are shown in Figure 4-2. The parameters for each algorithm, other than the sparsity-promoting term, were set to be the same during the experiments. The initial number of endmembers for all three runs was 20, and μ was set to 0.001. The γ parameter for the SPICE algorithm was set to 10, 20, and 5 for the three runs, respectively. ICE and SPICE were initialized to the same endmembers for each experiment. These initial endmembers were chosen randomly from the data set.

An endmember was pruned from either algorithm when the endmember’s maximum proportion over the data points dropped below 0.0005. In these three experiments, proportion values were averaged over the iterations in which an endmember was pruned.

$$MINMAX P_k = \min_k \left\{ \max_i \{p_{ik}\} \right\} \quad (4-1)$$

These were found to be 3.3×10^{-4} , 2.4×10^{-4} , and 2.4×10^{-4} for ICE, respectively. In comparison, 4.1×10^{-6} , 8.3×10^{-17} , and 7.8×10^{-17} are these mean values for SPICE in the three experiments, respectively. As shown, these values are significantly lower in SPICE compared to the pruning threshold than the values in ICE. SPICE consistently drives proportion values for unnecessary endmembers well below a 0.0005 pruning threshold. Despite this high pruning threshold, ICE did not find the correct number of endmembers with pruning without the use of a sparsity-promoting term.

As shown by the results, SPICE consistently determined that three endmembers was an appropriate number of endmembers to represent the data set. ICE ended the algorithm with six endmembers. In Figure 4-2D, two of the endmembers that are found by ICE were (-3.62, 7.94) and (-3.68, 7.94); they appear as one endmember in the figure.

SPICE was also applied to the two-dimensional data used to test the Morphological Associative Memory method in Section 2.1.4. The SPICE results on this data set are shown in Figure 4-3. For the results shown, Γ was set to 20 and μ was set to 0.01.

4.1.2 The SPICE AVIRIS Cuprite Data Results

SPICE was also tested on real hyperspectral imagery. The AVIRIS hyperspectral image data from Cuprite, NV was used. This data contained 51 contiguous spectral bands (in the range of 1978 - 2477 nm) from “Scene 4” of the AVIRIS Cuprite data set (AVIRIS). This data set was chosen to be able to compare the results with the NFINDR results presented by Winter (1999).

Following Berman et al. (2004), SPICE was run on a subset of pixels from the image to reduce computational time. These *candidate points* were selected using the

pixel purity index (PPI) method (Berman et al., 2004; Boardman et al., 1995). In these experiments, the candidate points were chosen from 10,000 random projections. Points within a distance of two from the boundary of the projection received increased purity indices. The 1011 pixels with the highest PPI were used as the candidate points. 1000 pixels were used by Berman et al. (2004) during the experiments on the real image sets. A PPI threshold that produced as close to 1000 pixels as possible (many pixels had the same PPI) was chosen. Also, fast implementations for the algorithm can be created as was done by Berman et al. (2004) to avoid the need to select a subset of the pixels.

The spectral profiles of the nine endmembers that were found by SPICE to represent this image are shown in Figure 4-4. The three endmembers in the Figures 4-4C, 4-4G, and 4-4I compare well to three endmembers that are found and identified as kaolinite, alunite, and calcite by Winter (1999), respectively. Figure 4-5 shows a comparison of 4-4I to the U.S. Geological Survey (USGS) spectral library data on alunite (Clark et al., 2004).

Although it is clear that SPICE was able to find some of the same endmembers that are identified by Winter (1999), it is not clear if the correct number of endmembers was found. The difficulty of using real image data is that the correct number of endmembers in the scene is unknown. To overcome this problem, a subset of the Cuprite data was used for further testing of the method.

Three endmembers, shown in Figure 4-6, were selected from the hyperspectral image by hand. The squared Euclidean distance was calculated from every pixel in the image to these three endmembers. The pixels within 500,000 squared Euclidean distance from these three hand-selected endmembers were collected and used as a test set for SPICE. The test set was normalized and is shown in Figure 4-7.

Table 4-1 shows the number of endmembers that are found using SPICE for a range of Γ 's and initial number of endmembers. As shown, SPICE consistently finds three endmembers for this data set. The results in Table 4-1 and in Figure 4-2 show that the SPICE algorithm is fairly stable with respect to Γ . SPICE is also very stable with respect

to the initial number of endmembers. Therefore, the initial number of endmembers should be set to a large value.

Figure 4-8 shows the endmembers that are found using SPICE in these experiments. These endmembers are clearly very similar to the three hand-selected endmembers used for this experiment.+

4.1.3 The SPICE AVIRIS Indian Pines Results

SPICE was also run on the June 1992 AVIRIS data set collected over the Indian Pines Test site in an agricultural area of northern Indiana. The image has 145×145 pixels with 220 spectral bands and contains approximately two-thirds agricultural land and one-third forest and other elements. The soybean and corn crops in the image are in early growth stages and, thus, have only about a 5% crop cover (Grana and Gallego, 2003; Serpico and Bruzzone, 2001). The remaining field area is soil covered with residue from the previous crop. The no till, min till, and clean till labels indicate the amount of previous crop residue remaining. No till corresponds to a large amount of residue, min till has a moderate amount, and clean till has a minimal amount of residue (Serpico and Bruzzone, 2001). Figure 4-9 shows band 10 (approximately $0.49 \mu\text{m}$) and the ground truth of the data set. Only 49% of the pixels in the image have ground truth information (Serpico and Bruzzone, 2001).

SPICE was run on a subset of the image pixels. 1100 pixels were randomly selected from the image. Before running SPICE, these pixels were normalized. The initial number of endmembers, μ , and Γ were set to 60, 0.1, and 1, respectively. Ten endmembers were found for this data set using SPICE. The resulting abundance maps are shown in Figure 4-10. SPICE pruned unnecessary endmembers and provided interpretable results that compare well to previously published results on this data set (Grana and Gallego, 2003; Grana et al., 2003; Miao et al., 2006).

In Figure 4-10, the images were found to roughly correspond to the following: (A) and (I) are woods and tree canopies; (B), (C), and (J) are a mixture of soybean and corn

crops; (D) and (E) are grass and background; (F) is hay windrows; (G) is steel towers, roads, and other man-made objects; and (H) is grass/pasture and wheat.

Since 49% of the pixels in the scene are unlabeled, SPICE was also tested on only the labeled pixels of the AVIRIS Indian Pines scene. A total of 1037 normalized pixels (every 10th labeled pixel) was selected from the image and used to determine the endmembers. The initial number of endmembers, μ , and Γ , were set to 20, .01 and .1, respectively. Six endmembers were found for the labeled pixels of the Indian Pines scene. The abundance maps are shown in Figure 4-11. The endmembers roughly correspond to the following classes: (A) grass/pasture and woods, (B) hay-windrowed, alfalfa and grass/pasture-mowed, (C) and (E) correspond to corn and soybean, (D) stone-steel towers, and (F) grass/trees, wheat, woods.

Normalized histograms showing the distribution of abundances values among endmembers in each ground truth class are shown in Figure 4-12. These histograms were computed by summing all the abundance values associated with an endmember in each ground truth class. Each histogram was normalized by dividing by the number of points in the corresponding ground truth class,

$$h_{lk} = \frac{\sum_{i: \mathbf{x}_i \in G_k} a_{il}}{N_k} \quad (4-2)$$

where G_k is the set of pixels in ground truth class k , N_k is the number of points in ground truth class k , a_{il} is the i^{th} data points' abundance value for the l^{th} endmember, and h_{lk} is the k^{th} histogram's value corresponding to the l^{th} endmember.

4.1.4 The SPICE AHI Vegetation Detection Results

The SPICE algorithm was run on data collected by AHI, the Airborne Hyperspectral Imager (Lucey et al., 1998; Zare and Gader, 2007b). AHI collects 256 spectral bands of data from the long wave infrared region in the range of 7.88 to 11.49 microns (Lucey et al., 1998). The AHI data set collected is trimmed and binned down to 70 bands over the same wavelengths (Lucey et al., 1998). The data set used for these results was collected from

an arid testing site containing both surface and buried landmines. Fiducial markers are also contained in the imagery for alignment and ground truthing purposes. SPICE was applied to this data to extract vegetation endmembers and create a vegetation mask for the reduction of false alarms during landmine detection (Zare and Gader, 2007b).

Three AHI images were used and scoring was performed to determine the reduction in false alarm rate. Scoring for the results in this paper was carried out over regions of interest in the imagery. The regions of interest for this study were defined as the areas where collected Lynx Synthetic Aperture Radar and AHI imagery intersect (LYNXSAR). Four mine types were distributed in the intersecting regions. Two of the mine types were plastic cased (PC) and two were metal cased (MC). The distribution of mines types in the intersecting regions of the AHI and Lynx images are displayed in Table 4-2 (Zare and Gader, 2007b).

Vegetation mapping in the long wave infrared. Vegetation detection in the LWIR is based on the emissive properties of vegetation. Vegetation behaves similar to a blackbody in the LWIR exhibiting a high mean emissivity and a low standard deviation of emissivity across spectral bands (French et al., 2000). Additionally, skewness of emissivity across spectral bands has been seen to be helpful in distinguishing vegetation (Zare et al., 2008). To exploit this information, the SPICE algorithm can be run on the emissivity spectra calculated from LWIR hyperspectral data. For this study, the emissivity spectrum of each pixel in the image is calculated using the Emissivity Normalization Method (Kealy and Hook, 2000).

After applying the SPICE algorithm to the emissivity spectra, the endmembers determined by the algorithm are examined. The endmember with the highest mean and the lowest standard deviation is determined to be the blackbody endmember,

$$\mathbf{E}_B = \begin{cases} \mathbf{E}_k & \text{if } \arg \max_{\mathbf{E}_k}(\mu_k) = \arg \min_{\mathbf{E}_k}(\sigma_k) \\ \emptyset & \text{otherwise} \end{cases} \quad (4-3)$$

where μ_k is the mean and σ_k is the standard deviation across the spectral bands of the k^{th} endmember, \mathbf{E}_k , found by SPICE.

Since the proportion maps generated by the SPICE algorithm represent the amount of a particular endmember in a pixel, the proportion map associated with the blackbody endmember is used as the blackbody map, ν , for the image,

$$\nu_j = \begin{cases} 0 & \text{if } \mathbf{E}_B = \emptyset \\ p_{jB} & \text{otherwise} \end{cases} \quad (4-4)$$

where j corresponds to the j^{th} pixel in the image. A mask \mathbf{V} , is defined using the blackbody map by inverting the values and enhancing the map using a local 3 minimum filter,

$$V_j = localmin(1 - \nu_j) \quad (4-5)$$

where j corresponds to the j^{th} pixel in the image. Following the local 3×3 minimum filter, a partial threshold was applied to the mask,

$$V_j^{thresh} = \begin{cases} V_j & \text{if } V_j \leq t \\ 1 & \text{otherwise} \end{cases} \quad (4-6)$$

where t is the threshold determined using Otsu's thresholding method (Otsu, 1979). The partial threshold is applied so that the only values modified by the mask are those that are associated with pixels that behave like a blackbody (Zare and Gader, 2007b).

Following Berman et al. (2004), SPICE was run on a subset of pixels from the image. The subset was selected using the Pixel Purity Index (PPI) algorithm (Boardman et al., 1995). The subset was chosen using 30,000 random projections. Points within a distance of three from the boundary of the projection received increased pixel purity values. A threshold was selected to allow as close to 1000 pixels as possible (many pixels have the same PPI). The number of points selected was 1095, 767 and 1103 for AHI Images 1, 2,

and 3, respectively. In order to compute proportion maps for the entire image, the entire image was unmixed using the endmembers found on the image subsets.

The results were compared to those generated using the vegetation mapping method described by Zare et al. (2008). Since Zare et al. (2008) used only the statistics of emissivity (mean, standard deviation and skewness across spectral bands) instead of the full emissivity curve, the results displayed are those generated by running SPICE on only the statistics of emissivity instead of the full emissivity spectra. This method was used to be able to compare performance of the clustering method by Zare et al. (2008) and the SPICE method directly without adding confusion over whether the difference in performance resulted from the methods or the input data. Furthermore, a partial threshold as defined in Equation 4–6 was also applied to the mask generated by the clustering method by Zare et al. (2008).

In contrast to SPICE, which finds the desired number of endmembers for a data set, the method by Zare et al. (2008) requires the number of clusters to be supplied to the algorithm. The method was run on this data with the number of clusters ranging from three to six. The results displayed below are the best results obtained over this range of number of cluster values. Figure 4-14 shows the blackbody mask generated using SPICE and the clustering method for four and five clusters. When comparing the two masks generated by the clustering method, it can be seen that when five clusters is chosen instead of four, many of the vegetation pixels are being split between multiple clusters and, thus, are farther from the selected vegetation cluster center. Since SPICE automatically selects the desired number of endmembers, this difficulty is eliminated. When examining the SPICE mask and the clustering mask generated with 4 clusters, it can be seen that the SPICE mask provides more solid vegetation regions thus providing a better mapping of the vegetation pixels than the clustering method.

Points of Interest, POIs, in the overlap regions of the imagery were found using the RX detector algorithm (Yu et al., 1993). The RX detector applied was an implementation

of the well-known anomaly detection algorithm by [Winter \(2004\)](#). The RX algorithm was applied to detect buried mines in the LWIR hyperspectral imagery. The blackbody mask is incorporated by multiplying the RX confidence of every POI with their corresponding blackbody mask value. This differs from the detection algorithms used by [Zare et al. \(2008\)](#) where the blackbody mask is applied to the output of a Choquet fusion system incorporating several detection algorithms. In these results, only the comparative performance of the two blackbody masks are being examined.

The results in each of the three overlap regions are shown in Tables [4-3](#), [4-4](#) and [4-5](#). The *probability of detection*, PD, is defined as the number of mines with a confidence above the threshold divided by the total number of mines. The *false alarm rate*, FAR, is defined as the number of non-mines above the threshold divided by the number of square meters in the overlap region. Although RX was applied to detect buried mines, the results are shown over all mine types in the overlap regions. If detected, fiducial markers in the scene are considered false alarms.

The first line in each table displays the false alarm rates without any blackbody mask being used on the RX values. The second line displays the FARs after applying the blackbody mask generated using the clustering method. The third line shows the reduction in the FAR after using the blackbody mask from the clustering method. The fourth line displays the FAR after applying the blackbody mask generated using SPICE. Finally, the fifth line shows the reduction in FAR after using the blackbody mask from SPICE when compared to the results without using a blackbody mask ([Zare and Gader, 2007b](#)).

The blackbody mask generated using the SPICE algorithm can provide false alarm reduction during landmine detection. In comparison to the clustering method by [Zare et al. \(2008\)](#), the SPICE method provides improved vegetation detection and eliminates the need to set the number of clusters or endmembers needed.

4.2 Band Selecting SPICE (B-SPICE) Algorithm Results

Results on hyperspectral imagery using the B-SPICE algorithm are shown with comparisons to other band selection methods in Sections 4.2.1 to 4.2.3.

4.2.1 The B-SPICE AVIRIS Cuprite Data Results

The B-SPICE algorithm was applied to a simulated data set generated using four normalized endmembers selected from the AVIRIS Cuprite data set ([AVIRIS](#)). The chosen endmembers are shown in Figure 4-15. The data set was generated from the endmembers following the convex geometry model in Equation 1-1.

A simulated data set was used to verify that the method can recover the endmembers, perform effective band selection, and produce accurate abundance values for each pixel. These can be tested using simulated data since the true endmembers and abundances are known. B-SPICE and SPICE were run on this data set for a range of Λ values. All parameters, other than those involved with band selection, were held constant for each run of the algorithm. The η parameter was set to 2000, β was 0.3, Γ was 0.3 for the first 150 iterations and then set to 0, the initial number of endmembers was set to 20, and the endmember pruning threshold was 1×10^{-8} . The initial endmembers were selected randomly from the data set. When running B-SPICE, band selection was not started until the 100th iteration, after which, the band weights were updated every fifth iteration. The band pruning threshold was set to 1×10^{-5} , and the band weight change threshold was set to 1×10^{-5} . Λ was set to 0 (for SPICE), 0.25, 0.5, 0.75, and 1 (for B-SPICE). B-SPICE and SPICE were run on the data 50 times for each parameter set. An example of the endmembers found using each Λ value is shown in Figure 4-16.

Table 4-6 shows the mean and standard deviation of the number of endmembers and the number of bands retained for each parameter set over the 50 runs of the algorithm. As can be seen, both SPICE and B-SPICE are able to consistently determine the correct number of endmembers.

To evaluate the effectiveness of the band selection performed, the average squared error per abundance value was calculated. The mean, median and standard deviation of the error values are shown in Table 4-7. The median average squared error per abundance value was computed by taking the median over 50 runs of the algorithm of the average squared error between each pixels true and computed abundance values. As shown, the median average squared error per abundance value is fairly stable across the Λ values, indicating that B-SPICE can be as effective at determining the true abundance values as the SPICE algorithm. Therefore, by using B-SPICE, the number of bands can be reduced while maintaining the ability to determine abundances. However, when examining the standard deviation of the average squared error per abundance value, it is seen that SPICE is more consistent than B-SPICE. There is an order of magnitude difference between the standard deviations of SPICE and B-SPICE.

4.2.2 The B-SPICE AVIRIS Indian Pines Results

The B-SPICE algorithm was also run on the June 1992 AVIRIS Indian Pines data set described in Section 4.1.3. SPICE and B-SPICE were run twice for five different Λ values. All parameters, other than the Λ parameter, were held constant for each run of the algorithm. To reduce run time, SPICE and B-SPICE were run on 1000 pixels randomly chosen from the data set. After determining the endmembers and selected bands using the subset, unmixing was performed on the entire data set to find abundance values for every pixel. The η parameters was set to 5000, β to 0.3, and Γ to 0.2 for the first 100 iterations and then to 0. The initial number of endmembers was set to 20 and the endmember pruning threshold was 1×10^{-8} . Initial endmembers were selected randomly from the data set. When running B-SPICE, band selection was not started until the 100th iteration, after which, the band weights were updated every fifth iteration. The band pruning threshold was set to 1×10^{-5} , and the band weight change threshold was set to 1×10^{-5} . Λ was set to 0 (for SPICE), 0.5, 1, 5, and 10 (for B-SPICE). The number of endmembers and the number of bands found are shown in Table 4-8.

In order to compare these results to those presented by [Guo et al. \(2006\)](#), supervised classification was performed. The features used for supervised classification were the abundance values computed for each pixel in the 16 classes of the data set. The unlabeled pixels were not included in these experiments. Since the abundance values were the features used for classification, the dimensionality of the feature vectors is equal to the number of endmembers found for the data set.

Two-fold cross-validation was performed on the data set using a 1-versus-1 Relevance Vector Machine (RVM) classification method ([Tipping, 2001](#)). The training and testing sets were defined by randomly splitting each of the 16 classes in half. An RVM was trained for each pair of classes. Since there are 16 classes, 120 RVMs were trained for each test set. Test pixels were classified by counting the number of RVMs that assigned the pixel to each class.

$$L_p = [v_1^p, \dots, v_{16}^p] \quad (4-7)$$

where v_i^p is the number of times the pixel p was assigned to class i by the trained RVMs. After every pixel was run through the entire set of trained RVMs, spatial smoothing was performed to assign a label to each pixel. Spatial smoothing was done by summing over the neighborhood of pixel p and assigning the class with the largest number of votes

$$C_p = \arg \max_{i \in \{1, \dots, 16\}} \left\{ \sum_{j \in N_p} v_i^j \right\} \quad (4-8)$$

where C_p is the label for pixel p and N_p is a set of pixels in the eight-connected neighborhood of pixel p . The overall classification accuracies for each run of the B-SPICE algorithm are shown in Table 4-8. Since the classification accuracies depend on the random splitting of the data into training and testing sets, classification was performed three times for each run of the B-SPICE algorithm.

[Wang et al. \(2006\)](#) provide supervised classification results with band selection on the Indian Pines data set. The results shown by [Wang et al. \(2006\)](#) show very good classification accuracies ranging from 90% to 94.5% with less than 50 bands; however,

their classification method was not described. Band selection results on the Indian Pines data set are also shown by Archibald and Fann (2007) and Huang and He (2005), but the results are provided on only a subset of the labeled classes. Table 4-8 also shows results from Martinez-Uso et al. (2006); only the results with less than 50 bands were provided.

4.2.3 The B-SPICE AVIRIS Indian Pines Results using Sampled Parameter Values

In order to reduce the need to set parameters by hand, parameters can be sampled from prior distributions. This was implemented by sampling η , β , Γ , and Λ from gamma distributions with means of 6000, 0.3, 0.2, and 1.240 parameter value sets that were sampled, respectively. B-SPICE was run on the Indian Pines data set using each of the 240 sets of sampled parameters. The results of 240 runs can be combined to determine the number of endmembers, the number of bands, and the bands to retain for the data set. Figure 4-17 shows the histograms of the number of endmembers, the number of bands, and the number of times each band was retained. Modes of the histograms in Figure 4-17 are 7 and 114, respectively. The most frequently retained bands over the 240 runs were 1-57, 61-76, 81-100, and 118-138. By using these modes and the most frequently retained bands, ICE can be run to find endmembers and abundance values. In other words, the number of endmembers and the bands to retain were determined using the histograms found by running B-SPICE over sampled parameter values. These values were then used to set the number of endmembers and the bands to use for the ICE algorithm.

The classification accuracies using the sampled parameter values were determined using the same classification method done in the previous Indian Pines experiment in Section 4.2.2. Table 4-9 shows two runs of the ICE algorithm and with three runs of 1-versus-1 classification.

4.3 Endmember Distribution (ED) Detection Results

Endmember Distribution detection results are shown on two-dimensional and hyperspectral imagery. Comparisons between results found using SPICE and ED are discussed in Section 4.3.1

4.3.1 Results on Two-Dimensional Data using ED

The ED algorithm was initially tested on the two-dimensional data shown in Figure 4-1. The results found on this data set using the ED algorithm with the parameters values listed in Table 4-10 are shown in Figure 4-18. After running the algorithm, the final \mathbf{c} vector found for the abundance prior was $[.47 \ .27 \ .26]$ where the values correspond to the endmember distributions centered at $(-11.9, 1.9)$, $(-0.1, 18.5)$ and $(7.5, 6.9)$, respectively. As can be seen ED performed as expected; the endmember distributions surround the data points and compare well to the endmember results found by SPICE in Figure 4-2.

ED was also run on the two-dimensional data shown in Figure 4-19. This data was generated by sampling endmembers from three endmember distributions and computing the data points as convex combinations of the sampled endmembers using randomly generated abundance values. The ED results on this data are shown in Figure 4-20. Parameters used to generate these results are shown in Table 4-10. Again, ED generated the expected results. The endmember distributions that were found are very similar to those used to generate the data.

For comparison, SPICE was also run on the two-dimensional data in Figure 4-19. The SPICE was run on the data set with $\mu = 0.01$, $\gamma = 1$ and 10 initial endmembers. The resulting endmembers are shown in Figure 4-21A. As shown in the figure, SPICE needed four endmembers to represent the data set.

SPICE results found using $\mu = 0.001$, $\gamma = 1$ and 10 initial endmembers are shown in Figure 4-21B. By decreasing μ , less emphasis is placed on the sum-of-squared distances term in the SPICE objective function which may result in SPICE requiring a smaller

number of endmembers. However, in this case, SPICE still required four endmembers to represent the data set.

The SPICE μ parameter was reduced rather than adjusting γ because, by adjusting μ , the residual error incurred by representing data points using endmembers is kept low. A smaller number of endmembers can be found by increasing γ , however, the residual error will increase.

4.3.2 Results on AVIRIS Cuprite data using ED

To examine ED's capabilities on hyperspectral imagery, ED was run on the simulated AVIRIS Cuprite data generated from the endmembers shown in Figure 4-15. The data set was generated based on the convex geometry model in Equation 1-1. The results using ED on this data set are shown in Figure 4-22.

Parameter values used to generate these results are shown in 4-11. As can be seen in Figure 4-22, the means of the endmember distributions match the true endmembers well.

ED was also run on the subset of AVIRIS Cuprite data shown in Figure 4-7. This data set is a compilation of the pixels spectrally similar to three endmembers selected from the AVIRIS Cuprite data. The endmembers are shown in Figure 4-6. Results on this data set found using the ED algorithm are shown in Figure 4-23. As can be seen in the figure, the means of the endmember distributions closely match the true endmembers and the data set. These results superimposed on the input data set are shown in Figure 4-24.

4.4 Piece-wise Convex Endmember (PCE) Detection Results

The PCE algorithm was tested on two-dimensional data and the AVIRIS Indian Pines hyperspectral data set. Results are presented and compared to SPICE algorithm results.

4.4.1 Detection Results on Two-Dimensional Data using PCE

The PCE algorithm was initially tested on two-dimensional data. The data set used is shown in Figure 4-25. This data set was generated from three sets of endmembers, each set forming a triangle of data points.

Results on this data, after running PCE for 100 iterations, are shown in Figure 4-26. Prior to running the algorithm, partitions were initialized using the Kernel Global Fuzzy C-Means (KG-FCM) algorithm and the Dirichlet Process Mixture Model algorithm resulting in 8 partitions (Heo and Gader, 2008). Endmembers and abundances for each partition were then initialized by running the ED algorithm on each cluster. The parameters used to generate these results are shown in Table 4-12. As can be seen, PCE partitioned the data set into the correct number of convex regions. Furthermore, PCE was able to identify an appropriate set of endmembers for each convex region.

4.4.2 Detection Results on the AVIRIS Indian Pines Data using PCE

The PCE algorithm was further tested on the labeled pixels of the AVIRIS Indian Pines hyperspectral data set described in Section 4.1.3. Prior to running the PCE algorithm, the data dimensionality was reduced from 220 bands to 6 dimensions using principal components analysis. A total of 1037 pixels (every 10th labeled pixel) were selected from the data set and used in the PCE algorithm. Partitions on this data were initialized using the KG-FCM algorithm and the DPMM algorithm resulting in 3 partitions. After initial partitions were found, endmembers for each partition were initialized using the ED algorithm. Each partition was restricted to 3 endmembers. The parameters used to generate results shown are listed in Table 4-12.

In order to compute abundance maps for the entire image, every data point was unmixed using each partitions' set of endmembers and the likelihood under each partition was computed. Every data point was then assigned to partition with the largest likelihood value. Also, all endmembers whose maximum proportion value was less than 0.05 were removed. Following these steps, 13 clusters were found with a total of 14 endmembers. Figure 4-27 shows the abundances maps associated with each endmember.

For comparison with the SPICE results in Figure 4-12, normalized histograms showing the distribution of abundance values across each endmember were computed using Equation 4-2. The histograms found are shown in Figure 4-28. When comparing

the SPICE and PCE histograms, the PCE results for each ground truth class are more concentrated than the SPICE results. This fact can be measured by computing Shannon's entropy for the normalized histogram associated with each ground truth class (Bishop, 2006). A smaller entropy value indicates that a fewer number of endmembers are being used to describe each ground truth class and that the endmembers are better representatives of the ground truth classes. The sum of the Shannon entropies for the SPICE histograms comes to 19.0. In contrast, the sum of the Shannon entropies for the PCE histograms is significantly lower at 9.4. This indicates that PCE produces endmembers which better represent the ground truth classes.

The histograms and abundance maps associated with several of the ground truth classes verify that PCE is producing endmembers which provide a better representation of the data than the endmembers produced by SPICE. Some of these ground truth classes are wheat, stone-steel towers, hay-windrowed.

Consider the wheat ground truth class in the SPICE and PCE results. The SPICE abundance map associated with the most amount of wheat is shown in Figure 4-11F and the corresponding histogram is found in Figure 4-12M. By examining the abundance map, it can be seen that many pixels other than wheat have non-zero abundance values associated with wheat's SPICE endmember. In contrast, very few pixels outside of the wheat ground truth class share wheat's endmember. This is shown in the PCE abundance map in Figure 4-27J. Furthermore, by examining the SPICE histogram for wheat, only about 60% of the wheat pixels' abundance values are associated with that endmember whereas 100% of wheat's abundance values are placed with the associated endmember found using PCE.

For the stone-steel towers ground truth class, more than 70% of the pixels assigned to a single endmember using PCE and that endmember is not associated with any other ground truth classes. The SPICE endmember most associated with the stone-steel towers ground truth class is also used by every other ground truth class.

The hay-windrowed (Figure 4-28H), grass/pasture-mowed (Figure 4-28G) and alfalfa (Figure 4-28A) PCE histograms show that they are associated with the same endmember. This can also be seen in the abundance map in Figure 4-27I. The corresponding SPICE histograms for hay-windrowed, grass/pasture-mowed and alfalfa in Figures 4-12H, 4-12G, and 4-12A show that the three ground-truth classes have similar histogram shapes and share the same endmembers. However, the abundances found by SPICE are spread among three endmembers whereas PCE placed their full weight with one endmember.

Soybean and corn ground truth classes constitute a large majority of the Indian Pines scene. In the SPICE results, abundance values associated with the soybean and corn classes are spread over all of the six endmember found. In contrast, the PCE endmember results places nearly all soybean and corn abundances with the 2nd, 6th, and 10th endmembers.

Another indication that PCE is producing representative endmembers is found with the Building/Grass/Trees/Drive ground truth class. This class is composed of a variety of material types. Interestingly, this is clearly shown in the class' PCE histogram (Figure 4-28O). The abundance values for the class are spread across many endmembers.

In order to verify that the difference in the results between PCE and SPICE are not due to different data dimensionality and a different number of endmembers, the ICE algorithm was run on the same AVIRIS PCA-reduced Indian Pines data set discussed in this section. The ICE algorithm was employed rather than SPICE since the number of endmembers can be set to the same number found by PCE. The ICE algorithm was restricted to 14 endmembers and the μ parameter was set to 0.01. The resulting abundance maps are shown in Figure 4-29 and the corresponding histograms are in Figure 4-30.

The sum of the ICE histogram entropies from these results is 29.2. In comparison, PCE's value was 9.4. Therefore, although ICE was restricted to the same number of endmembers found using PCE, ICE did not produce endmembers that represent the

ground truth classes as well as PCE. The comparison between the SPICE histograms in Figures 4-30I, 4-30J, 4-30K, and 4-30L and the PCE histograms in Figures 4-28I, 4-28J, 4-28K, and 4-28L are indicative of this. These histograms correspond to the oats and soybean classes. In the SPICE histograms, the abundance values are spread across all of the endmembers. In contrast, the PCE histograms for these ground truth classes concentrate the abundance values to a few endmembers. The PCE results in this section strongly indicate that the algorithm produces endmembers which correspond very well to the true ground truth classes.

The PCE results on AVIRIS Indian Pines data with hierarchical dimension reduction. The PCE algorithm was run again on the AVIRIS Indian Pines data set. However, rather than reducing dimensionality using PCA, hierarchical dimensionality reduction was used (Martinez-Uso et al., 2006). The data dimensionality was reduced from 220 to 3 dimensions. The hierarchical dimensionality reduction computed the pair-wise KL-divergences between the bands' normalized histograms of intensity values. The KL-divergences were then used to hierarchically group similar bands. The average value across each group of bands was used to form the reduced dimensionality data set.

Partitions were initialized using the KG-FCM algorithm and the DPMM algorithm resulting in 3 clusters. Initial endmembers were found for each partition using the ED algorithm. A total of 1037 pixels (every 10th labeled pixel) were selected from the data set and used in the PCE algorithm. Parameter values used to generate the results shown on this data set are listed in Table 4-12.

In order to compute abundance maps for the entire image, after finding endmembers on the subset of pixels using PCE, every data point was unmixed using each clusters' set of endmembers and the likelihood under each cluster is computed. Every data point was assigned to cluster with the largest likelihood value. Furthermore, partitions with less than 5 assigned pixels were pruned. Following these steps, 2 clusters were found with a total of 6 endmembers. Figure 4-31 shows the abundances maps associated with each endmember.

Abundances maps from this experiment are shown in Figure 4-31 and histograms are shown in Figure 4-32. The first partition found using PCE on the three-dimensionality data corresponded to the majority of the corn and soybean ground truth classes. Hay and alfalfa were also associated with the first partition. The second partition included the majority of the grass, trees and woods. The sum of the Shannon entropy values over the histograms from the PCE results came to 16.3 compared to SPICE's 19.0 value. Again, PCE provided more compact histograms and SPICE indicating that the endmembers are better representatives of the true ground truth classes.

The PCE results on full spectra AVIRIS Indian Pines data. The PCE algorithm was run again on the AVIRIS Indian Pines data set. In this run, the data dimensionality was not reduced; the full 220 bands were used. Partitions were initialized using the KG-FCM algorithm and the DPMM algorithm to 3 clusters. Initial endmembers were found for each partition using the ED algorithm. A total of 1037 pixels (every 10th labeled pixel) were selected from the data set and used in the PCE algorithm. Parameter values used to generate the results shown on this data set are listed in Table 4-12.

After finding endmembers on the subset of pixels using PCE, every data point in the image was unmixed using each clusters' set of endmembers and, for every data point, the likelihood under each cluster was computed. Each data point was then assigned to the partition with the maximum likelihood value. Partitions with less than 3 points were removed. Following these steps, two partitions were found with a total of six endmembers. Figure 4-33 shows the abundances maps associated with each endmember. Figure 4-34 contains the normalized histograms for this set of results.

The first partition roughly corresponds to the various tilled fields in the imagery whereas the second partition has many of the abundances associated with trees, grass and woods. The sum of the entropies of the histograms from this results came to 15.4. This is value smaller than the SPICE results of 19.0 indicating that the endmembers are better representatives of the ground truth classes than the endmembers found by SPICE.

Table 4-1. Number of endmembers found by SPICE and ICE on test pixels from AVIRIS Cuprite data over a range of Γ values and initial number of endmembers. Each experiment had the same initialization for ICE and SPICE. The μ parameter was 0.1 for all experiments. The pruning threshold was set to 1×10^{-9} .

| Experiment | Initial number of endmembers | Gamma constant for SPICE | Number of endmembers found, SPICE | Number of endmembers found, ICE |
|------------|---------------------------------|-----------------------------|---|---------------------------------------|
| 1 | 5 | 1.0 | 3 | 5 |
| 2 | 10 | 0.5 | 3 | 9 |
| 3 | 10 | 0.5 | 3 | 8 |
| 4 | 10 | 10.0 | 3 | 9 |
| 5 | 10 | 10.0 | 3 | 8 |
| 6 | 15 | 1.0 | 3 | 12 |
| 7 | 30 | 1.0 | 3 | 12 |
| 8 | 40 | 1.0 | 3 | 13 |
| 9 | 50 | 1.0 | 3 | 11 |

Table 4-2. Mine distributions in overlap regions of AHI and Lynx imagery

| Mine type | Depth | AHI image 1 quantity | AHI image 2 quantity | AHI image 3 quantity |
|--------------|---------|-------------------------|-------------------------|-------------------------|
| PC1 | 10 cm | 44 | 17 | 17 |
| MC1 | 10 cm | 57 | 48 | 26 |
| MC1 | Flush | 34 | 34 | 20 |
| MC1 | Surface | 16 | 16 | 16 |
| MC2 | Surface | 14 | 14 | 0 |
| PC2 | Surface | 5 | 0 | 0 |
| Total | | 170 | 129 | 79 |

Table 4-3. False alarm rate reduction using blackbody mask in AHI image 1

| | Probability of detection | | | | |
|---------------------------|--------------------------|----------------------|----------------------|----------------------|----------------------|
| | 20% | 30% | 40% | 50% | 60% |
| RX without BB mask | 2.3×10^{-3} | 3.3×10^{-3} | 5.8×10^{-3} | 6.7×10^{-3} | 9.0×10^{-3} |
| Clustering BB mask | 2.1×10^{-3} | 3.0×10^{-3} | 5.3×10^{-3} | 6.1×10^{-3} | 8.3×10^{-3} |
| FAR reduction | 8.7% | 9.1% | 8.6% | 9.0% | 7.78% |
| SPICE BB mask | 1.0×10^{-3} | 1.2×10^{-3} | 2.3×10^{-3} | 2.8×10^{-3} | 4.2×10^{-3} |
| FAR reduction | 56.5% | 63.6% | 60.3% | 58.2% | 53.3% |

Table 4-4. False alarm rate reduction using blackbody mask in AHI image 2

| | Probability of detection | | | | |
|---------------------------|--------------------------|----------------------|----------------------|----------------------|----------------------|
| | 20% | 30% | 40% | 50% | 60% |
| RX without BB mask | 1.7×10^{-3} | 2.6×10^{-3} | 3.8×10^{-3} | 6.1×10^{-3} | 8.5×10^{-3} |
| Clustering BB mask | 1.4×10^{-3} | 2.2×10^{-3} | 2.8×10^{-3} | 4.2×10^{-3} | 6.1×10^{-3} |
| FAR reduction | 17.6% | 15.4% | 26.3% | 31.1% | 28.2% |
| SPICE BB mask | 1.2×10^{-3} | 1.9×10^{-3} | 2.5×10^{-3} | 3.8×10^{-3} | 5.8×10^{-3} |
| FAR reduction | 29.4% | 26.9% | 34.2% | 37.7% | 31.8% |

Table 4-5. False alarm rate reduction using blackbody mask in AHI image 3

| | Probability of detection | | | | |
|---------------------------|--------------------------|----------------------|----------------------|----------------------|----------------------|
| | 20% | 30% | 40% | 50% | 60% |
| RX without BB mask | 3.7×10^{-3} | 5.2×10^{-3} | 9.2×10^{-3} | 1.2×10^{-3} | 1.6×10^{-3} |
| Clustering BB mask | 3.7×10^{-3} | 5.2×10^{-3} | 9.2×10^{-3} | 1.2×10^{-3} | 1.6×10^{-3} |
| FAR reduction | 0% | 0% | 0% | 0% | 0% |
| SPICE BB mask | 3.3×10^{-3} | 4.4×10^{-3} | 6.8×10^{-3} | 1.0×10^{-2} | 1.4×10^{-2} |
| FAR reduction | 10.8% | 15.4% | 26.1% | 16.7% | 12.5% |

Table 4-6. Mean number and standard deviation of endmembers and bands found over 50 runs of SPICE or B-SPICE on the simulated data set. The true number of endmembers for this data set is 4.

| Λ | Mean number of endmembers | Standard deviation of number of endmembers | Mean number of bands retained | Standard deviation of number of bands retained |
|-----------|---------------------------------|---|--|---|
| 0 (SPICE) | 4 | 0 | 51.0 | 0.0 |
| 0.25 | 4 | 0 | 34.6 | 1.1 |
| 0.50 | 4 | 0 | 25.0 | 1.4 |
| 0.75 | 4 | 0 | 20.9 | 1.2 |
| 1.00 | 4 | 0 | 16.2 | 3.7 |

Table 4-7. Statistics of the averaged squared error per abundance value between calculated and true abundance values

| Λ | Median average squared error per abundance | Mean average squared error per abundance | Standard deviation of average squared error per abundance |
|-----------|--|--|--|
| 0 (SPICE) | 0.005 | 0.005 | 0.0005 |
| 0.25 | 0.004 | 0.008 | 0.0066 |
| 0.50 | 0.004 | 0.007 | 0.0057 |
| 0.75 | 0.004 | 0.007 | 0.0050 |
| 1.00 | 0.006 | 0.010 | 0.0069 |

Table 4-8. Indian Pines Data Set Results and Comparison. Comparison Values Estimated from Graphs in (Guo et al., 2006) and (Martinez-Uso et al., 2006)

| Exp. | Λ | Num. of endmembers | Num. of bands | Classification accuracy | | | Comparison | |
|------|-----------|-----------------------|------------------|-------------------------|-------|-------|-----------------------|------------------------|
| | | | | in percentage | | | results in percentage | |
| | | | | Run 1 | Run 2 | Run 3 | Ref. Guo, et al. | Ref. M.-Uso, et al. |
| 1 | 0.0 | 8 | 220 | 93.6 | 93.9 | 93.9 | 90 | - |
| 2 | 0.0 | 7 | 220 | 93.1 | 93.1 | 92.9 | 90 | - |
| 3 | 0.5 | 7 | 124 | 93.3 | 93.7 | 93.7 | 90 | - |
| 4 | 0.5 | 7 | 122 | 93.0 | 92.9 | 93.2 | 90 | - |
| 5 | 1.0 | 7 | 89 | 93.4 | 93.3 | 93.6 | 90 | - |
| 6 | 1.0 | 7 | 103 | 93.3 | 93.3 | 93.5 | 90 | - |
| 7 | 5.0 | 7 | 34 | 86.4 | 86.4 | 86.3 | 88 | 80 |
| 8 | 5.0 | 8 | 34 | 86.5 | 86.0 | 86.4 | 88 | 80 |
| 9 | 10.0 | 7 | 19 | 83.4 | 83.9 | 82.5 | 82 | 81 |
| 10 | 10.0 | 8 | 18 | 77.8 | 80.0 | 78.3 | 82 | 82 |

Table 4-9. Indian Pines Data Set results using sampled parameter values and comparison with (Guo et al., 2006)

| Experiment number | Number of endmembers | Number of bands kept | Classification accuracy | | | Comparison results | |
|----------------------|-------------------------|-------------------------|-------------------------|-------|-------|--------------------|--|
| | | | in percentage | | | in percentage | |
| | | | Run 1 | Run 2 | Run 3 | Ref. Guo, et al. | |
| 1 | 7 | 114 | 92.1 | 92.1 | 92.2 | 90 | |
| 2 | 7 | 114 | 92.6 | 92.4 | 92.5 | 90 | |

Table 4-10. Parameter values used to generate ED results on two-dimensional data sets. All covariance matrices are diagonal with elements equal to the values shown in the table.

| Data Set | Variance of data | Likelihood variance | SSD variance | b_k |
|------------------------------|---------------------|------------------------|-----------------|-------|
| Triangle Data (Fig. 4-1) | 55.9 | 0.1 | 0.5 | 0.01 |
| Data from Dists. (Fig. 4-19) | 5.2 | 0.5 | 1.0 | 0.01 |

Table 4-11. Parameter values used to generate ED results on hyperspectral data sets. All covariance matrices are diagonal with elements equal to the values shown in the table.

| Data Set | Dimensionality of data | Variance of data | Likelihood variance | SSD variance | b_k |
|-------------------|---------------------------|---------------------|------------------------|-----------------|-------|
| Cuprite Data | 51 | 0.01 | 0.200 | 0.20 | 0.001 |
| Indian Pines Data | 220 | 0.02 | 0.001 | 0.01 | 0.010 |

Table 4-12. Parameter values used to generate PCE results. All covariance matrices are diagonal with elements equal to the values shown in the table.

| Data Set | Data dimen. | Variance of data | Likelihood variance | Likelihood α | ED likelihood variance | ED SSD variance | b_k |
|-----------------|-------------|------------------|---------------------|---------------------|------------------------|-----------------|-------|
| 2D Data | 2 | 2.16 | 0.005 | 2 | 0.010 | 1.000 | 0.001 |
| PCA IP | 6 | 0.05 | 0.005 | 1 | 0.005 | 0.001 | 0.001 |
| Hierarchical IP | 3 | 0.03 | 0.001 | 1 | 0.010 | 0.010 | 0.010 |
| Full Spectra IP | 220 | 0.03 | 0.001 | 1 | 0.010 | 0.010 | 0.010 |

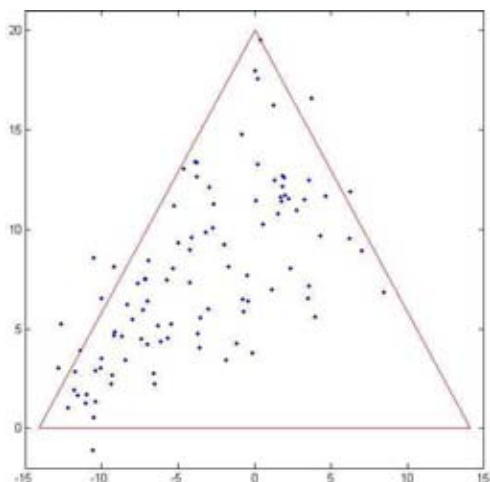


Figure 4-1. Two-dimensional SPICE example data set. 100 data points generated from the corners of the simplex shown.

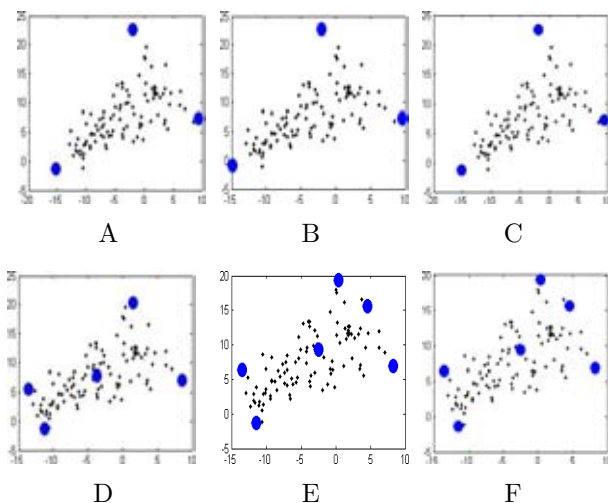


Figure 4-2. Comparison of SPICE (top) and ICE with pruning (bottom). In these three experiments, $\mu = 0.001$ and the pruning threshold was set to 0.0005. Initial number of endmembers was 20.

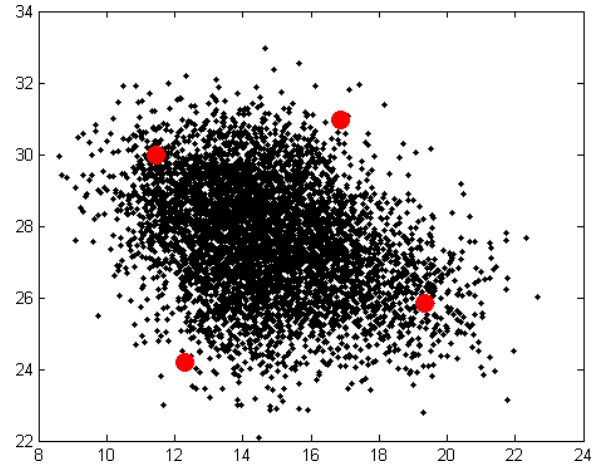


Figure 4-3. The SPICE results on two-dimensional data

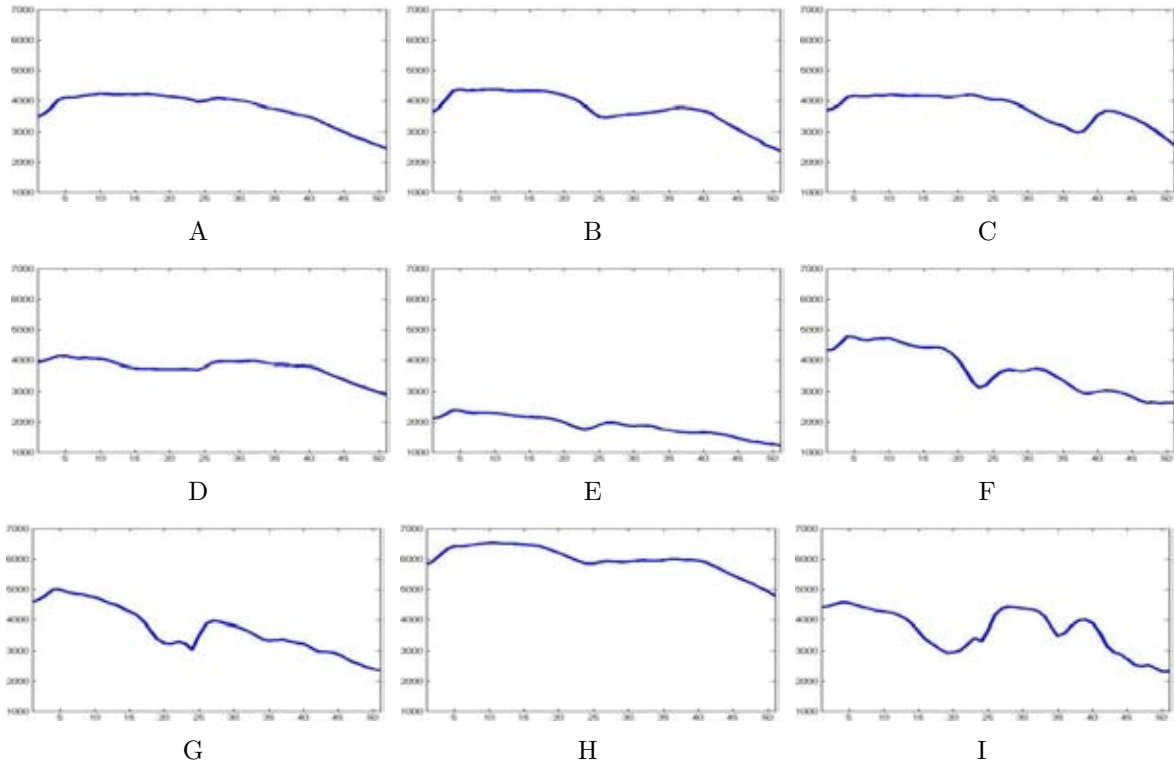


Figure 4-4. Endmembers found using SPICE on AVIRIS Cuprite hyperspectral data. μ was 0.1 for all experiments. The pruning threshold was set to 1×10^{-9} . The limits of the x -axis are 1978 to 2477 nm and the limits of the y -axis are 1000 to 7000 in units of 10,000 times the reflectance factor (Clark et al., 2004).

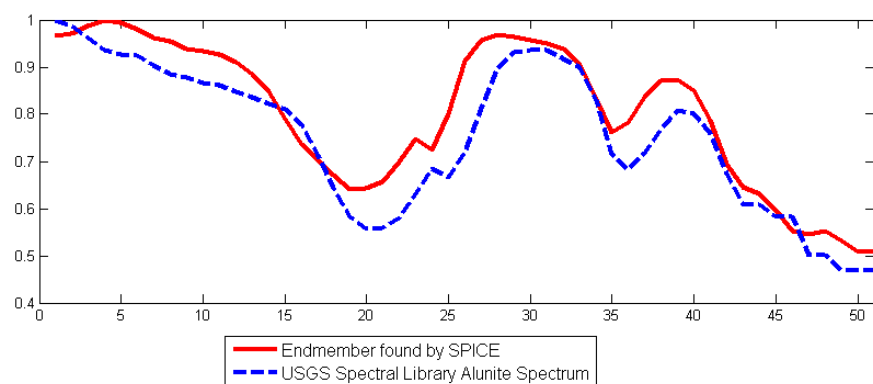


Figure 4-5. Comparison of one endmember found by SPICE and a USGS Alunite spectrum (“Alunite SUSTDA-20 W1R1Ba AREF”) from the 2005 USGS spectral library.

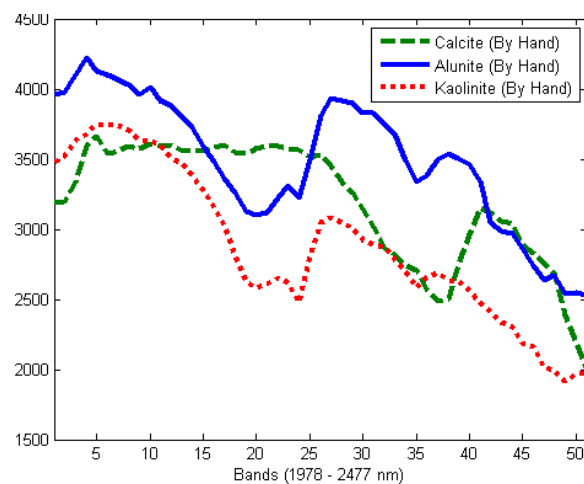


Figure 4-6. Endmembers selected from AVIRIS Cuprite data image by hand.

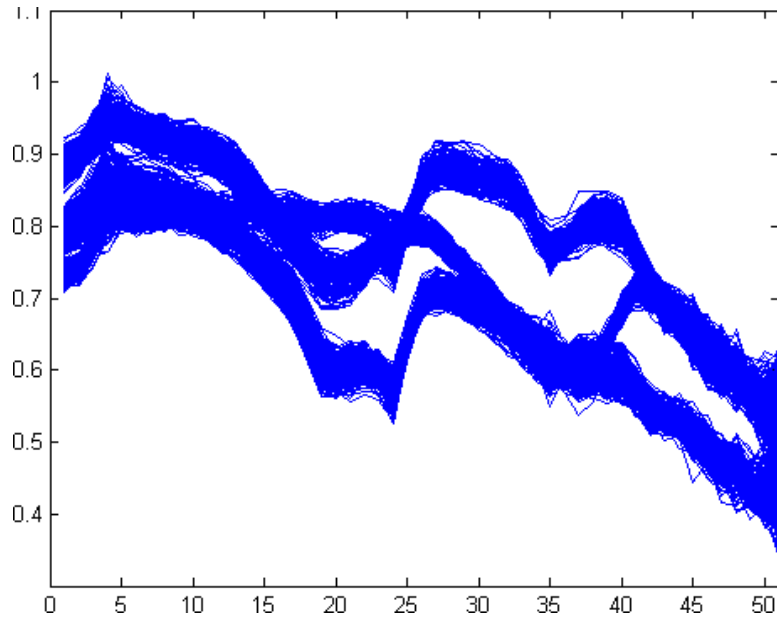


Figure 4-7. Normalized test pixels selected from AVIRIS Cuprite data.

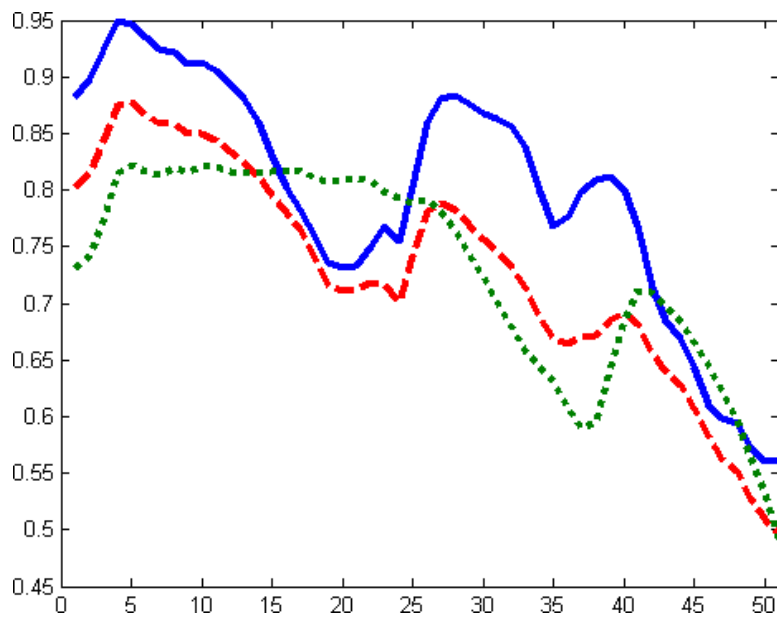


Figure 4-8. SPICE endmember results found on normalized test data selected from the AVIRIS Cuprite scene

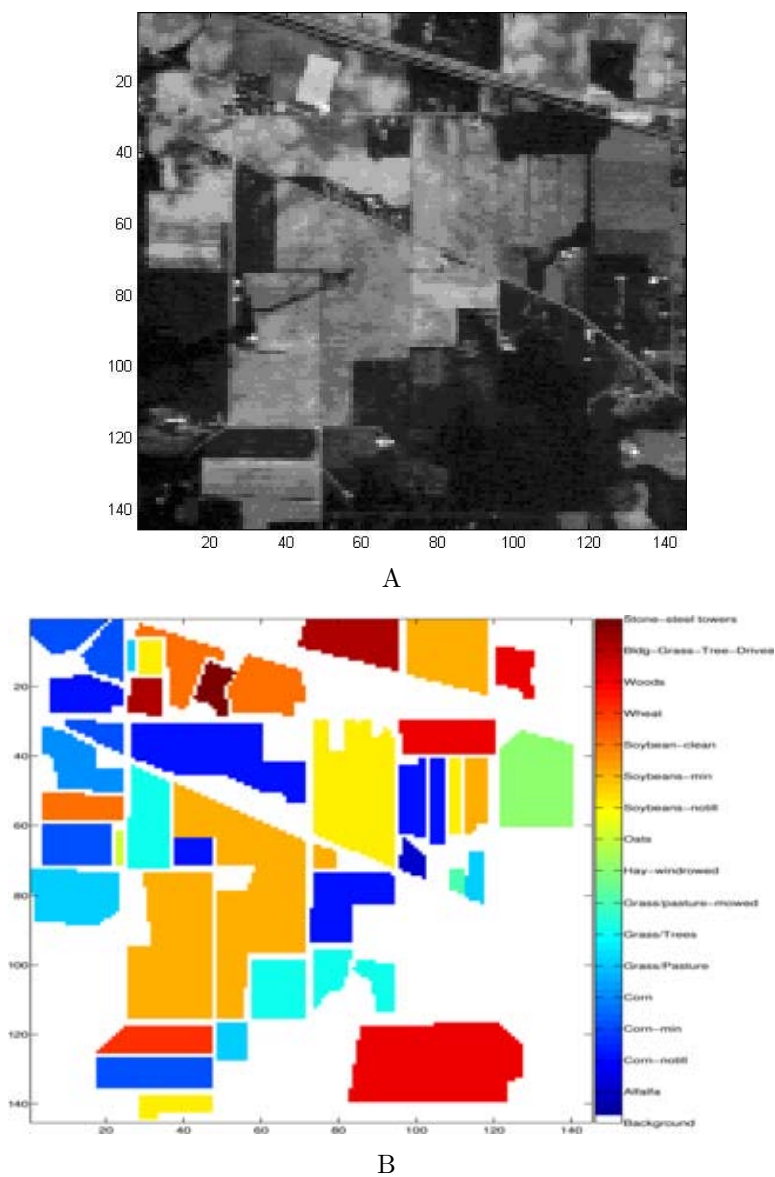


Figure 4-9. Band 10 ($\sim 0.5 \mu\text{m}$) of the AVIRIS Indian Pines data set and ground truth.

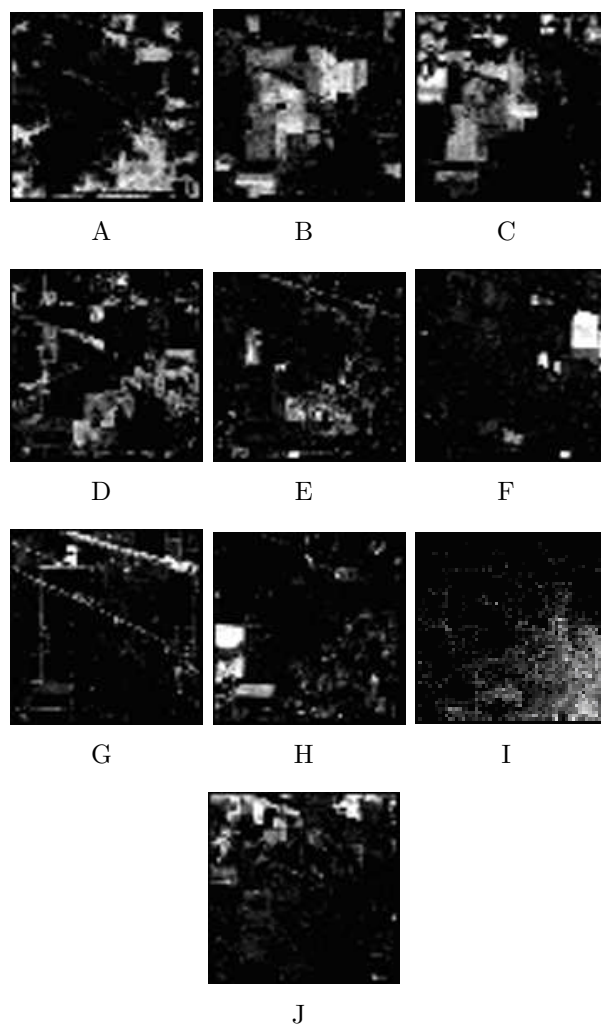


Figure 4-10. Abundance maps generated by SPICE on the Indian Pines data set.

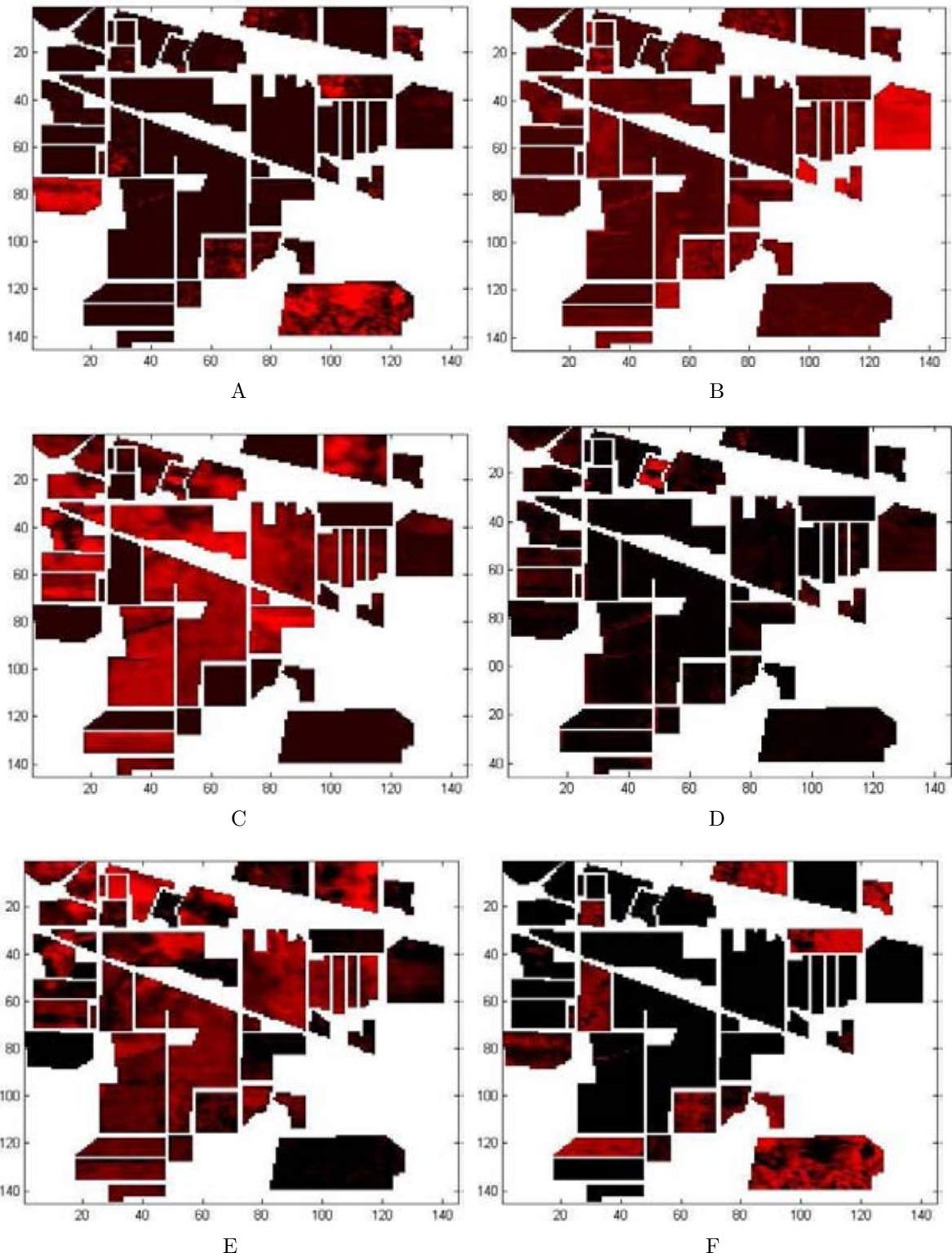


Figure 4-11. Abundance maps generated by SPICE on the labeled AVIRIS Indian Pines data set. Pixels in white correspond to unlabeled pixels. Remaining pixels range from black (abundance value of zero) to red (abundance of one).

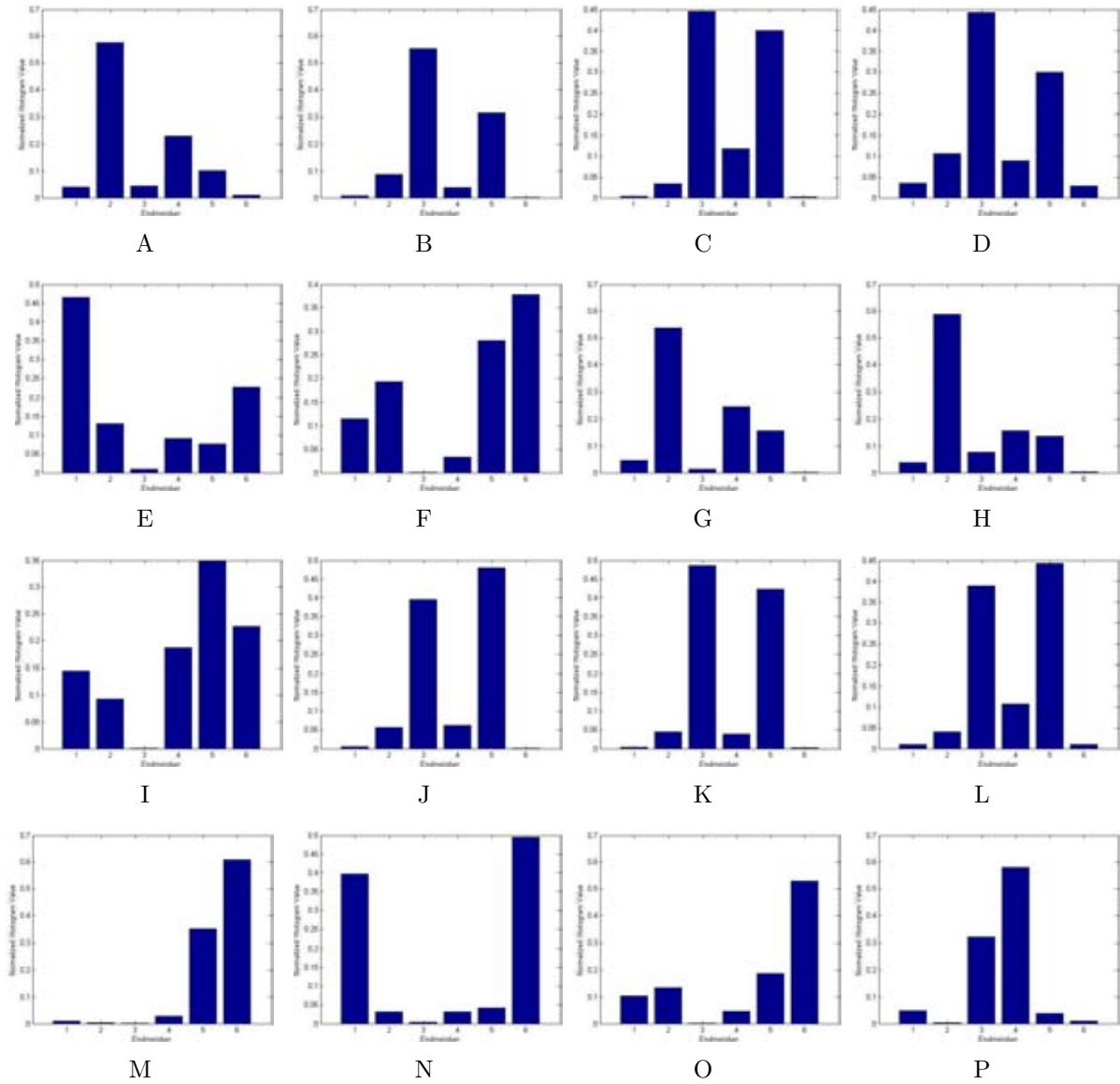


Figure 4-12. Histogram of SPICE endmember results on labeled AVIRIS Indian Pines data. Histograms show distribution of abundances values among endmembers in each ground truth class. Histograms were computed according to Equation 4-2. The sum of these histograms' Shannon's entropy values is 19.0. The histograms correspond to the following ground truth classes: (A) alfalfa, (B) corn-notill, (C) corn-min, (D) corn, (E) grass/pasture, (F) grass/trees, (G) grass/pasture-mowed, (H) hay-windrowed, (I) oats, (J) soybeans-notill, (K) soybeans-min, (L) soybean-clean, (M) wheat, (N) woods, (O) building-grass-trees-drive, and (P) stone-steel towers.

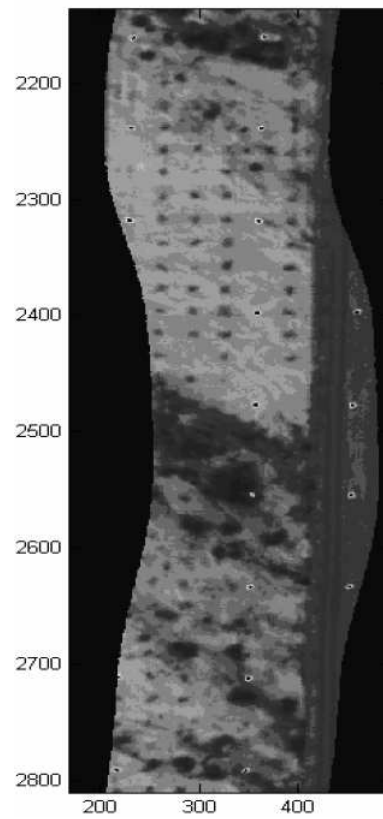


Figure 4-13. Subset at 9.19 microns of AHI hyperspectral image 2 including the overlap region

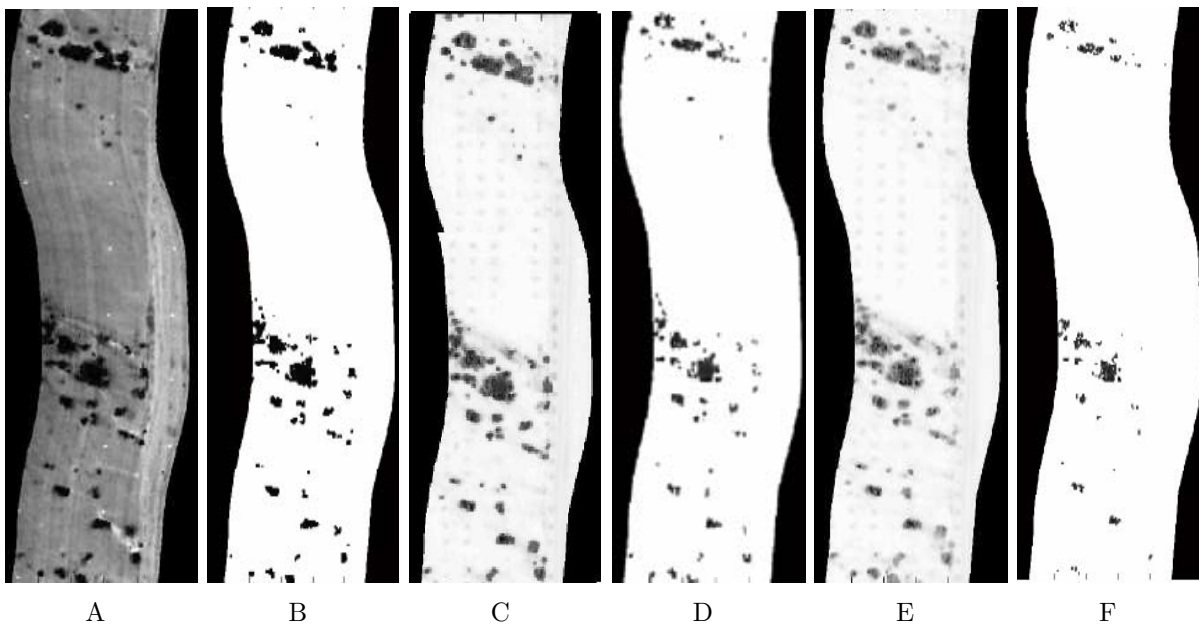


Figure 4-14. Blackbody masks created using SPICE and the clustering method. A) is the blackbody mask generated using SPICE and B) is the thresholded SPICE mask. C) is the mask generated using 4 clusters in the clustering method; D) is the thresholded version of this mask. E) is the mask generated using 5 clusters in the clustering method and F) is the thresholded version of this mask.

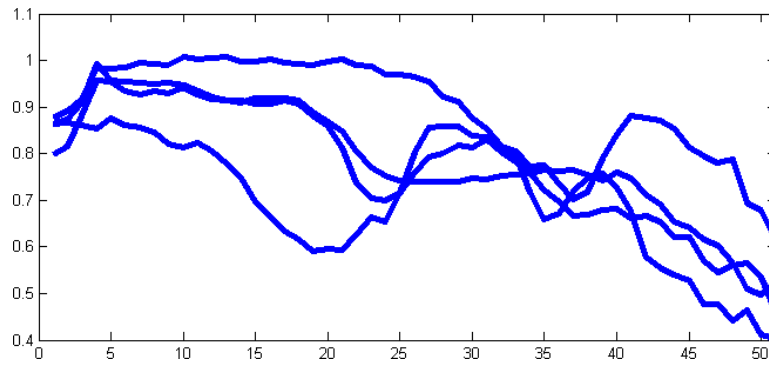


Figure 4-15. Endmembers used to generate simulated data set selected by hand from the AVIRIS Cuprite data set.

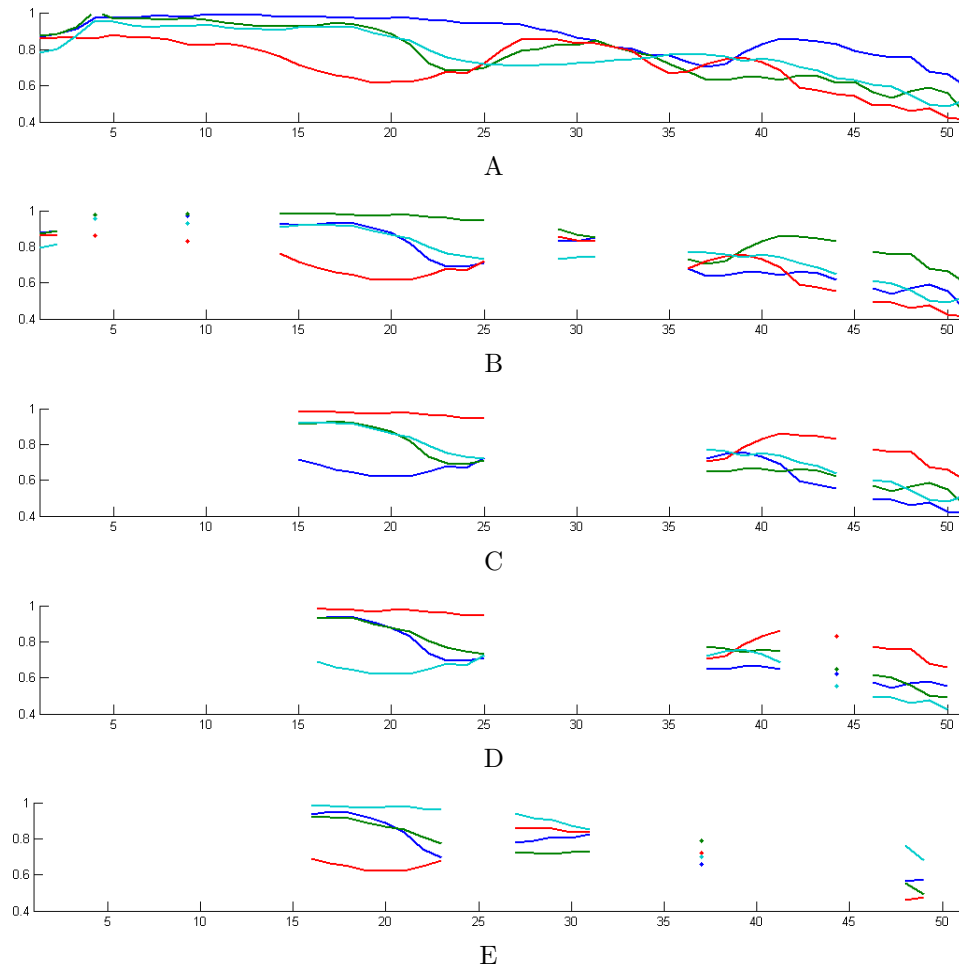


Figure 4-16. Endmembers determined using SPICE and B-SPICE with parameters $\Lambda = 0, 0.25, 0.5, 0.75,$ and 1 on the simulated data set.

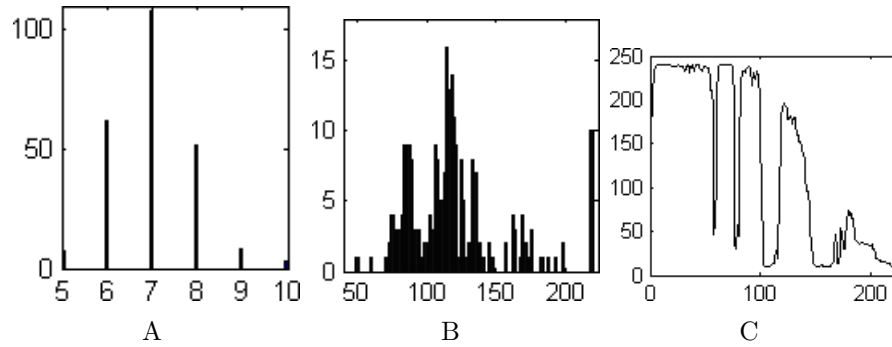


Figure 4-17. Histograms of (a) the number of endmembers (b) bands found and (c) the number of times each band is retained over 240 runs of B-SPICE using sampled parameter values.

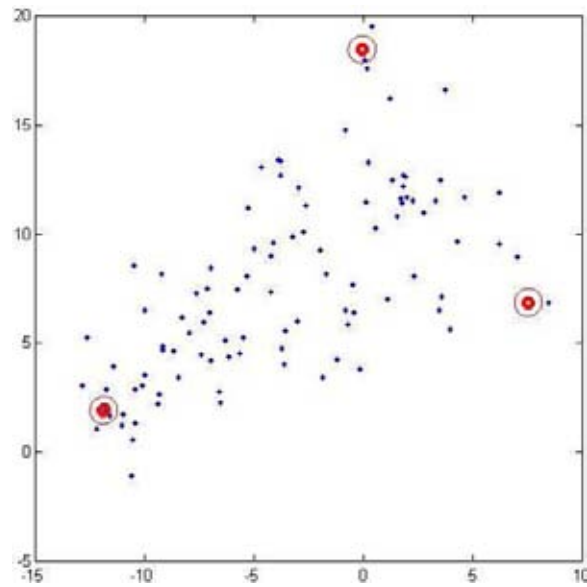


Figure 4-18. Results on two-dimensional triangle data found using ED. Blue points show the input data set. Red points are the mean endmembers of the endmember distributions found by ED. Red curves correspond to the 1st and 2nd standard deviations in each endmember distribution.

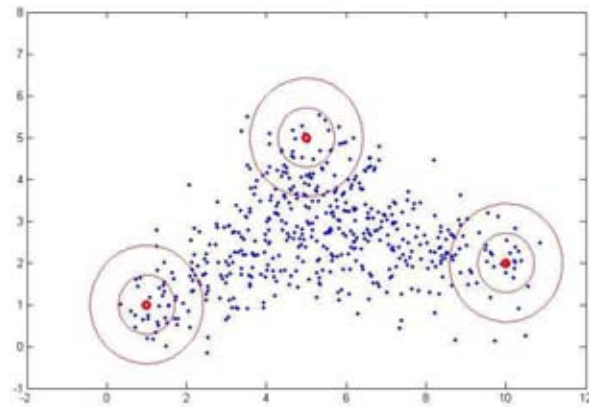


Figure 4-19. Data points generated from three endmember distributions. Blue points show the generated data set. Red points are the mean endmembers of the endmember distributions used to generate the data points. Red curves correspond to the 1st and 2nd standard deviations in each endmember distribution used to generate the data points.

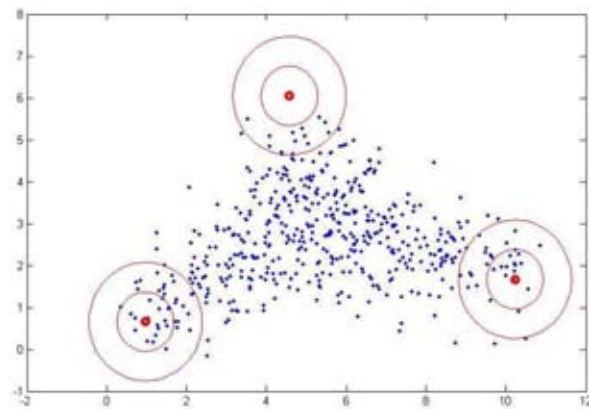


Figure 4-20. Results on two dimensional data using ED. Blue points show the generated data set. Red points are the mean endmembers of the endmember distributions found by ED. Red curves correspond to the 1st and 2nd standard deviations in each endmember distribution.

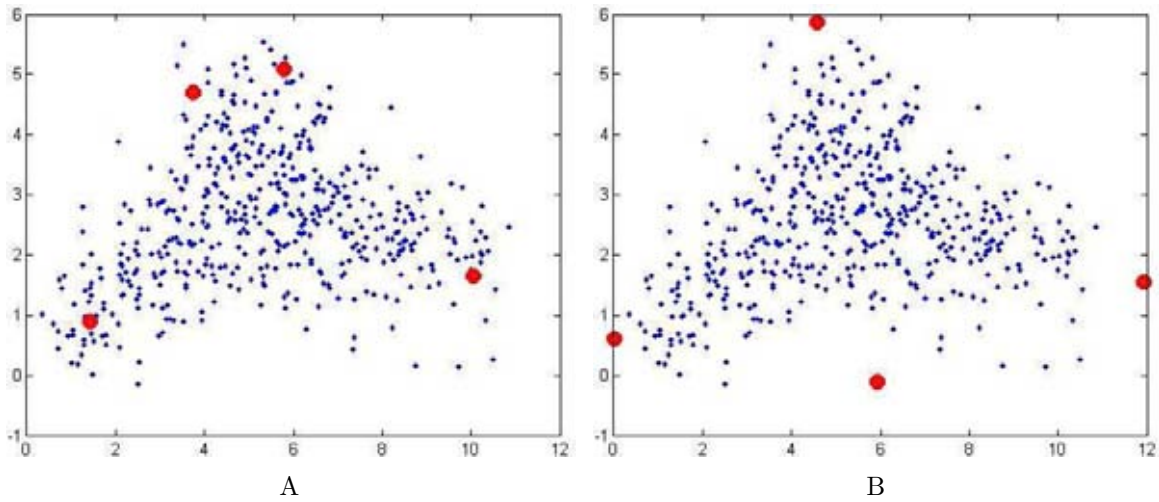


Figure 4-21. Results on two dimensional data using SPICE. Blue points show the generated data set. Red points are the endmembers found by SPICE.

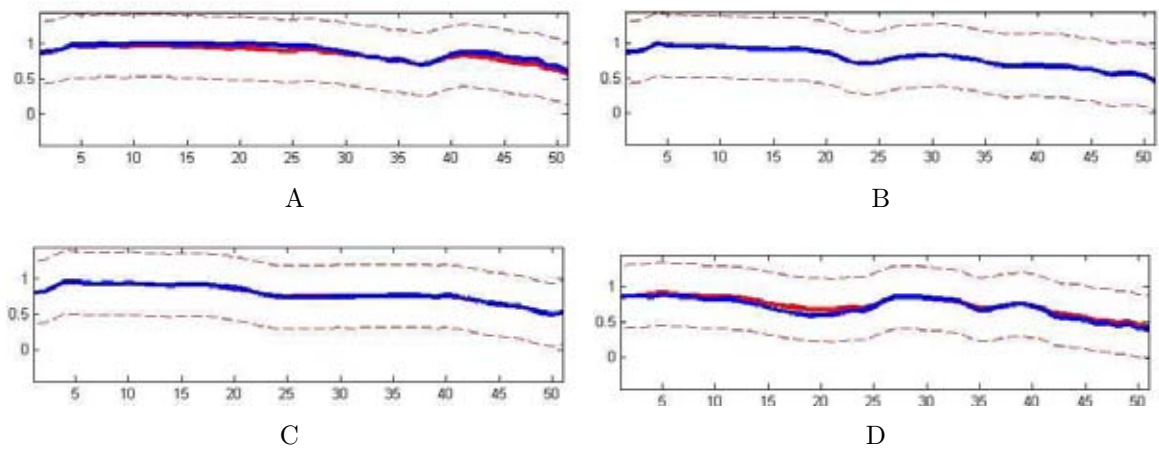


Figure 4-22. Results on simulated AVIRIS Cuprite data using ED. Solid blue curves show the true endmembers from which the data was generated. Solid red curves are the mean endmembers of the endmember distributions found by ED. Dashed red curves correspond to the 1st standard deviation in each endmember distribution.

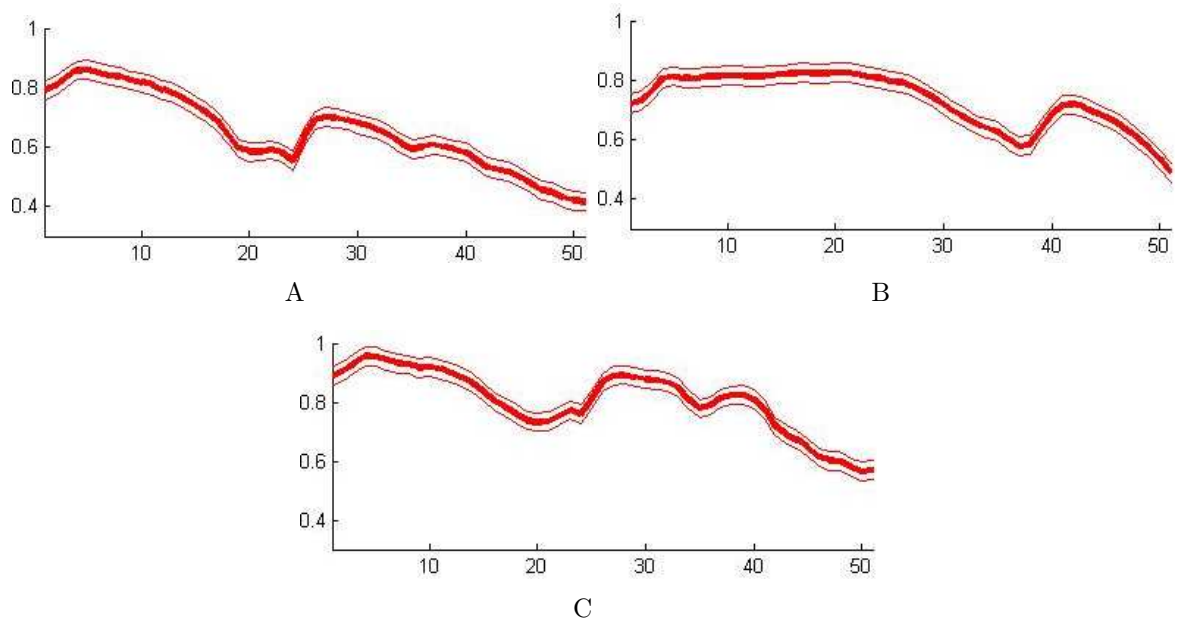


Figure 4-23. Results on a subset of AVIRIS Cuprite data found using ED. Solid red curves are the mean endmembers of the endmember distributions found by ED. Dashed red curves correspond to the 1st standard deviation in each endmember distribution.

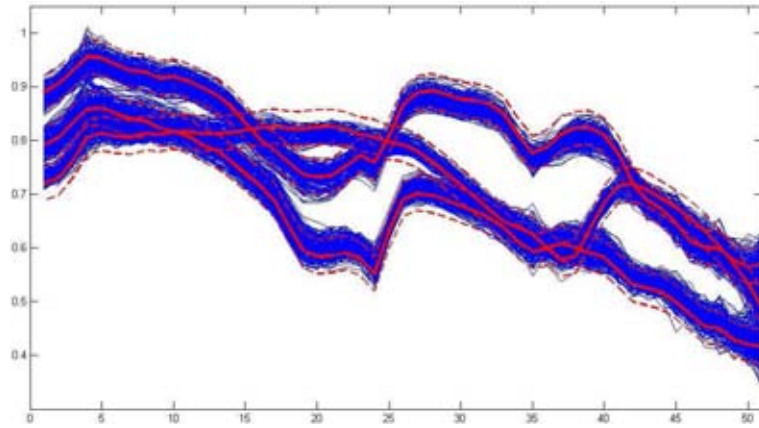


Figure 4-24. Results on a subset of AVIRIS Cuprite data found using ED. Blue curves show the input data set. Solid red curves are the mean endmembers of the endmember distributions found by ED. Dashed red curves correspond to the 1st standard deviation in each endmember distribution.

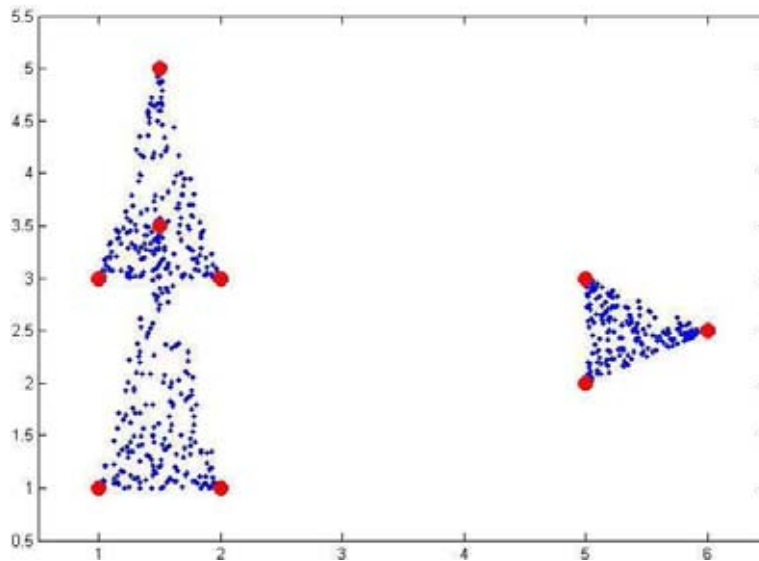


Figure 4-25. Two-dimensional data generated from three sets of endmembers. Blue points correspond to the input data set. Red points correspond to the endmembers from which the data was generated. Each triangle of data points was generated from three of the endmembers.

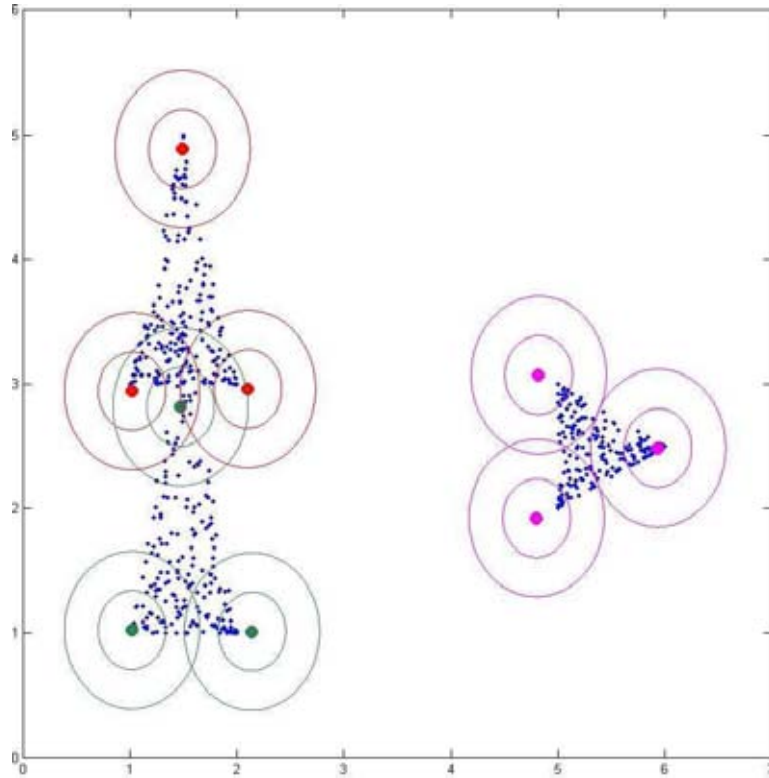


Figure 4-26. Two-dimensional data results found using PCE. Small blue points correspond to the input data set. Large points correspond to the mean endmembers for each endmember distribution. Thin curves correspond to the 1st and 2nd standard deviation curves from each endmember distribution. The color of each endmember distribution corresponds to the convex region to which it belongs.

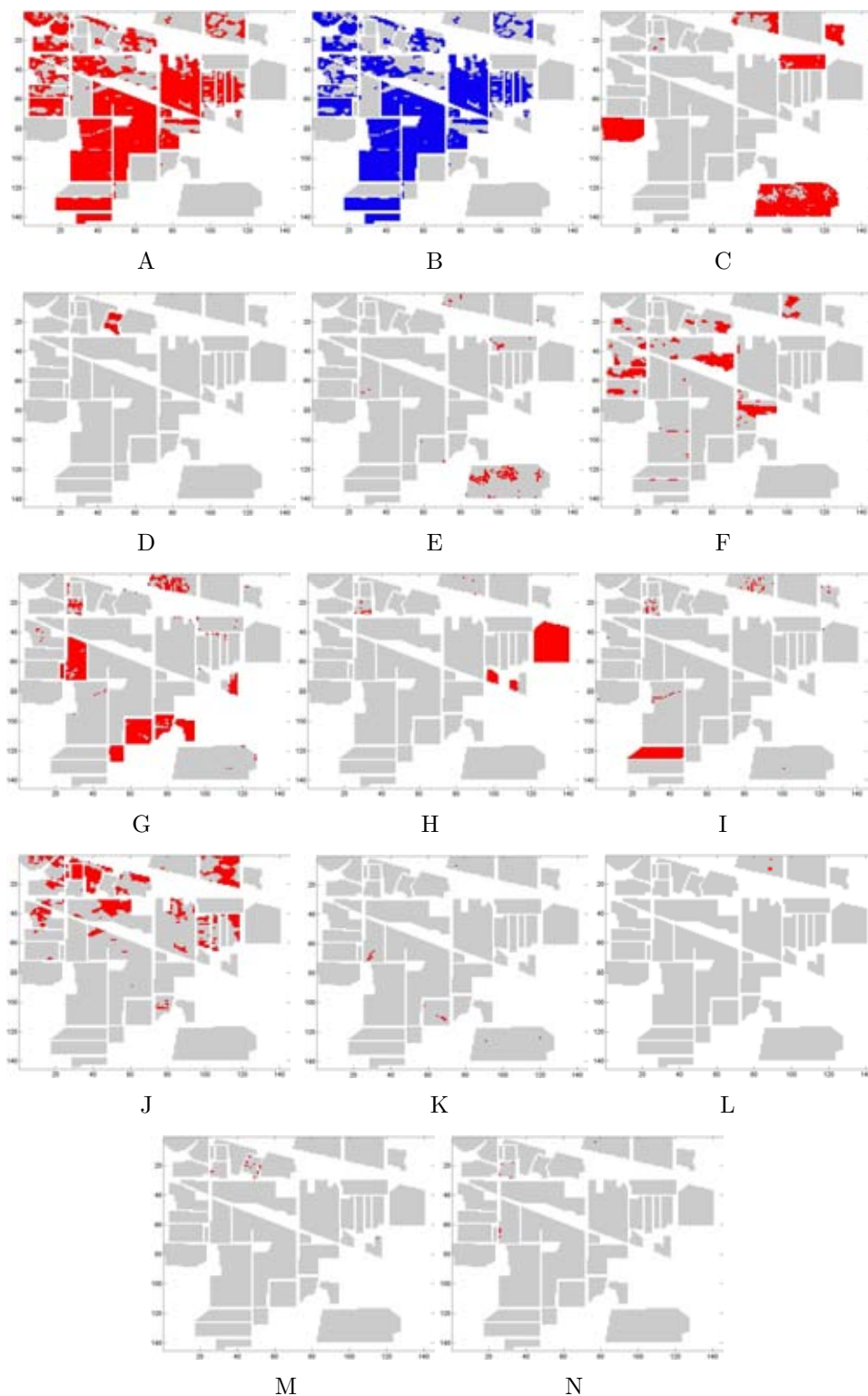


Figure 4-27. Abundance maps found using PCE on labeled PCA-reduced AVIRIS Indian Pines data. Pixels in white are unlabeled. Pixels in gray indicate pixels from another convex partition. Remaining pixels range from blue (abundance value of zero) to red (abundance value of one).

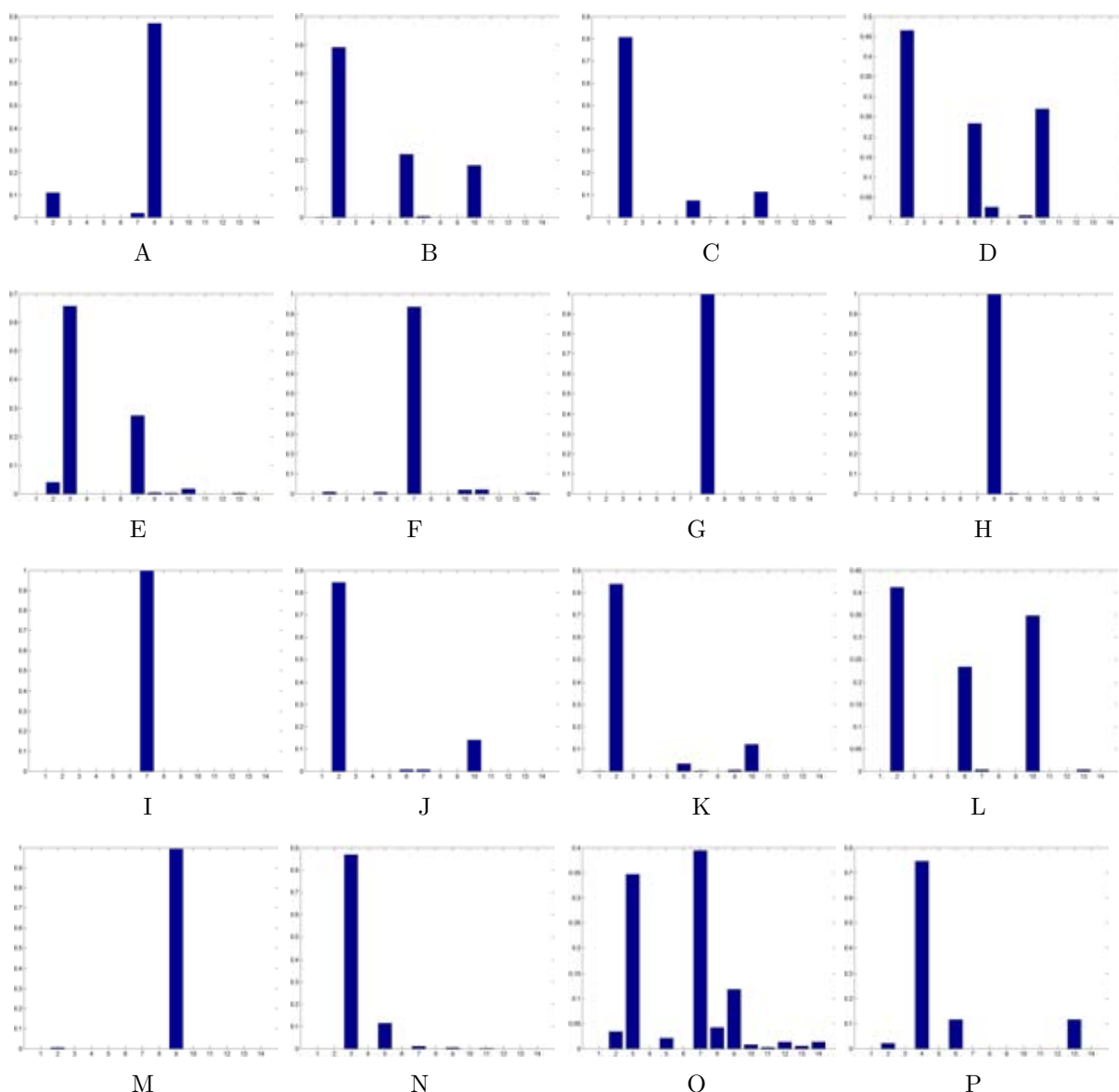


Figure 4-28. Histogram of PCE endmember results on labeled PCA-reduced AVIRIS Indian Pines data. Histograms show distribution of abundances values among endmembers in each ground truth class. Histograms were computed according to Equation 4-2. The sum of the histograms' Shannon's entropy values is 9.4. The histograms correspond to the following ground truth classes: (A) alfalfa, (B) corn-notill, (C) corn-min, (D) corn, (E) grass/pasture, (F) grass/trees, (G) grass/pasture-mowed, (H) hay-windrowed, (I) oats, (J) soybeans-notill, (K) soybeans-min, (L) soybean-clean, (M) wheat, (N) woods, (O) building-grass-trees-drive, and (P) stone-steel towers.

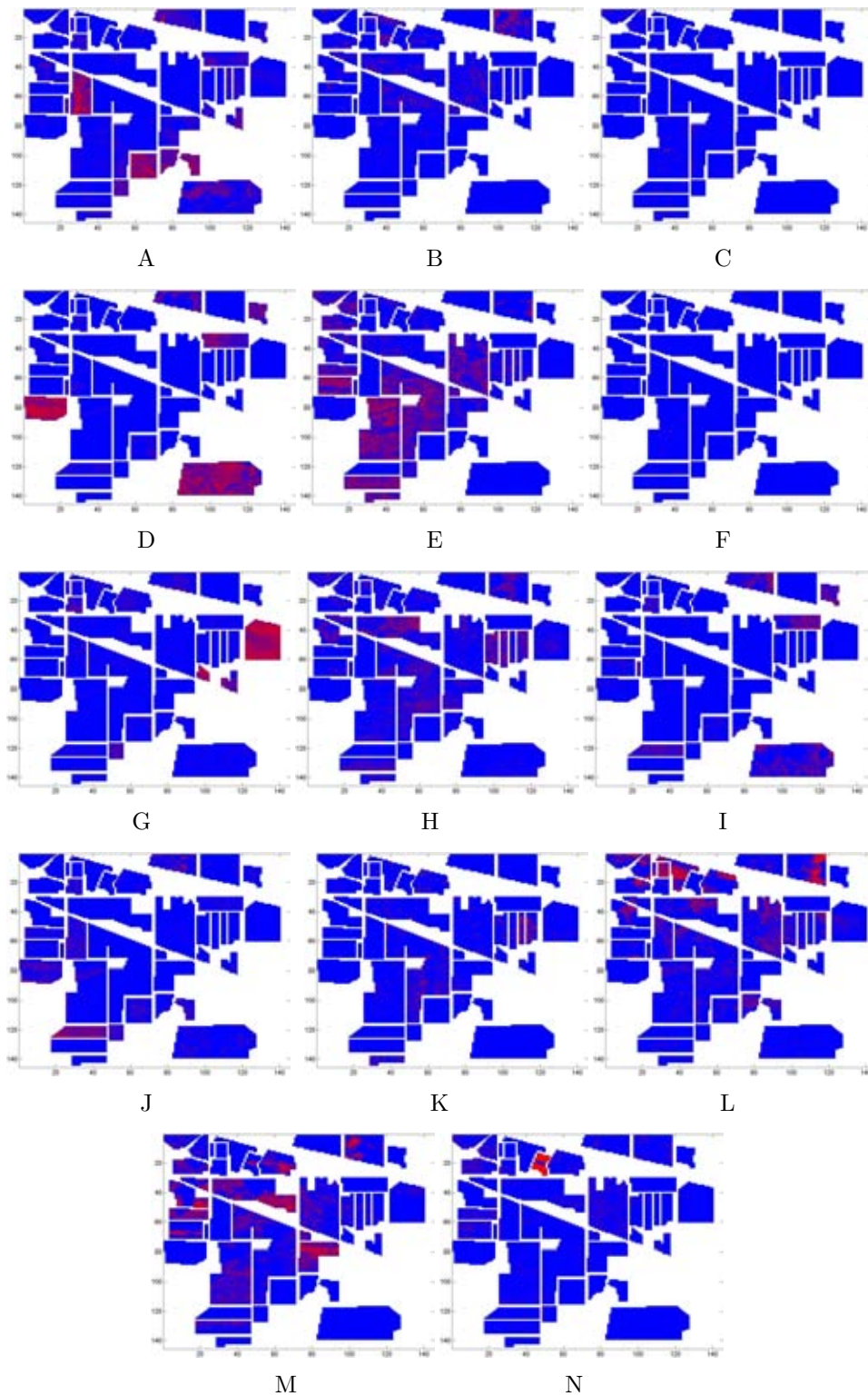


Figure 4-29. Abundance maps found using SPICE on labeled PCA-reduced AVIRIS Indian Pines data. Pixels in white are unlabeled. Remaining pixels range from blue (abundance value of zero) to red (abundance value of one).

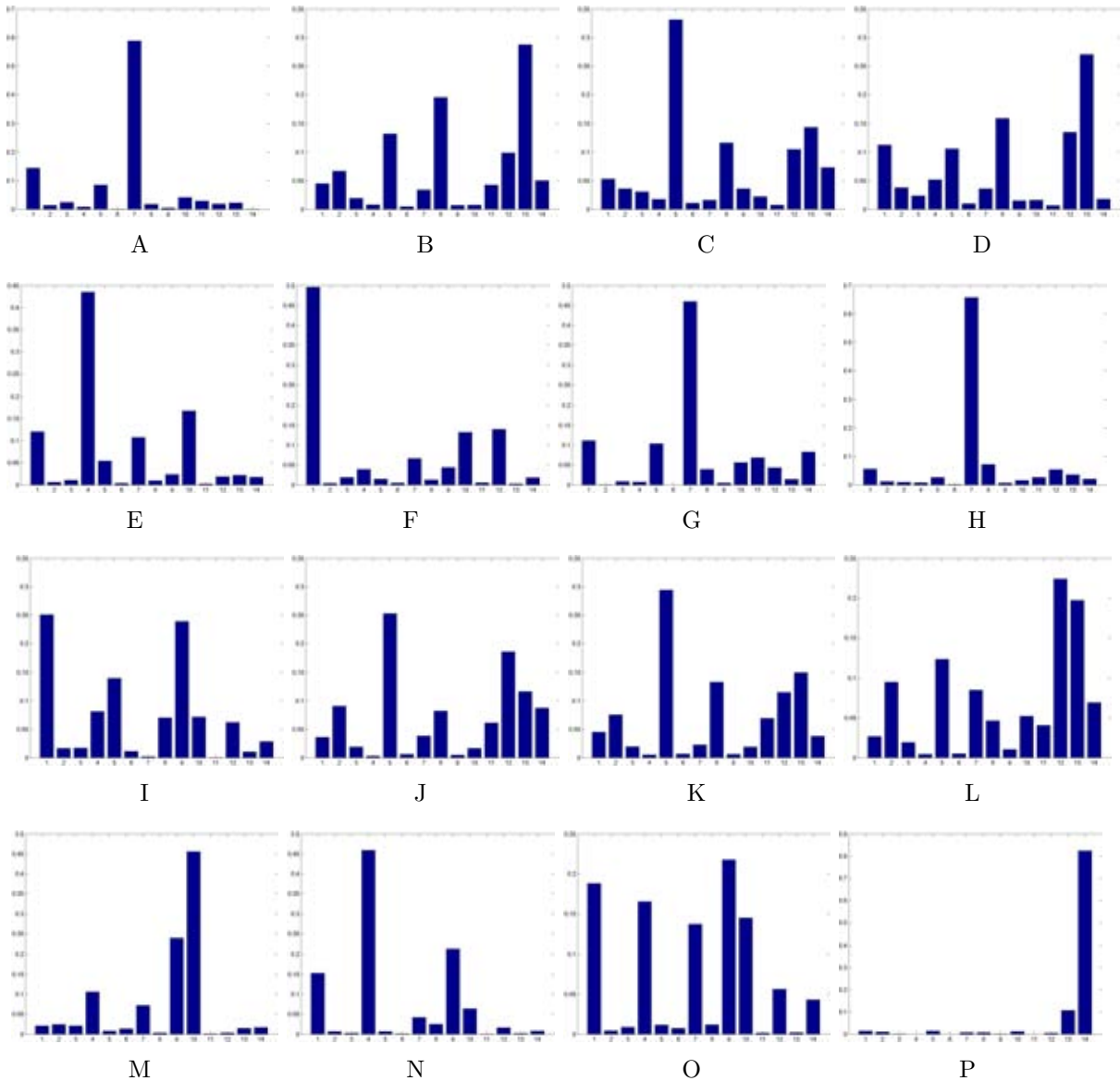


Figure 4-30. Histogram of SPICE endmember results on labeled PCA-reduced AVIRIS Indian Pines data. Histograms show distribution of abundances values among endmembers in each ground truth class. Histograms were computed according to Equation 4-2. The sum of the histograms' Shannon's entropy values is 29.2. The histograms correspond to the following ground truth classes: (A) alfalfa, (B) corn-notill, (C) corn-min, (D) corn, (E) grass/pasture, (F) grass/trees, (G) grass/pasture-mowed, (H) hay-windrowed, (I) oats, (J) soybeans-notill, (K) soybeans-min, (L) soybean-clean, (M) wheat, (N) woods, (O) building-grass-trees-drive, and (P) stone-steel towers.

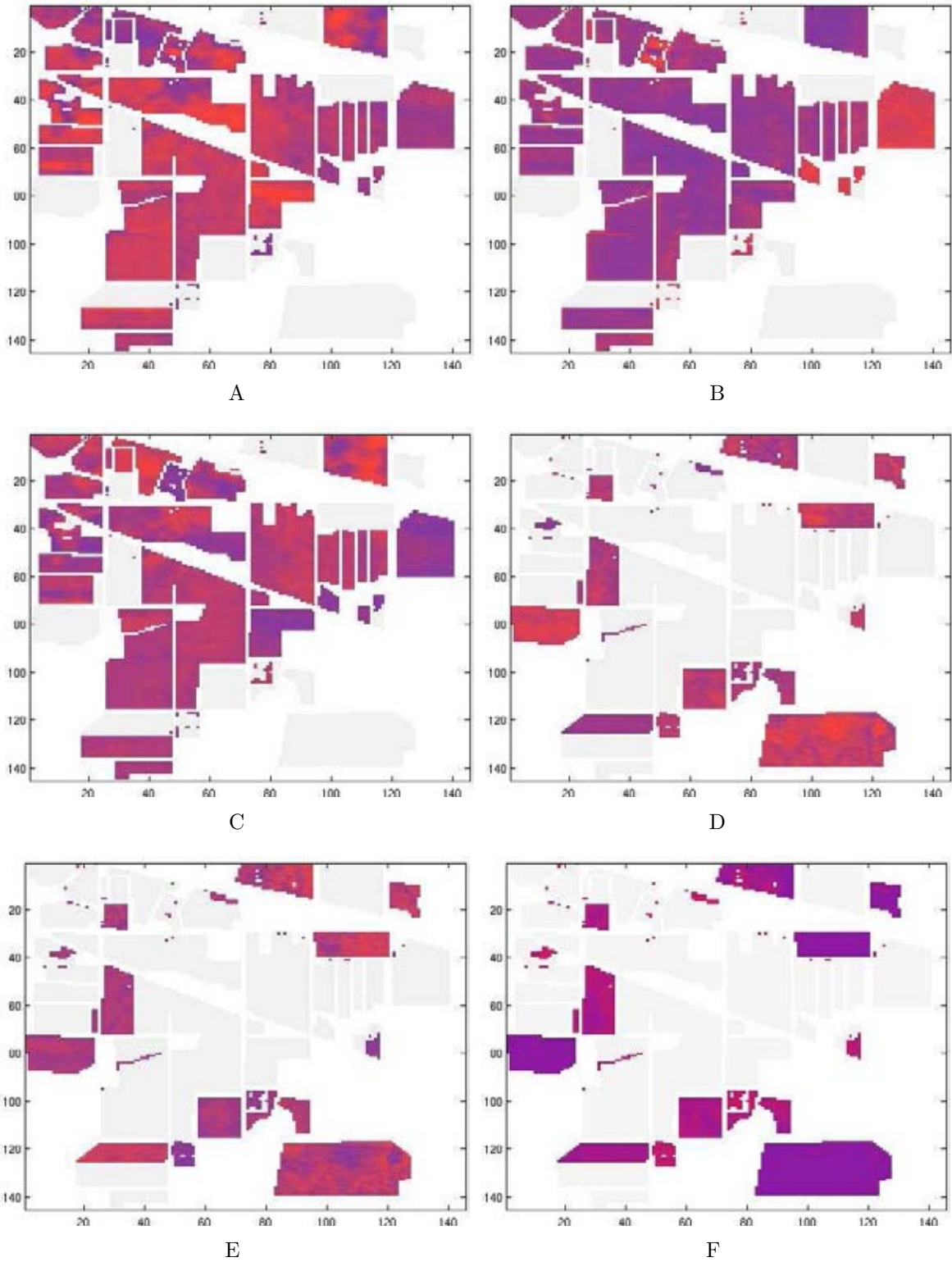


Figure 4-31. Abundance maps found using PCE on labeled AVIRIS Indian Pines data with hierarchical dimensionality reduction. Pixels in white are unlabeled. Pixels in gray indicate pixels from another convex partition. Remaining pixels range from blue (abundance value of zero) to red (abundance value of one).

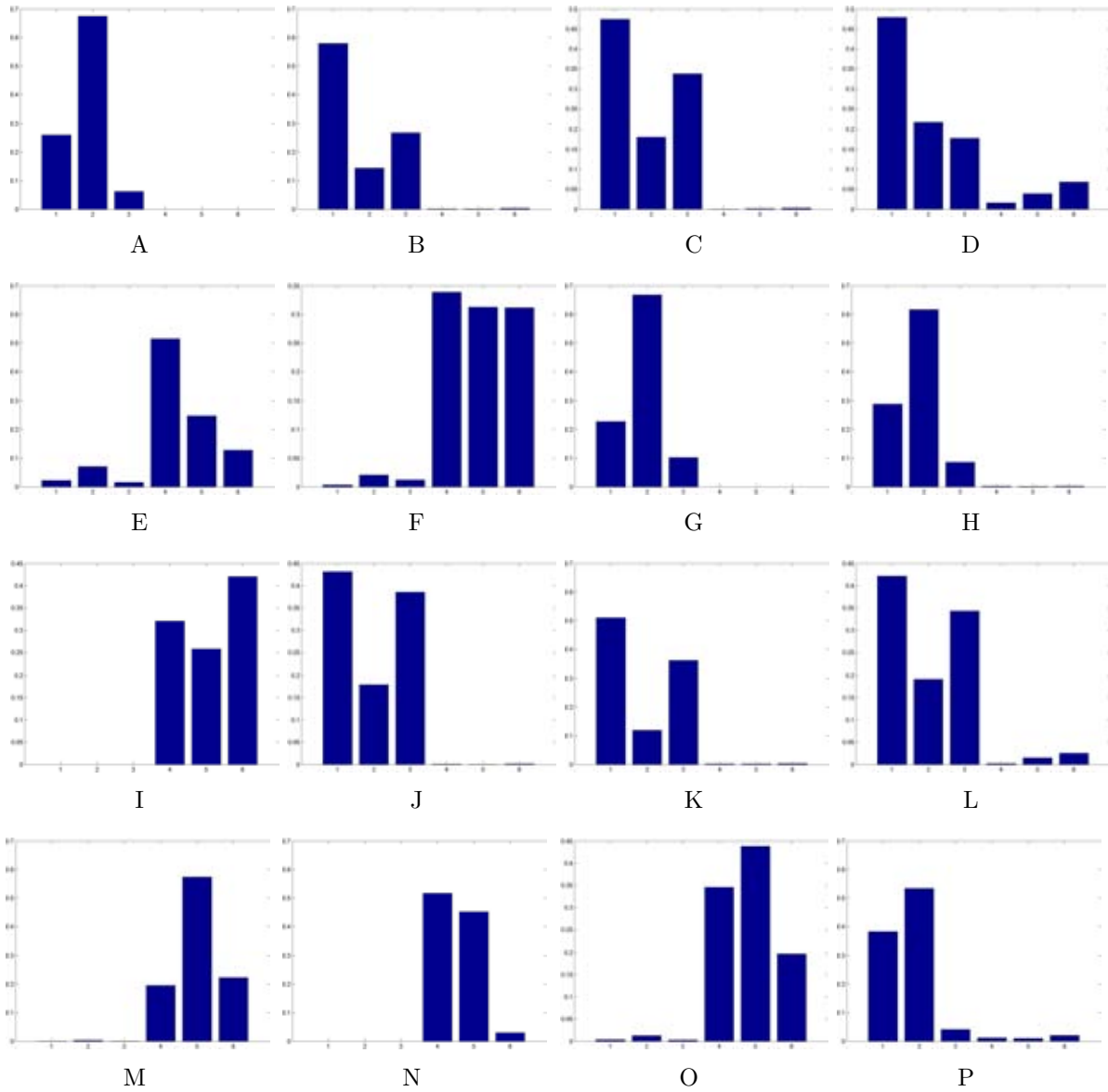


Figure 4-32. Histogram of PCE endmember results on labeled AVIRIS Indian Pines data with hierarchical dimensionality reduction. Histograms show distribution of abundances values among endmembers in each ground truth class. The sum of the histogram's Shannon's entropy values is 16.3. Histograms were computed according to Equation 4-2. The histograms correspond to the following ground truth classes: (A) alfalfa, (B) corn-notill, (C) corn-min, (D) corn, (E) grass/pasture, (F) grass/trees, (G) grass/pasture-mowed, (H) hay-windrowed, (I) oats, (J) soybeans-notill, (K) soybeans-min, (L) soybean-clean, (M) wheat, (N) woods, (O) building-grass-trees-drive, and (P) stone-steel towers.

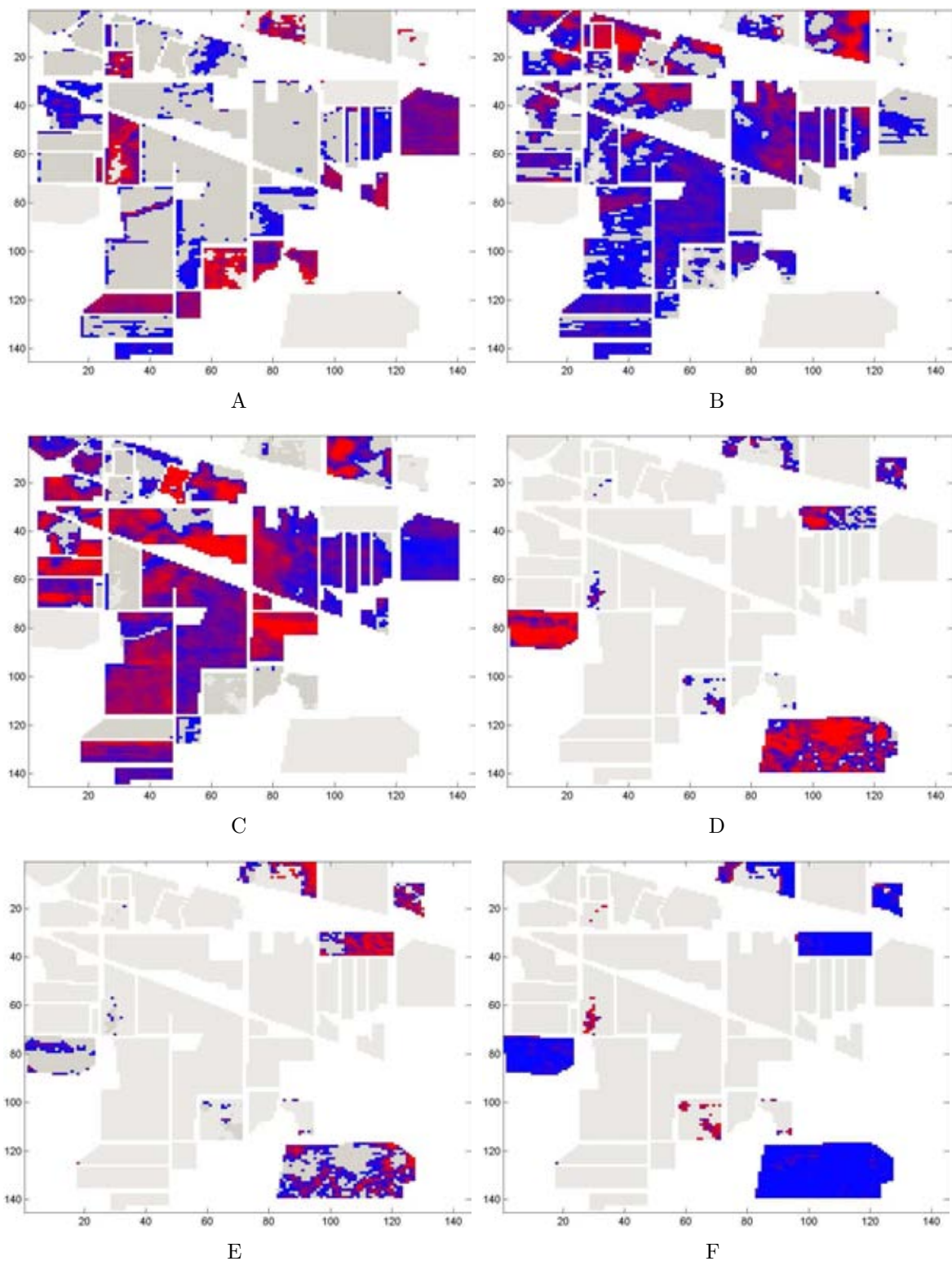


Figure 4-33. Abundance maps found using PCE on labeled AVIRIS Indian Pines data. Pixels in white are unlabeled. Pixels in gray indicate pixels from another convex partition. Remaining pixels range from blue (abundance value of zero) to red (abundance value of one).

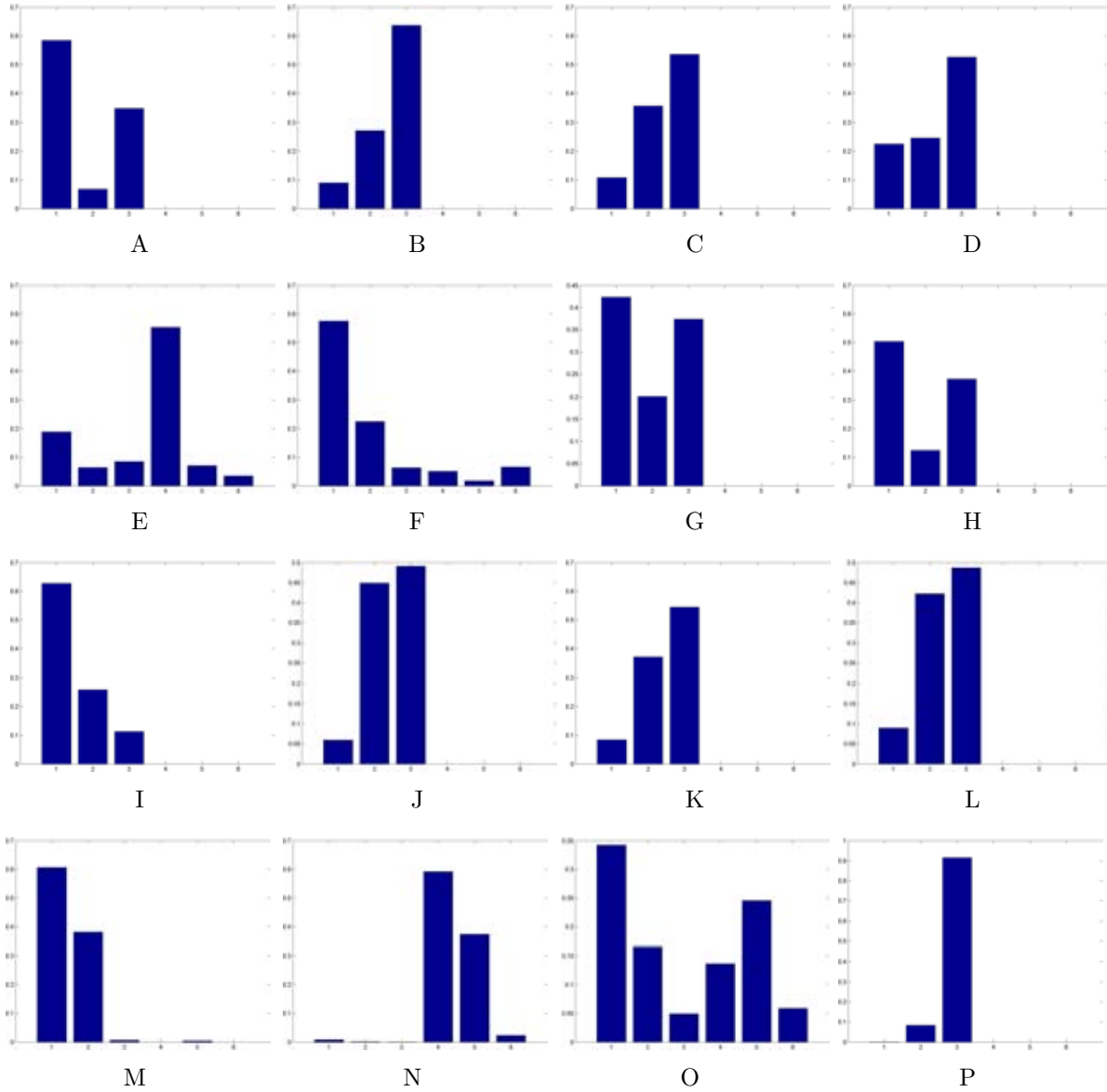


Figure 4-34. Histogram of PCE endmember results on labeled AVIRIS Indian Pines data. Histograms show distribution of abundances values among endmembers in each ground truth class. Histograms were computed according to Equation 4-2. The sum of the histogram's Shannon's entropy values is 15.4. The histograms correspond to the following ground truth classes: (A) alfalfa, (B) corn-notill, (C) corn-min, (D) corn, (E) grass/pasture, (F) grass/trees, (G) grass/pasture-mowed, (H) hay-windrowed, (I) oats, (J) soybeans-notill, (K) soybeans-min, (L) soybean-clean, (M) wheat, (N) woods, (O) building-grass-trees-drive, and (P) stone-steel towers.

CHAPTER 5

CONCLUSION

Four novel methods for hyperspectral image spectroscopy based on Bayesian methodologies were developed and tested. The Sparsity Promoting Iterated Constrained Endmembers (SPICE) algorithm incorporates sparsity promoting priors to estimate the number of endmembers while simultaneously performing endmember detection and spectral unmixing. Previously, most endmember detection algorithms required the number of endmembers in advance. The algorithm's sparsity promoting priors drive the proportion values of unneeded endmembers to zero allowing SPICE to remove those endmembers without any effect on pixel representation via endmembers and proportions.

The Band Selecting Sparsity Promoting Iterated Constrained Endmember (B-SPICE) algorithm extends SPICE to include hyperspectral band selection. Sparsity promoting priors are applied to band weights to determine the hyperspectral bands which distinguish between the endmembers in the data set. Therefore, B-SPICE autonomously determines the number of needed wavelengths. In addition, B-SPICE is able to identify the needed wavelengths, perform endmember detection and determine the number of endmembers needed, simultaneously.

The Endmember Distribution (ED) detection algorithm learns endmember distributions to incorporate spectral variability into the endmember detection model. Previously, endmember detection algorithms constrained endmembers to be single spectral vectors. By utilizing endmember distributions, several pixels of the same material with some spectral variation can all be identified as having a full abundance for the same endmember.

The Piece-wise Convex Endmember (PCE) detection algorithm used the Dirichlet process to learn the number of convex regions needed to describe an input hyperspectral scene. For each convex region, an individual set of endmember distributions and proportion values were determined. In contrast, previous endmember detection algorithms applied the same set of endmembers to every data point in a scene. Using PCE, different

portions of an input hyperspectral scene can be represented using a separate set of endmembers. This results in better suited endmembers for all of the various regions in an input image.

During development and testing of the methods, several interesting areas for future research were uncovered. Currently, the ED algorithm assumes a constant and known covariance matrix for each endmember distribution. Investigations into methods of learning appropriate covariance matrices for each endmember distribution can be done. By learning covariance matrices, endmembers distributions can be further tailored to the input data set. Also, both the B-SPICE and PCE algorithms utilize optimization schedules and many parameter values. Studies on methods to determine the appropriate optimization schedules and parameter values with regard to the input data set can be conducted.

The PCE algorithm currently assigns each data point to a single partition. Investigations into methods of allowing data points to have partial membership in several partitions can be conducted. By allowing partial memberships, overlapping clusters may be more likely to be found using PCE. The membership values could also provide additional insight into how well-suited each set of endmembers are for an individual data point.

REFERENCES

- C. E. Antoniak. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174, 1974.
- R. Archibald and G. Fann. Feature selection and classification of hyperspectral images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 4(4):674–677, Oct. 2007.
- AVIRIS. (2004, Sep) AVIRIS free standard data products. Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA. URL <http://aviris.jpl.nasa.gov/html/aviris.freedata.html>.
- M. Berman, H. Kiiveri, R. Lagerstrom, A. Ernst, R. Donne, and J. F. Huntington. ICE: A statistical approach to identifying endmembers in hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 42:2085–2095, Oct. 2004.
- C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
- J. Boardman, F. Kruse, and R. Green. Mapping target signatures via partial unmixing of AVIRIS data. In R. Green, editor, *Summaries of the 5th Annu. JPL Airborne Geoscience Workshop*, volume 1, pages 23–26, Pasadena, CA, 1995. JPL Publ.
- L. M. Bruce, C. H. Koger, and J. Li. Dimensionality reduction of hyperspectral data using discrete wavelet transform feature extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 40(10):2331–2338, Oct. 2002.
- G. Casella and E. I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):167–174, Aug. 1992.
- C.-I. Chang and Q. Du. Estimation of number of spectrally distinct signal sources in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 42(3):608–619, Mar. 2004.
- C.-I. Chang and S. Wang. Constrained band selection for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 44(6):1575–1585, June 2006.
- C.-I. Chang, Q. Du, T.-L. Sun, and M. L. G. Althouse. A joint band prioritization and band decorrelation approach to band selection for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 37(6):2631–2641, Nov. 1999.
- S.-S. Chiang, C.-I. Chang, and I. W. Ginsberg. Unsupervised hyperspectral image analysis using independent component analysis. In *Proceedings of the IEEE: Geoscience and Remote Sensing Symposium*, pages 3136–3138, July 2000.
- S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49(4):327–335, Nov. 1995.

- R. N. Clark, A. Swayze, R. Wise, K. E. Livo, T. M. Heofen, R. F. Kokaly, and S. J. Sutley. USGS digital spectral library (splib05a), January 2004. URL <http://speclab.cr.usgs.gov/spectral-lib.html>.
- M. D. Craig. Minimum-volume transforms for remotely sensed data. *IEEE Transactions on Geoscience and Remote Sensing*, 32(3):542–552, May 1994.
- S. DeBacker, P. Kempeneers, W. Debruyne, and P. Scheunders. A band selection technique for spectral classification. *IEEE Geoscience and Remote Sensing Letters*, 2(3):319–323, July 2005.
- L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986.
- H. Du, H. Qi, X. Wang, R. Ramanath, and W. E. Snyder. Band selection using independent components analysis for hyperspectral image processing. In *Proceedings of the IEEE: 32nd Applied Imagery Pattern Recognition Workshop*, 2003.
- R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2 edition, Oct. 2001.
- T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.
- M. A. T. Figueiredo. Adaptive sparseness for supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25:1150–1159, September 2003.
- A. N. French, T. J. Schmugge, and W. P. Kustas. Discrimination of senescent vegetation using thermal emissivity contrast. *Remote Sensing of Environment*, 74:249–254, 2000.
- M. Grana and M. J. Gallego. Associative morphological memories for spectral unmixing. In *European Symposium on Artificial Neural Networks*, pages 481–486, Apr. 2003.
- M. Grana, P. Sussner, and G. Ritter. Associative morphological memories for endmember detection in spectral unmixing. In *The 12th IEEE International Conference on Fuzzy Systems*, volume 2, pages 1285–1290, May 2003.
- M. Grana, C. Hernandez, and J. Gallego. A single individual evolutionary strategy for endmember search in hyperspectral images. *Information Sciences*, 161(3-4):181–197, Apr. 2004.
- M. Grana, C. Hernandez, and A. d’Anjou. An evolutionary algorithm based on morphological associative memories for endmember selection in hyperspectral images. In M. Grana, R. Duro, A. d’Anjou, and P. P. Wang, editors, *Information Processing with Evolutionary Algorithms*, pages 45–59. Springer Berlin Heidelberg, 2005.
- A. A. Green, M. Berman, P. Switzer, and M. D. Craig. A transformation for ordering multispectral data in terms of image quality with implications for noise removal. *IEEE Transactions on Geoscience and Remote Sensing*, 26:65–73, Jan. 1988.

- B. Guo, S. R. Gunn, R. I. Damper, and J. D. B. Nelson. Band selection for hyperspectral image classification using mutual information. *IEEE Geoscience and Remote Sensing Letters*, 3(4):522–526, Oct. 2006.
- T. Han, D. G. Goodenough, A. Dyk, and H. Chen. Hyperspectral feature selection for forest classification. In *Proceedings of the IEEE: Geoscience and Remote Sensing Symposium*, pages 1471–1474, 2004.
- J. C. Harsanyi and C.-I. Chang. Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach. *IEEE Transactions on Geoscience and Remote Sensing*, 32(4):779–785, July 1994.
- S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, 2 edition, 1999.
- G. Heo and P. Gader. KG-FCM: Kernel-based global fuzzy c-means clustering algorithm. Technical report, Department of Computer and Information Science and Engineering, University of Florida, 2008.
- R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1985.
- R. Huang and M. He. Band selection based on feature weighting for classification of hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 2(2):156–159, Apr. 2005.
- A. Hyvarinen and E. Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13:411–430, Mar. 2000.
- A. Ifarraguerri and C.-I. Chang. Multispectral and hyperspectral image analysis with convex cones. *IEEE Transactions on Geoscience and Remote Sensing*, 73(2):756–770, Mar. 1999.
- S. Jain and R. M. Neal. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. Technical Report 2003, University of Toronto, Toronto, ON, Canada, Jul. 2000.
- P. Kealy and S. Hook. Separating temperature and emissivity in thermal infrared multispectral scanner data: Implications for recovering land surface temperatures. *IEEE Transactions on Geoscience and Remote Sensing*, 21(10):2127–2132, 2000.
- N. Keshava. Best bands selection for detection in hyperspectral processing. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 3149–3152, May 2001.
- N. Keshava. Distance metrics and band selection in hyperspectral processing with applications to material identification and spectral libraries. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1552–1565, 2004.
- N. Keshava and J. F. Mustard. Spectral unmixing. *IEEE Signal Processing Magazine*, 19: 44–57, 2002.

- B. Krishnapuram, A. J. Harternik, L. Carin, and M. A. T. Figueiredo. A Bayesian approach to joint feature selection and classifier design. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(9):1105–1111, Sep. 2004.
- S. Kumar, J. Ghosh, and M. M. Crawford. Best-bases feature extraction algorithms for classification of hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 39(7):1368–1379, July 2001.
- D. Lee and H. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
- D. Lee and H. Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems 13*, pages 556–562, 2000.
- J. B. Lee, A. S. Woodyatt, and M. Berman. Enhancement of high spectral resolution remote-sensing data by a noise-adjusted principal components transform. *IEEE Transactions on Geoscience and Remote Sensing*, 28(3):295–304, May 1990.
- H.-D. Lin and L. M. Bruce. Projection pursuits for dimensionality reduction of hyperspectral signals in target recognition applications. In *Proceedings of the IEEE: Geoscience and Remote Sensing Symposium*, pages 960–963, 2004.
- C. Liou and K. O. Yang. Unsupervised classification of remote sensing imagery with non-negative matrix factorization. In *Proceedings of Twelfth International Conference on Neural Inf. Proc.*, pages 295–304, 2005.
- P.G. Lucey, T. J. Williams, M. Mignard, J. Julian, D. Kobubun, G. Allen, D. Hampton, W. Schaff, M. J. Schlagen, E. M. Winter, W. B. Kendall, A. D. Stocker, K. A. Horton, and A. P. Bowman. Ahi: An airborne long-wave infrared hyperspectral imager. In *Proceedings of the SPIE*, volume 3431, pages 36–43, Nov 1998.
- LYNXSAR. General Atomics, San Diengo, CA, URL <http://www.LYNXSAR.com>.
- D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- D. Manolakis, D. Marden, and G. A. Shaw. Hyperspectral image processing for automatic target detection applications. *Lincoln Laboratory Journal*, 14(1):79–116, 2003.
- A. Martinez-Usó, F. Pla, P. Garcia-Sevilla, and J. M. Sotoca. Automatic band selection in multispectral images using mutual information-based clustering. In *Proceedings of the 11th Iberoamerican Congress on Pattern Recognition*, pages 644–654, Cancun, Mexico, Nov. 2006.
- A. Martinez-Usó, F. Pla, J. M. Sotoca, and P. Garcia-Sevilla. Clustering-based hyperspectral band selection using information measures. *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):4158–4171, Dec. 2007.

- A. Mendez-Vazquez and P. Gader. Sparsity promotion models for the Choquet integral. In *Proceedings of the IEEE: Symposium on Foundations of Computations Intelligence*, pages 454–459, Apr. 2007.
- L. Miao and H. Qi. Endmember extraction from highly mixed data using minimum volume constrained nonnegative matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 45(3):765–777, Mar. 2007.
- L. Miao, H. Qi, and H. Szu. Unsupervised decomposition of mixed pixels using the maximum entropy principle. In *Proceedings of the 18th International Conference On Pattern Recognition*, volume 1, pages 1067–1070, 2006.
- D. S. Myers. Hyperspectral endmember detection using morphological autoassociative memories. Masters thesis, University of Florida, Gainesville, Florida, 2005.
- J. Nascimento and J. Bioucas-Dias. Blind hyperspectral unmixing. In *Proceedings of the VI Conference on Telecommunications*, pages 617–620, Peniche, Portugal, May 2007a.
- J. Nascimento and J. Bioucas-Dias. Hyperspectral unmixing algorithm via dependent component analysis. In *Proceedings of IEEE: Geoscience and Remote Sensing Symposium*, Barcelona, Spain, July 2007b.
- J. M. P. Nascimento and J. M. Bioucas-Dias. Does independent component analysis play a role in unmixing hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(1):175–187, Jan. 2005a.
- J. M. P. Nascimento and J. M. Bioucas-Dias. Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 43(4):898–910, Apr. 2005b.
- R. M. Neal. Bayesian mixture modeling by Monte Carlo simulation. Technical Report CRG-TR-91-2, University of Toronto, Toronto, ON, Canada, Jun. 1991.
- R. M. Neal. Markov chain sampling methods for Dirichlet process mixture models. Technical Report 9815, University of Toronto, Toronto, ON, Canada, Sep. 1998.
- N. Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber.*, 9:62–66, 1979.
- V. P. Pauca, J. Piper, and R. J. Plemmons. Nonnegative matrix factorization for spectral data analysis. *Linear Algebra Applications*, 416(1):321–331, Jul. 2005.
- G. M. Petrie, P. G. Heasler, and T. A. Warner. Optimal band selection strategies for hyperspectral data sets. In *Proceedings of the IEEE: Geoscience and Remote Sensing Symposium 1998: Sensing and Managing the Environment*, pages 1582–1584, 1998.
- A. Plaza, P. Martinez, R. Perez, and J. Plazas. Spatial/spectral endmember extraction by multidimensional morphological operators. *IEEE Transactions on Geoscience and Remote Sensing*, 40(9):2025–2041, September 2002.

- A. Ranganathan. The Dirichlet process mixture (DPM) model. Internet draft, October 2006. URL <http://www.cc.gatech.edu/~ananth/docs/dirichlet.pdf>.
- C. Rasmussen. The infinite Gaussian mixture model. In S. A. Solla, T. K. Leen, and K. R. Muller, editors, *Advances in Neural Information Processing Systems*, volume 12, pages 554–560. MIT Press, 2000.
- M. Riedmann and E. J. Milton. Supervised band selection for optimal use of data from airborne hyperspectral sensors. In *Proceedings of the IEEE: Geoscience and Remote Sensing Symposium*, volume 3, pages 1770–1772, Jul. 2003.
- G. X. Ritter and P. D. Gader. Fixed points of lattice transforms and lattice associative memories. *Advances in Imaging and Electron Physics*, 144:165–242, 2006.
- G. X. Ritter and G. Urcid. Fast autonomous endmember determination using lattice algebra, 2008. Manuscript submitted for publication.
- D. M. Rogge, B. Rivard, J. Zhang, A. Sanchez, J. Harris, and J. Feng. Integration of spatial-spectral information for the improved extraction of endmembers. *Remote Sensing of Environment*, 110:287–303, 2007.
- S. B. Serpico and L. Bruzzone. A new search algorithm for feature selection in hyperspectral remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 39(7):1360–1367, July 2001.
- Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Technical Report 653, Department of Statistics, UC Berkeley*, 2004.
- Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006.
- J. Theiler and K. Gloer. Sparse linear filters for detection and classification in hyperspectral imagery. In *Proceedings of the SPIE*, volume 6233, page CID:62330H, May 2006.
- S. Theodoridis and K. Koutroubas. *Pattern Recognition*. Academic Press, 2003.
- R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society B*, 58(1):267–288, 1996.
- M. Tipping. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research*, 1:211–244, 2001.
- T.-M. Tu. Unsupervised signature extraction and separation in hyperspectral images: A noise-adjusted fast independent components analysis approach. *Optical Engineering*, 39(4):897–906, 2000.

- T.-M. Tu, H.-C. Shyu, Y.-S. Sun, and C.-H. Lee. Determination of data dimensionality in hyperspectral imagery - pnapca. *Multidimensional Systems and Signal Processing*, 10(3): 255–273, July 1999.
- T.-M. Tu, P. S. Huang, and P.-Y. Chen. Blind separation of spectral signatures in hyperspectral imagery. In *Proceedings of the IEEE: Vision, Image and Signal Processing*, volume 148, pages 217–226, Aug. 2001.
- S. Venkataraman, L. M. Bruce, A. Cheriyyadat, and A. Mathur. Hyperspectral dimensionality reduction via localized discriminant bases. In *Proceedings of the IEEE: Geoscience and Remote Sensing Symposium*, pages 1245–1248, July 2005.
- D. D. Wackerly, W. Mendenhall, and R. L. Scheaffer. *Mathematical Statistics with Applications*. Wadsworth, 5 edition, 1996.
- J. Wang and C.-I. Chang. Applications of independent component analysis in endmember extraction and abundance quantification for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 44(9):2601–2616, September 2006.
- L. Wang, Y. Zhang, and Y. Gu. Unsupervised band selection method based on improved N-Findr algorithm for spectral unmixing. In *Proceedings of the 1st ISCAA*, pages 1018–1021, 2006.
- M. West, P. Muller, and M. D. Escobar. Hierarchical priors and mixture models with application in regression and density estimation. In P. R. Freeman and A. F. M. Smith, editors, *Aspects of Uncertainty*, pages 363–386. John Wiley, 1994.
- D. Whitley. An overview of evolutionary algorithms: Practical issues and common pitfalls. *Information and Software Technology*, 43(14):817–831, Dec. 2001.
- P. Williams. Bayesian regularization and pruning using a Laplace prior. *Neural Computation*, 7:117–143, 1995.
- E. M. Winter. Detection of surface mines using hyperspectral sensors. In *Proceedings of the IEEE: Geoscience and Remote Sensing Symposium*, pages 1597–1600, Anchorage, AK, September 2004.
- M. E. Winter. Fast autonomous spectral end-member determination in hyperspectral data. In *Proceedings of the Thirteenth International Conference on Applied Geologic Remote Sensing*, pages 337–344, Vancouver, B.C., Canada, 1999.
- H.-T. Wu, J.-F. Yang, and F.-K. Chen. Source number estimators using transformed Gerschgorin radii. *IEEE Transactions on Signal Processing*, 43(6):1325–1333, June 1995.
- E. P. Xing, M. I. Jordan, and R. Sharan. Bayesian haplotype inference via the Dirichlet process. *Journal of Computational Biology*, 14:267–284, 2007.

- X. Yu, S. Reed, and A. D. Stocker. Comparative performance analysis of adaptive multispectral detectors. *IEEE Transactions on Signal Processing*, 41(8):2639–2656, August 1993.
- A. Zare and P. Gader. Sparsity promoting iterated constrained endmember detection for hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 4(3):446–450, July 2007a.
- A. Zare and P. Gader. SPICE: A sparsity promoting iterated constrained endmember extraction algorithm with applications to landmine detection from hyperspectral imagery. In *Proceedings of the SPIE*, page CID:655319, Orlando, FL, 2007b.
- A. Zare and P. Gader. Hyperspectral band selection and endmember detection using sparsity promoting priors. *IEEE Geoscience and Remote Sensing Letters*, 5(2):256–260, Apr. 2008.
- A. Zare, J. Bolton, P. Gader, and M. Schatten. Vegetation mapping for landmine detection using long wave hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 46(1):172–178, January 2008.

BIOGRAPHICAL SKETCH

Alina Zare received her Bachelor of Science degree in computer engineering from the University of Florida in 2003. She continued her studies at the University of Florida to graduate with her Master of Science and Doctor of Philosophy degrees from the department of Computer and Information Science and Engineering in 2008. Her research interests include sparsity promotion, machine learning, Bayesian methods, image analysis and pattern recognition.