

ACCOUNTING FOR SPECTRAL VARIABILITY IN HYPERSPECTRAL  
UNMIXING USING BETA ENDMEMBER DISTRIBUTIONS

---

A Thesis  
presented to  
the Faculty of the Graduate School  
at the University of Missouri-Columbia

---

In Partial Fulfillment  
of the Requirements for the Degree  
Master of Science

---

by  
XIAOXIAO DU  
Dr. Alina Zare, Thesis Supervisor  
DECEMBER 2013

© Copyright by Xiaoxiao Du 2013

All Rights Reserved

The undersigned, appointed by the dean of the Graduate School, have examined the thesis entitled

**ACCOUNTING FOR SPECTRAL VARIABILITY IN HYPERSPECTRAL  
UNMIXING USING BETA ENDMEMBER DISTRIBUTIONS**

presented by Xiaoxiao Du,

a candidate for the degree of Master of Science,

and hereby certify that, in their opinion, it is worthy of acceptance.

---

Alina Zare, Ph.D.

---

Dominic Ho, Ph.D.

---

Dmitry Korkin, Ph.D.

.....Thanks, Mom and Dad.

## ACKNOWLEDGEMENTS

I would like to thank my advisor and committee chair, Dr. Alina Zare, for her wholehearted guidance and support throughout my studies and research. I would also like to thank my committee members, Dr. Dominic Ho and Dr. Dmitry Korkin, for all of their help and valuable suggestions.

Thank you to those who shed light on the field of hyperspectral unmixing and endmember estimation. Thank you to those who make the data sets used in this thesis available.

I am grateful for my labmates and friends Josephine Dula, Piyush Khopkar, Chao Chen, Changzhe Jiao, Matthew Cook and Kaylie Vogt for the times they provided insightful comments as well as encouragements in my studies and in life.

Finally, a warm thank you goes to my parents and all of my family, for their love, support, understanding and inspiration, as always.

## TABLE OF CONTENTS

ACKNOWLEDGEMENTS .....	ii
LIST OF FIGURES .....	vi
LIST OF TABLES .....	ix
LIST OF ABBREVIATIONS .....	x
ABSTRACT .....	xii
1. INTRODUCTION .....	1
1.1. Hyperspectral Imaging .....	1
1.2. Hyperspectral Unmixing and Endmember Estimation .....	3
1.2.1. Linear Mixing Model .....	3
1.2.2. Nonlinear Mixing Model .....	8
1.3. Research Goals .....	8
2. LITERATURE REVIEW .....	11
2.1. Hyperspectral Unmixing with Spectral Variability Overview .....	11
2.2. Endmember as Sets .....	11
2.2.1. Support Vector Machine Methods .....	12
2.2.2. Multiple Endmember Spectral Mixture Analysis Methods .....	16
2.2.3. Endmember Bundles Methods .....	20
2.2.4. Local Unmixing .....	24
2.3. Endmember as Statistical Distributions .....	25
2.3.1. Gaussian Distribution .....	25
3. HYPERSPECTRAL UNMIXING USING QUADRATIC PROGRAMMING WITH GAUSSIAN DISTRIBUTIONS .....	31
3.1. Model Description .....	31

4. HYPERSPECTRAL UNMIXING USING SAMPLING METHOD WITH GAUSSIAN DISTRIBUTIONS .....	34
4.1. Model Description .....	34
4.2. Metropolis-Hastings (MH) Sampling Approach .....	35
5. HYPERSPECTRAL UNMIXING USING QUADRATIC PROGRAMMING METHOD WITH BETA DISTRIBUTIONS .....	38
5.1. Model Description .....	38
5.2. Quadratic Programming Unmixing with Beta Distribution.....	41
6. HYPERSPECTRAL UNMIXING USING SAMPLING METHOD WITH BETA DISTRIBUTION.....	44
6.1. Model Description .....	44
6.2. Sampling Unmixing Method with Beta Distribution.....	44
7. HYPERSPECTRAL UNMIXING BASED ON FLICM WITH BETA DISTRIBUTION.....	47
7.1. Model Description .....	47
7.2. Quadratic Programming and MH Sampling Unmixing Approaches with Beta Distribution .....	50
8. HYPERSPECTRAL UNMIXING BASED ON superpixels WITH BETA DISTRIBUTION.....	52
8.1. Superpixels.....	52
8.2. Graph-Based Superpixel Segmentation Model.....	52
8.3. Entropy Rate Superpixel Model.....	55
8.4. QP and MH Sampling Approaches with Beta Distribution .....	58

9. HYPERSPECTRAL UNMIXING BASED ON SPATIAL K-MEANS WITH BETA DISTRIBUTION.....	59
9.1. Model Description .....	59
10. DATA SETS AND RESULTS .....	60
10.1. Data Sets .....	60
10.1.1. Toy Data.....	60
10.1.2. Simulated Hyperspectral Image Dataset 1 .....	63
10.1.3. Simulated Hyperspectral Image Dataset 2 .....	65
10.1.4. Real Hyperspectral Image Dataset 1 .....	67
10.1.5. Real Hyperspectral Image Dataset 2.....	68
10.2. Results.....	69
10.2.1. Toy data .....	69
10.2.2. Simulated HSI Dataset 1 .....	71
10.2.3. Simulated HSI Dataset 2 .....	71
10.2.4. PaviaU Hyperspectral Image Data.....	76
10.2.5. Gulfport Hyperspectral Image Data.....	90
10.3. Varying Parameters in Unmixing Methods .....	96
10.3.1. Relationship between $N$ and Proportion Errors .....	96
10.3.2. Relationship between Endmember Covariance and Proportion Errors ....	99
10.3.3. Relationship between Noise Covariance and Proportion Errors.....	101
10.3.4. Relationship between Cluster Number $K$ and Proportion Errors.....	103
10.3.5. Exploration of band selection in Superpixel 2 method.....	104
10.4. Computational Complexity.....	106
11. CONCLUSIONS AND FUTURE WORK .....	107
11.1. Conclusions.....	107
11.2. Future Work .....	108
BIBLIOGRAPHY .....	110
VITA .....	121

## LIST OF FIGURES

Figure	Page
Figure 1-1 Hyperspectral imaging 3D data cube .....	2
Figure 1-2 Linear mixing model surface illustration with three endmembers.....	4
Figure 1-3 k-simplex projection plots.....	6
Figure 1-4 2-simplex (triangle).....	6
Figure 10-1 Toy data plots for two-dimensional, three-endmember Gaussian toy data and Beta toy data. ....	62
Figure 10-2 Endmember plot of reflectance versus wavelength from ASTER spectral library.....	63
Figure 10-3 Beta distribution plot when alpha is less than 1.....	64
Figure 10-4 Skewed Beta distribution plot.....	65
Figure 10-5 Four Endmembers Spectra measured using a handheld ASD spectrometer for Simulated Dataset 2. ....	65
Figure 10-6 RGB image of simulated data with low noise and heavy noise.....	66
Figure 10-7 PaviaU hyperspectral image data ground truth and RGB image. ....	67
Figure 10-8 Gulfport hyperspectral image data RGB image. ....	68
Figure 10-9 Clustering results of simulated dataset 2 by $K$ -means, Fuzzy C-means, FLICM, Superpixel 1, Superpixel 2, and Spatial- $K$ -means. ....	72
Figure 10-10 Unmixing results of Simulated data set 2 with low noise.....	73
Figure 10-11 Unmixing results of Simulated data set 2 with high noise.....	74
Figure 10-12 PaviaU hyperspectral image data endmember demonstration..	77

Figure 10-13 Illustration for substituting zero data for KL-Divergence method.....	79
Figure 10-14 Ground Truth of PaviaU data subset 1.....	81
Figure 10-15 Proportion map of shadows, bitumen, bare soil and trees unmixing method on PaviaU data subset .....	83
Figure 10-16 Ground Truth of PaviaU data subset 2.....	84
Figure 10-17 Proportion map of shadows, meadows, painted metal sheets, self-blocking blocks and background unmixing method on PaviaU data subset.....	87
Figure 10-18 Ground Truth of PaviaU data subset 3.....	88
Figure 10-19 Proportion map of trees, asphalt and background unmixing method on PaviaU data subset .....	89
Figure 10-20 Gulfport hyperspectral image data endmember demonstration. ....	91
Figure 10-21 Gulfport hyperspectral image data endmember q-q plot with endmember Grass at 368nm. ....	92
Figure 10-22 Gulfport hyperspectral image data cluster result of spatial K-means .....	92
Figure 10-23 Gulfport hyperspectral image data cluster result of superpixel 2 .....	93
Figure 10-24 Unmixing results of Gulfport data. ....	94
Figure 10-25 Proportion error versus number of data points $N$ of Gaussian toy data....	97
Figure 10-26 Proportion error versus number of data points $N$ of Beta toy data .....	98
Figure 10-27 Proportion error versus Endmember covariance on Gaussian toy data ...	100
Figure 10-28 Proportion error versus Endmember covariance on Beta toy data.....	101
Figure 10-29 Proportion error versus noise covariance on Gaussian toy data. ....	102

Figure 10-30 Proportion error versus noise covariance on Beta toy data..... 103

Figure 10-31 Proportion error versus cluster number  $K$  using BCM-Spatial- $K$ -means  
unmixing on Simulated data set 2..... 104

## LIST OF TABLES

Table	Page
Table 1 Beta toydata endmember parameter settings .....	61
Table 2 Groundtruth classes for the PaviaU scene .....	68
Table 3 <i>PError</i> when unmixing Toydata .....	70
Table 4 <i>PError</i> when unmixing Simulated HSI Dataset 1 .....	71
Table 5 <i>PError</i> when unmixing Simulated HSI Dataset 2 (Q: Quadratic programming, S: Sampling).....	75
Table 6 PaviaU hyperspectral image data endmember approximation .....	80
Table 7 Proportion Error and Band Selection.....	105

## LIST OF ABBREVIATIONS

AVIRIS	Airborne Visible/Infrared Imaging Spectrometer
BCM	Beta Compositional Model
BSMA	Bayesian Spectral Mixture Analysis
CCA	Convex Cone Analysis
CGAL	Computational Geometry Algorithms Library
<b>E</b>	Matrix of endmembers where the $i^{th}$ column contains the spectral signature for the $i^{th}$ endmember, $\mathbf{e}_i$
FLICM	Fuzzy Local Information C-Means algorithm
HSI	Hyperspectral Imaging
HU	Hyperspectral Unmixing
ICA	Independent Component Analysis
ICE	Iterative Constrained Endmembers
MAP	Maximum A Posteriori
MCMC	Markov Chain Monte Carlo
MESMA	Multiple Endmember Spectral Mixture Analysis
MH	Metropolis-Hastings
MNF	Maximum Noise Fraction Transform
MV	Minimum Volume
MVC-NMF	Minimum Volume Transform-Nonnegative Matrix Factorization
NCM	Normal Compositional Model

<b>P</b>	Matrix of proportion (abundance) values where each element, $p_{ik}$ , is the abundances of endmember $k$ in pixel $i$
PCE	Piece-wise Convex Endmember detection algorithm
PP	Pure Pixel
PPI	Pure Pixel Index algorithm
QP	Quadratic Programming
SMA	Spectral Mixture Analysis
SPICE	Sparsity-Promoting Iterative Constrained Endmembers
SVM	Support Vector Machine
VCA	Vertex Component Analysis

## ABSTRACT

Hyperspectral imaging is widely used in the field of remote sensing (Goetz, et al., 1985; Green, et al., 1998). In a hyperspectral imaging system, sensors collect radiance/reflectance values over an area (or a scene) across hundreds of spectral bands (Goetz, et al., 1985). The hyperspectral image yielded by such system can be represented by a three-dimensional data cube containing those radiance/reflectance values in a range of wavelengths (Landgrebe, 2002). There are two common analysis methods for hyperspectral imagery (Hu, et al., 1999): endmember estimation and hyperspectral unmixing. Endmember estimation aims at finding pure individual spectral signatures of the materials (endmembers) in the scene (Adams, et al., 1986). Hyperspectral unmixing, on the other hand, estimates the proportions of each endmember at every pixel of the image. Each pixel in the image can then be represented by endmember spectra weighted by its corresponding proportions.

In order to increase the accuracy of hyperspectral unmixing, sufficient temporal and spatial spectral variability of endmembers must be taken into consideration (Roberts, et al., 1992; Roberts, et al., 1998; Bateson, et al., 2000). The most common factors contributing to spectral variability include environmental factors, such as atmospheric effects, illumination, moisture conditions, and inherent spectral variability of the material itself, such as the variations in biophysical and biochemical composition in vegetation (Song, 2005). Under such influence, the spectral signature of endmembers may vary from time to time and from pixel to pixel in the scene.

In order to account for such endmember spectral variability, endmembers are regarded as either a set, or a “bundle”, of individual spectra (Roberts, et al., 1998; Bateson, et al., 2000), or as a sample from a full distribution. The application of the

Normal Compositional Model with Gaussian-distributed endmembers has been discussed in the literature (Eches, et al., 2010; Zare, et al., 2012). Since the domain of Gaussian distribution is  $(-\infty, \infty)$ , Gaussian endmembers allow samples outside the interval of  $[0, 1]$ . However, in reality, the reflectance value of endmember spectra usually only vary between zero and one. Beta distributions, on the other hand, are defined only over the interval of  $[0, 1]$ . Therefore, in this thesis, the Beta distribution is considered for endmembers in order to make it more physically realistic. The Beta Compositional Model (Zare, et al., 2013) is considered as the mixing model in this case.

Two approaches based on the Normal Compositional Model (Stein, 2003; Eismann, 2006) and the Beta Compositional Model, quadratic programming (QP) approach and Metropolis-Hastings (MH) sampling approach, are presented in this thesis for hyperspectral unmixing, i.e., finding the proportions of each endmember in a hyperspectral image. QP approach determines the proportion values by minimizing the difference between the mean of Beta approximation to the convex combination of Beta endmember distributions, while MH sampling method takes both the mean and variance into consideration.

Furthermore, in this thesis, unmixing algorithms that incorporate spatial information are proposed under the Beta Compositional Model (BCM-Spatial algorithms). These include algorithms based on Fuzzy Local Information C-Means Clustering Algorithm (FLICM), superpixel methods, and spatial K-means algorithm.

Results indicate that unmixing algorithms based on NCM and BCM are able to successfully perform unmixing on simulated data and real hyperspectral data and can incorporate endmember spectral variability. BCM unmixing does a better job than NCM unmixing on data generated from Beta endmembers than those from Gaussian

endmembers. The results from BCM-Spatial unmixing algorithms on hyperspectral image data show that the new algorithms are effective at unmixing.

## 1. INTRODUCTION

### 1.1. Hyperspectral Imaging

Spectral imaging refers to the process of collecting and processing spectral information over a range of the electromagnetic spectrum. Hyperspectral imagers (HSI) image a scene over a large number of contiguous, narrow spectral wavelengths and measure the spectral response of all pixels in a scene.

In HSI, hyperspectral sensors collect radiance values from an area (a scene), and such values will be stored and presented in the form of a data cube (Keshava, et al., 2002). This cube can be seen as a set of images where each image represents radiance or reflectance information over the scene in a certain wavelength. These images are then organized into planes and form a three-dimensional hyperspectral data cube for processing and analysis. Each element of the data cube represents the radiance/reflectance values measured in one particular wavelength at one pixel location. Each spectral vector corresponds to the radiance acquired at a given pixel across all spectral bands. Figure 1-1 illustrates the concept of hyperspectral imaging.

Hyperspectral images can provide information about the ground of an area than a multispectral pixel spectrum. There are many applications that can take advantage of hyperspectral remote sensing, including agriculture forecasting (Corp, et al., 2010), environmental regulation (Krezhova, et al., 2011; Wan, et al., 2011), urban planning (Heiden, et al., 2003; Guglietta, et al., 2007; Gamba, 2013) and geology and medical image analysis (Martin, et al., 2006; Akbari, et al., 2012).

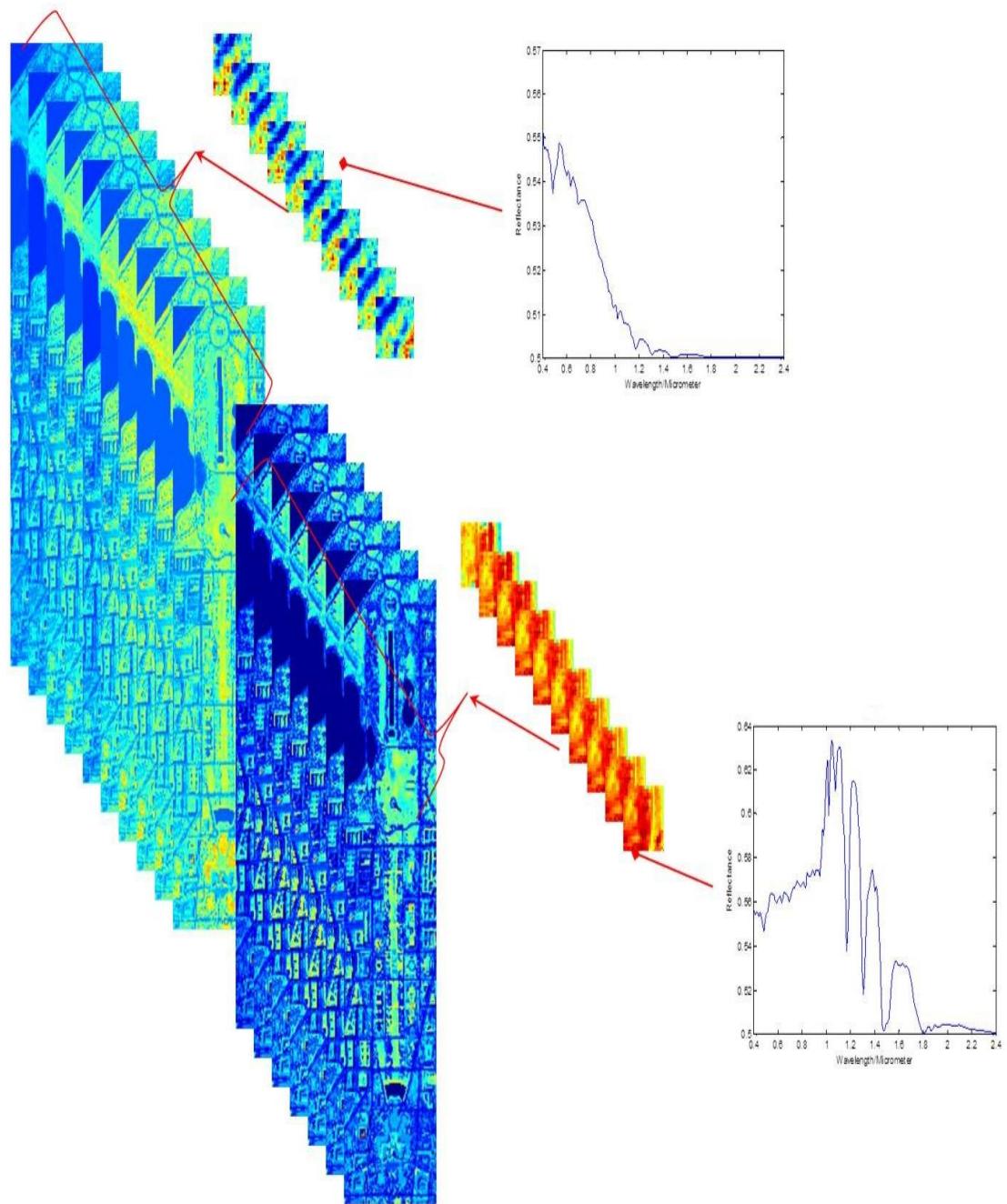


Figure 1-1 Hyperspectral imaging 3D data cube.  
Image data come from AVIRIS Washington DC Mall imagery (Landgrebe, et al.).

## 1.2. Hyperspectral Unmixing and Endmember Estimation

The analysis of hyperspectral imagery often consists of the following two problems (Hu, et al., 1999): endmember estimation and hyperspectral unmixing. Endmember estimation aims at finding the pure individual spectral signatures of materials (endmembers) in the scene (Adams, et al., 1986). It is assumed that each of the materials has relatively constant, yet distinct spectral characteristics (Keshava, et al., 2002). The percentage of the materials presented in each pixel is called a proportion (Bioucas-Dias, et al., 2012). Under such assumption, each and every spectral pixel in the image can then be represented by endmember spectra weighted by its corresponding proportions. Hyperspectral unmixing, thus, refers to the process of estimating the proportions of each endmember at every pixel of the image.

Hyperspectral unmixing algorithms depend on a mixing model. Mixing models can be categorized into two categories (Keshava, et al., 2002; Bioucas-Dias, et al., 2012): linear mixing and non-linear mixing models. The linear mixing model is the most widely used.

### 1.2.1. Linear Mixing Model

The linear mixing model applies when the scale of each material in a pixel is macroscopic (Singer, et al., 1979) and the incident light only interacts with one material, as shown in the “checkerboard” mixture situation in Figure 1-2. If the scattering of light between different materials can be ignored (Bioucas-Dias, et al., 2012), then the reflectance value collected by the sensor is the linear combination of spectral signature of each endmember with the same proportion. That is to say, the spectral value of the pixels

can be represented with a convex combination of endmembers (the pure spectra for each material) weighted by its corresponding proportions.

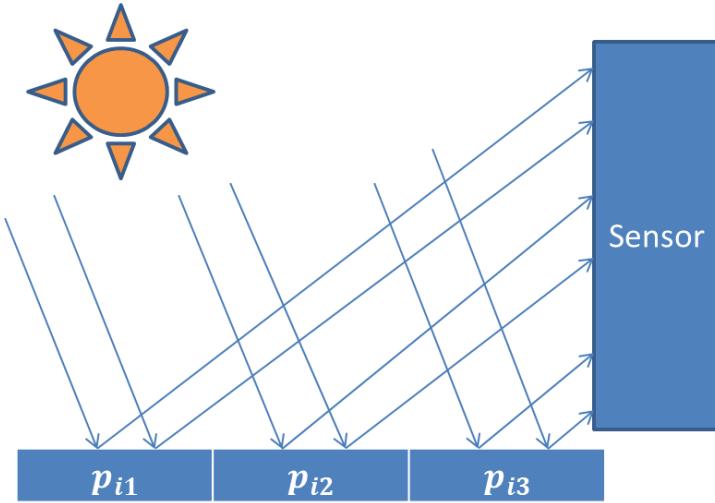


Figure 1-2 Linear mixing model surface illustration with three endmembers.

The model can be written as (Keshava, 2003):

$$\mathbf{x}_i = \sum_{j=1}^M p_{ij} \mathbf{e}_j + \boldsymbol{\varepsilon}_i, j = 1, \dots M \quad 1$$

where  $\mathbf{x}_i$  is the spectral measurement (actual image data point spectrum value) at pixel  $i$ ,  $M$  is the total number of endmembers,  $\mathbf{e}_j$  is the  $j^{th}$  endmember (which is a  $1 \times D$  vector,  $D$  is the total number of spectral bands),  $p_{ij}$  is the proportion (scalar) of endmember  $j$  at pixel  $i$ . The proportions of this model satisfy the following constraints:

$$p_{ij} \geq 0, j = 1, \dots M$$

$$\sum_{j=1}^M p_{ij} = 1 \quad 2$$

Note that  $\varepsilon_i$  here is an error term for the  $i^{th}$  pixel, including any measurement errors, environmental disturbance and modelling errors.

Linear mixing model assumes that all pixels in a scene are included in one convex region. If the model and both constraints on the proportions are satisfied, the data point spectrum is a convex combination of endmembers and the endmembers will be found at the corners of the convex region (Keshava, et al., 2002; Gillis, et al., 2008). Bioucas-Dias, et al. (Bioucas-Dias, et al., 2012) reviews four different categories of approaches for hyperspectral endmember extraction and unmixing under the linear mixing model: geometrical, statistical, sparse regression-based and spatial-spectral contextual information-based approaches.

Geometrical-based approaches can be further categorized into Pure Pixel (PP)-based algorithms and Minimum Volume (MV)-based algorithms. Pure Pixel-based algorithms assume that, in the input data set, there exists at least one pure pixel for each of the endmembers. This assumption generally results in computationally efficient methods (Bioucas-Dias, et al., 2012), yet when pure pixels cannot be found in the scene for each endmember, it can result in inaccurate endmembers. Furthermore, it requires the hyperspectral imaging device to perform at such a spatial resolution that the pure endmember spectra and the spectra of neighboring materials do not combine. Representative algorithms based on the Pure Pixel assumption include: Pixel Purity Index (PPI) algorithm (Boardman, 1994; Chaudhry, et al., 2006), N-FINDR algorithm (Winter, 1999) and Vertex Component Analysis (VCA) algorithm (Nascimento, et al., 2005).

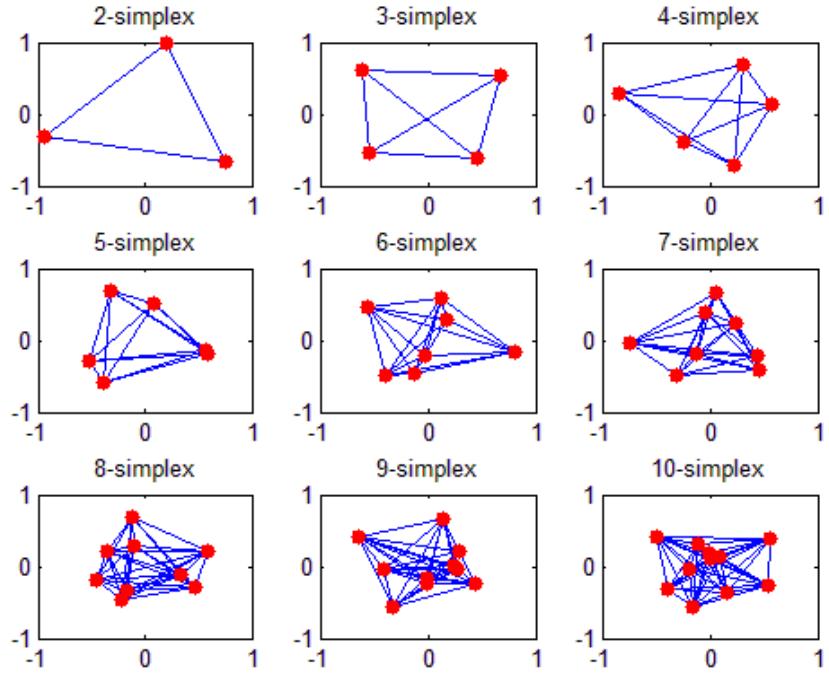


Figure 1-3 k-simplex projection plots, k takes value from 2 to 10.

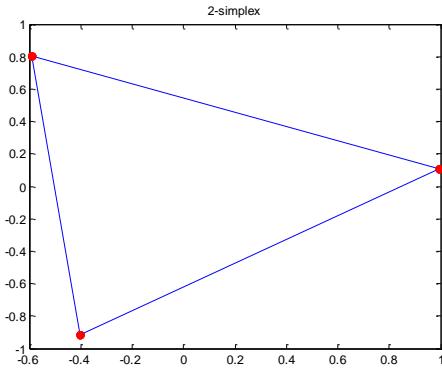


Figure 1-4 2-simplex (triangle)

On the other hand, if the endmembers are treated as vertices, a simplex can be constructed based on the endmembers. Minimum Volume based approaches try to find a mixing matrix  $\mathbf{M}$  which minimizes the volume of such simplex defined by endmembers, where  $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \dots, \mathbf{m}_D]$ ,  $D$  is the number of dimensions of the data set and  $\mathbf{m}_i$  is a  $1 \times D$  vector, representing the  $i^{th}$  endmember. A  $k$ -simplex is defined as a  $k$ -

dimensional polytope which is the convex hull (the smallest convex set that contains the  $X$  data points) of its  $k + 1$  vertices. Figure 1-3 illustrates the projection on a two-dimensional plane of a series of simplexes, including a 2-simplex (triangle), a 3-simplex (a tetrahedron), a 4-simplex (a 5-cell), etc. and a 10-simplex (a hendecaxennon). Figure 1-4 illustrates the true plot of a 2-simplex.

MV-based algorithms are optimization problems that do not assume pure pixel constraints (Bioucas-Dias, et al., 2012). However, MV-based algorithms are subject to the constraint that the simplex contains all the observed image data points. The purpose of minimizing the volume of the convex region is to constrain the number of possible solutions of endmember combinations. That is to say, any simplex that contains the image data points can make the error between the original data set and the convex combination of endmembers weighted by the estimated proportion values zero, but the vertices (endmembers) that can yield the minimum volume assures that the endmember combination is the one and only result. Representative algorithms include Minimum Volume Transform-Nonnegative Matrix Factorization (MVC-NMF) (Miao, et al., 2007), Iterative Constrained Endmembers (ICE) algorithm (Berman, et al., 2004), Sparsity-Promoting Iterative Constrained Endmembers (SPICE) algorithm (Zare, et al., 2007), and Convex Cone Analysis (CCA) algorithm (Ifarraguerri, et al., 1999).

Statistical methods turn the hyperspectral unmixing problem into a statistical inference problem and these methods were developed to deal with highly mixed spectral mixtures in the image. However, they are usually computationally complex (Bioucas-Dias, et al., 2012). Independent Component Analysis (ICA) (Comon, 1994; Nascimento, et al., 2005) is a well-known algorithm that has been used for unmixing hyperspectral image data without the knowledge of the number of endmembers or their reflectance values. Bayesian unmixing algorithms have been used, such as the joint maximum a

posteriori (MAP) estimator (Berman, et al., 2004; Miao, et al., 2007; Zare, et al., 2007), joint Bayesian estimator (Dobigeon, et al., 2009), piecewise unmixing using Normal Compositional Model (Zare, et al., 2010), etc.

Sparse regression based unmixing finds the best subset of endmember spectral information from a given spectral library and, then, use the subset to estimate endmembers and proportions for each pixel of the image. Sparse regression methods can be seen in (Donoho, et al., 2003; Bioucas-Dias, et al., 2010; Iordache, et al., 2011).

### 1.2.2. Nonlinear Mixing Model

In a more complicated situation, different materials can scatter radiation and the sensor picks up a mixed reflectance resulting from the physical interactions between the radiation off of many materials. The interaction can be multilayered, which means the radiation light will be reflected from one or more materials, bouncing to another one or more materials, and then reach the sensor. Alternatively, there can be intimate interactions, which means the surface is homogenously mixed (Bioucas-Dias, et al., 2012) and the interactions of light occur at a molecular level. In this case, nonlinear mixing models need to be applied. However, as our research are devoted to linear mixing model, linear mixing model and algorithms based on linear mixing model are discussed in this thesis.

### 1.3. Research Goals

Spectral information can be affected by factors including environmental and atmospheric effects, illumination, moisture conditions, and inherent spectral variability of

the material itself, such as the variations in biophysical and biochemical composition in vegetation (Song, 2005). Due to spectral variability, the measured spectral information can vary for an endmember. To account for such spectral variability, each endmember is either modeled as a set, or a “bundle”, of individual spectra (Roberts, et al., 1998; Bateson, et al., 2000) and proportion values are estimated through optimization approaches (Zare, et al., 2012; Dobigeon, et al., 2009). The literature has shown the application of the Gaussian distribution to represent endmembers. The relevant literature has shown that putting a distribution on endmembers can successfully reflect spectral variability. However, the Gaussian distribution can find endmembers with reflectance value between minus infinity and infinity, yet the reflectance values of the spectra of endmembers usually vary between zero and one. In this thesis, the Beta distribution is considered to represent the endmembers in order to make it more physically realistic as Beta distribution is defined on the interval between zero and one.

Recent work in the literature has introduced the Beta Compositional Model (Zare, et al., 2013). In this work, the parameters of a Beta distribution are estimated using the  $K$ -nearest spectral neighbors of a pixel in order to approximate the pixel’s distribution. This method assumes that the  $K$ -nearest spectral neighbors of a pixel can best represent the pixel. However, in reality, it is likely that materials that are close to each other spatially rather than spectrally correspond to the same composition of materials, thus the same proportion structures. Therefore, as an improvement to existing Beta Compositional Model approaches, new Beta unmixing algorithms incorporating spatial information are proposed in the thesis. These new methods rely on Fuzzy Local Information C-Means Clustering Algorithm (FLICM), superpixel and spatial  $K$ -means algorithms.

Experiments comparing quadratic programming (QP) approach and Metropolis-Hastings (MH) sampling approach for hyperspectral unmixing are shown and discussed

in the thesis. Toy data, simulated hyperspectral data, and real hyperspectral image data are used to conduct hyperspectral unmixing experiments.

## 2. LITERATURE REVIEW

### 2.1. Hyperspectral Unmixing with Spectral Variability Overview

Spectral variability is the variation in spectral information of a material (Chang, 2000). The most common factors contributing to spectral variability include environmental factors, such as atmospheric effects, illumination, moisture conditions, and inherent spectral variability of the material itself, such as the variations in biophysical and biochemical composition in vegetation (Song, 2005; Cho, et al., 2009). Under such effects, the spectral signature of endmembers may vary from time to time and from pixel to pixel in the scene.

It is, thus, reasonable to take spectral variability into consideration during the hyperspectral unmixing process. Given spectral variability, one single spectral signature is not adequate to fully represent each endmember. There are two dominant categories of endmember representation in hyperspectral unmixing with spectral variability: endmember as sets and endmember as statistical distributions (Zare, et al., In Press). The rest of this chapter will be devoted to discuss the existing algorithms that incorporate endmember spectral variability for each endmember while performing hyperspectral unmixing.

### 2.2. Endmember as Sets

In this representation, endmembers are regarded as a set, or a “bundle”, of individual spectra. A pixel from an input hyperspectral image is represented as the convex combination of one or more endmember representation selected from the set of spectra. Proportion values are generally estimated through minimizing objective

functions. Prominent algorithms include methods using the support vector machine (SVM), multiple endmember spectral mixture analysis, endmember bundles and local unmixing methods.

### 2.2.1. Support Vector Machine Methods

Support vector machine methods such as (Brown, et al., 1999; Rabe, et al., 2009; Bovolo, et al., 2010) model mixed pixels as mixed regions in a support vector machine classification problem and use pure pixels as endmembers. Representative algorithms are discussed in the following.

**Extended support vector machines (ESVM).** The ESVM algorithm (Wang, et al., 2009; Jia, et al., 2010) models an endmember using a group of spectra to incorporate endmember spectral variation. The unmixing process of ESVM is transformed to a classification problem in SVM (Bishop, 2006) where each class of the SVM is treated as an endmember. Suppose a two-class classification problem (two endmembers). The ESVM classifies data by finding the optimal hyperplane that separates all data points of one class from those of the other class. The desired hyperplane has the largest margin between the two classes. Here, the set of relative pure pixels of each endmember is regarded as each “class”. The samples between the margins are regarded as mixed pixels and the samples falling on the boundary between pure and mixed pixels (support vectors) are “just pure pixels”. The samples outside the margin are relative pure pixels with full membership of that endmember, and variation within the relative pure pixels accommodates variations of the same endmember.

The hyperplane  $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$  is derived by maximizing the margin  $2/\|\mathbf{w}\|$  between the two hyperplanes.  $\mathbf{w}$  is the weight vector. Once  $f(\mathbf{x})$  is known, the proportions of endmembers are found by the following decision rules:

$$\text{If } f(\mathbf{x}) \geq 1, \mathbf{x} \in \omega_A, \text{i.e. } p_{\mathbf{x}}(\omega_A) = 1, p_{\mathbf{x}}(\omega_B) = 0;$$

3

$$\text{If } f(\mathbf{x}) \leq -1, \mathbf{x} \in \omega_B, \text{i.e. } p_{\mathbf{x}}(\omega_A) = 0, p_{\mathbf{x}}(\omega_B) = 1;$$

$$\text{If } -1 \leq f(\mathbf{x}) \leq 1, \mathbf{x} \in \{\omega_A, \omega_B\}, \text{i.e. } p_{\mathbf{x}}(\omega_A) = \frac{1}{2}(f(\mathbf{x}) + 1), p_{\mathbf{x}}(\omega_B) = \frac{1}{2}(1 - f(\mathbf{x})).$$

where  $\omega_A, \omega_B$  are classes (endmembers) A and B,  $p_{\mathbf{x}}(\omega_A)$  and  $p_{\mathbf{x}}(\omega_B)$  are proportions of endmembers A and B.

**Improved extended support vector machines (Improved-ESVM).** The Improved-ESVM algorithm (Li, et al., 2012) improves on ESVM in that it allows multiple endmembers to unmix the mixed pixel and instead of directly assigning proportions to 0 or 1 if  $f(\mathbf{x}) \geq 1$  or  $f(\mathbf{x}) \leq -1$ , Improved-ESVM finds proportion values by minimizing the residual.

Both the ESVM and the Improved-ESVM algorithms take endmember variability into account by modeling an endmember using a group of spectra. ESVM performs better when the input image has pure pixels (the proportion values are 0 and 1), while Improved-ESVM works better on mixed pixels. A significant drawback of the Improved-ESVM algorithm is the occurrence of overlapping problems if the number and variability of the endmembers increase: model overlaps (the same mixed pixel can be unmixed in different combination of proportions) (Youngentob, et al., 2011), and fraction overlaps (different mixed pixels can be unmixed by the same proportion).

**Pairwise Coupling Support Vector Machines (SVM-PWC) algorithm.** SVM-PWC algorithm (Li, et al., 2011) uses SVM to solve the classification problem and

extends with pairwise coupling to find proportion values. Here, the set of relative pure pixels of each endmember is regarded as each “class” of the data. SVM-PWC first finds boundaries between classes of data in a way similar to SVM (Bishop, 2006).  $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$  is the hyperplane derived from the SVM algorithm,  $f = 1$  is the positive margin and  $f = -1$  is the negative margin (Platt, 1999).  $y_i = k$  means the  $i^{th}$  data point belongs to  $k^{th}$  class. Then, the proportion values are estimated by combining pairwise class probabilities as described in (Wu, et al., 2004). Given  $K$  classes of data, for any  $x$ , the posterior probability of  $x$  being in class  $k$  can be written as

$$p_k = p(y = k|x), k = 1, \dots, K \quad 4$$

The pairwise class probabilities  $r_{kl}$  between class  $k$  and class  $l$  are estimated as follows:

$$r_{kl} \approx p(y = k|y = k \text{ or } l, x), k, l = 1, \dots, K \quad 5$$

According to (Platt, 1999),

$$r_{kl} \approx \frac{1}{1 + e^{Af+B}}. \quad 6$$

Given a known training data set  $(f_i, y_i)$ , where  $i$  is the  $i^{th}$  data in the training data set, the posterior probability  $p_i$  is defined as

$$p_i = \frac{1}{1 + e^{Af_i+B}}, \quad 7$$

and the target probability  $t_i$  is defined as:

$$t_i = \frac{y_i + 1}{2}. \quad 8$$

Therefore, the parameters  $A$  and  $B$  from (6) are estimated by minimizing the negative log likelihood of the training data given by

$$\min - \sum_i t_i \log p_i + (1 - t_i) \log(1 - p_i). \quad 9$$

After finding parameters  $A$  and  $B$ , the PWC method finds proportion values  $p_k$  for class  $k$  by solving the following optimization problem (Wu, et al., 2004):

$$\begin{aligned} & \min_p \frac{1}{2} \sum_{k=1}^K \sum_{l:l \neq k} (r_{lk} p_k - r_{kl} p_l)^2 \\ & \text{subject to } \sum_{k=1}^K p_k = 1, p_k \geq 0 \quad \forall k \end{aligned} \quad 10$$

Thus, the proportions found by SVM-PWC method satisfy both the sum-to-one and non-negativity constraints.

**Unmixing-to-Classification Conversion Model.** Mianji and Zhang proposed an unmixing-to-classification conversion model (Mianji, et al., 2011) that treats unmixing problem as an SVM classification problem. In this model, proportions from 0 to 1 are assigned to a set of class labels. The number of classes  $n_c$  is determined based on resolution  $r\%$  as follows:

$$n_c = \left\| \left( \frac{100}{r} - 1 \right) \right\| \quad 11$$

The endmember sets are known from a spectral library. Therefore, after pre-assigning different proportion values for each endmember, artificial data is generated by combining the endmember elements selected from the endmember sets with the assigned proportion values. SVM is used on the artificial data to classify and quantify proportions. Note that there are  $(n_c + 2)$  SVM classifiers in total for all proportion classes. In testing, a pixel is classified by all SVM classifiers and whichever classifier produces the largest margin, the pixel is classified to belong to the proportion class with the same label as the SVM classifier.

### 2.2.2. Multiple Endmember Spectral Mixture Analysis Methods

**Bayesian Spectral Mixture Analysis (BSMA).** The BSMA model (Song, 2005) is based on Bayes Theorem. BSMA considers intra-class endmember spectral variability by using all possible combinations of elements from the endmember set in the unmixing process.

In BSMA, the prior probability density function  $\pi_c$  for each endmember set is known from the given spectral library. If there is no prior information on a pixel,  $\pi_c$  should be a uniform distribution from 0 to 1. Suppose BSMA with three endmembers and two independent spectral measurements  $R_1$  and  $R_2$  for each of the endmember, the linear mixture can be written as

$$\begin{aligned} c_1 R_{11} + c_2 R_{12} + (1 - c_1 - c_2) R_{13} &= R_1 \\ c_2 R_{21} + c_2 R_{22} + (1 - c_1 - c_2) R_{23} &= R_2 \end{aligned} \quad 12$$

where  $R_{ij}$  is the spectral signature of  $j^{th}$  endmember for spectral measurement  $i$ .  $c_i$  is the proportion values. Also, the likelihood function,  $f(R_1|c_i)$  and  $f(R_2|c_i)$  can be obtained using the convolution

$$\begin{aligned} f(R_1|c) &= f(R_{11}|c) * f(R_{12}|c) * f(R_{13}|c) \\ f(R_2|c) &= f(R_{21}|c) * f(R_{22}|c) * f(R_{23}|c) \end{aligned} \quad 13$$

The different combination of endmembers is considered by using a recursive equation that calculates the proportion value  $p(c_i|R_1, R_2)$  given two independent spectral measurements  $R_1$  and  $R_2$  in each endmember set. The equation is written as follows,

$$p(c_i|R_1, R_2) = \frac{f(R_2|c_i) g(c_i|R_1)}{\int_0^1 f(R_2|u) g(u|R_1) du} \quad 14$$

where  $g(c_i|R_1) = \frac{f(R_1|c_i) h(c_i)}{\int_0^1 f(R_1|u) h(u) du}$

If there are more than two spectral measurements, i.e. elements, in an endmember set, the above equation will be used recursively to find proportions.

The drawback of BSMA is that the recursive equation only allows for two measurements within an endmember set during one calculation cycle. Hyperspectral remote sensing image data often contains large amounts of information (Song, 2005) and there are usually more than two independent spectral measurements available for an endmember, BSMA has a high computing expense by having to do the equation recursively. Moreover, this study limits the number of endmembers to two and three only. It is also reported that the traditional SMA based on the means the endmember signature distributions is able to yield the same or even higher accuracy than the BSMA with the same endmember signature distributions (Song, 2005).

**MESMA** (Roberts, et al., 1998). Multiple Endmember Spectral Mixture Analysis (MESMA) algorithm was a new technique to map hyperspectral image as linear combination of two or more endmembers. It constructs a library of candidate endmember “models” from a mixture of shade and an endmember in the scene. The proportion values are estimated by selecting zero or one element from the endmember set, and the proportion values should satisfy three conditions:

- 1) The proportions found are between -0.01 and 0.01 (1% permitted error);
- 2) The root mean square (RMS) error between the true pixel value and the convex combination of the selected endmember sand proportions should be lower than a threshold,  $\epsilon$ ,

$$RMS(\mathbf{x}_i, \mathbf{p} \cdot \mathbf{e}) = \sqrt{\frac{1}{D} \|\mathbf{x}_i - \sum_{m=1}^M p_{mi} \mathbf{e}_m\|_2^2} < \epsilon \quad 15$$

where  $\mathbf{x}_i$  is the  $i^{th}$  pixel,  $D$  is the number of bands,  $M$  is the number of endmembers.

- 3) Any individual residual, i.e. the RMS error on each band, is below a threshold,  $\epsilon'$ .

The objective function of MESMA can be written as

$$Z = \sum_I a_i Y_i , \quad 16$$

with constraints

$$\sum_j a_{ij} Y_j \geq 1 \text{ for each } i \in I \quad 17$$

$$\sum_j X_j = N_p \quad 18$$

where  $a_i$  is the number of pixels represented as element  $I$ , initially set to 1.  $Y_i$  is 1 if pixel  $I$  cannot be classified by the selected set of models, 0 otherwise.  $X_j$  is 1 if model  $j$  is chosen, 0 if not.  $N_p$  is the number of models to be chosen.  $a_{ij}$  is 1 if pixel  $i$  can be classified by model  $j$ , 0 otherwise.

The pseudo code for MESMA method is as follows:

---

**Algorithm 1 : MESMA-Original (Roberts, et al., 1998)**

---

```

1 Input: endmember candidates  $E_L$  from endmember library; number of endmember
2 candidates  $N_E$ ; input image data  $x$ ; number of pixels  $N$ ; number of bands  $\lambda$ ;
3 Output: proportion values  $p$ 
4 Initialize endmember library  $\leftarrow$  laboratory measured spectra data
Selection of Endmember candidate models
5 Initialize model matrix  $Model \leftarrow zeros(N \times N_E)$ 
6 for  $i \leftarrow 1$  to  $N$  do
7   Select endmember candidates  $E_L$  from endmember library
8   for  $j \leftarrow 1$  to  $N_E$  do
9     Estimate proportions using modified Gramm-Schmidt orthogonalization
10    Residual  $\epsilon = x - p \cdot e$ 
11    RMS error  $rms \leftarrow (\sum_{k=1}^{\lambda} (\epsilon_{ik})^2 / N_E)^{1/2}$ 
12    Residual Count  $N_\epsilon \leftarrow 0$  (Initialization)
13    while ( $N_\epsilon < N_{\epsilon, thres}$ ) do
14      if  $p \in [-0.01, 1.01]$  and  $rms < 0.025$  then
15        Return  $E_L$ 
16         $Model(i,j) \leftarrow 1$ 
17      elseif  $\epsilon > \epsilon_{thres}$  then
18         $N_\epsilon \leftarrow N_\epsilon + 1$ 

```

```

19           endif
20           until converge (or until maxIteration)
21       end while
22       Return  $p$ 
23   end for
24   Return  $Model$ 

```

#### **Reduction of candidate models**

```

25   if num. of pixels covered by model < 0.001% *N or spatially fragmented then
26       Remove model
27   endif

```

#### **Pixel Subsampling**

```

28   Subsample candidate models every  $m^{th}$  sample &  $m^{th}$  line
29   Eliminate dominated models
30   Consolidate equivalent pixels if pixel can be represented by the same model

```

#### **Optimization of candidate models**

```

31   Minimizing (16) with constraints (17) (18) using CPLEX and heuristic  $\lambda$ -Opt
32   Return optimized selected endmember candidate models
33
34   Return Pestimate for each of the selected endmember candidate models

```

#### **Map Fraction image**

---

34      Categorize each fraction image and combined at pixels

The MESMA algorithm is able to model pixels using elements (endmember candidates) from endmember sets, thus incorporating endmember variability. However, if the spectral library is large and the number of elements is large for each endmember set, MESMA may result in high computation expense and inefficiency, as MESMA will search for all combinations of endmember candidates to find a suitable model.

Extensions based on MESMA algorithm can be found in (Dennison, et al., 2003; Dennison, et al., 2004; Roberts, et al., 2003; Franke, et al., 2009; Raksuntorn, et al., 2008; Tits, et al., 2012).

**Multiple-Endmember Linear Spectral Unmixing (MELSUM).** Similar to MESMA method introduced above, MELSUM approach (Combe, et al., 2008) builds a

library of candidate endmembers from known endmember library and searches the library to select endmember elements. On the other hand, MELSUM finds proportion values that satisfy the minimum root mean square (RMS) error for each pixel, while the above MESMA method uses modified Gramm-Schmidt orthogonalization to find proportions.

MELSUM method claims to offer a better compromise between time and computing complexity (computing time). However, MELSUM ignores the sum-to-one constraint and only considers the non-negative constraint of the proportion values.

**Automated Monte Carlo Unmixing (AutoMCU).** AutoMCU algorithm (Asner, et al., 2003) randomly selects one element from each of the endmember sets instead of constructing and searching for all possible endmember candidates like MESMA and MELSUM mentioned above. The pixel is represented using the linear mixing model by the selected endmember elements weighted by proportions. The proportion values in AutoMCU are then estimated from the randomly selected endmember elements by solving linear equations.

AutoMCU runs the endmember selection process iteratively and the mean and standard deviation of proportion values of each pixel can be calculated based on results from the multiple runs.

### 2.2.3. Endmember Bundles Methods

**Endmember Bundles.** Endmember Bundles algorithm (Bateson, et al., 2000) represents each endmember as “a set or bundle of spectra”, and each “bundle” contains multiple examples of an endmember spectral signature.

The endmember bundles in Endmember Bundles algorithm are initialized with an automatically generated endmembers (“seeds”) from the manual endmember selection method (Bateson, et al., 1996). Nearby pixels are added to the bundle as long as their reflectance values range from 0 to 1 and are sufficiently correlated (correlation coefficient  $R > 0.99$ ) with seed endmembers generated above. After constructing the endmember bundle, proportions are estimated through bundle unmixing, which is similar to the linear programming method used in (Bajjouk, et al., 1998). The objective function of Endmember Bundles unmixing method, as shown in equation (19), is minimized,

$$\sum_{i=1}^{TB} \chi_i C_i , \quad 19$$

where  $\chi_i = \begin{cases} 1, & bh_i \in B_t \\ 0, & bh_i \notin B_t \end{cases}$

$C_i$  is the proportion of endmember  $i$ ,  $B_t$  is the convex hull of an endmember bundle. The following constraints are satisfied,

$$Spectrum (component j) = \sum_{i=1}^{TB} C_i bh_i (component j) , j = 1, \dots N - 1; \quad 20$$

$$\sum_{i=1}^{TB} C_i bh_i (component j) = 1, \quad 0 \leq C_i \leq 1. \quad 21$$

where  $bh_i$  is the endmember value in bundle  $i$ . Each  $bh_i$  belongs to the convex hull of a bundle,  $TB$  is the total number of points in all bundle hulls.

The objective function for Endmember Bundles unmixing using Dikin’s method (Dikin, 1967) is written as

$$\sum_{i=1}^{TB} x_i C_i + Inf * s , \quad \text{where } x_i = f(x) = \begin{cases} 1, & bh_i \in B_t \\ 0, & bh_i \notin B_t \end{cases} \quad 22$$

with constraints

$$Spectrum (component j) = \sum_{i=1}^{TB} C_i bh_i (component j) + s * d_j , j = 1, \dots N - 1 \quad 23$$

where  $d_j = \text{pixel}(\text{component } j) - \frac{1}{TB} * \sum_{i=1}^{TB} bh_i(\text{component } j)$ , and  $\text{Inf}$  is a very large number driving dummy variable  $s$  to zero when objective function (22) is minimized.

The mean proportion values can be calculated using

$$\mathbf{p}_{\text{mean}} = \frac{\mathbf{p}_{\min} + \mathbf{p}_{\max}}{2} \quad 24$$

after calculating the maximum proportions  $\mathbf{p}_{\max}$  and minimum proportions  $\mathbf{p}_{\min}$  through bundle unmixing.

In order to ensure the sum-to-one constraint on proportion values, the perturbed proportion value,  $C_{p_i}$  for bundle  $i$ , is calculated using

$$C_{p_i} = \mathbf{p}_{\text{mean}_i} + (1 - S) \cdot \frac{\mathbf{p}_{\max_i} - \mathbf{p}_{\min_i}}{2 \cdot SI} \quad 25$$

where  $S = \sum_{i=1}^N \mathbf{p}_{\text{mean}_i}$  is the sum of mean proportions for each pixel  $i$ .  $SI = \sum_{i=1}^N \frac{\mathbf{p}_{\max_i} - \mathbf{p}_{\min_i}}{2}$  is the sum of half intervals between the maximum and minimum proportions for pixel  $i$ .

The pseudo code for Endmember Bundles algorithm is as follows.

#### Algorithm 2 : Endmember Bundles (Bateson, et al., 2000)

1 Reduce the dimension of input data using PCA

##### **Endmember bundle construction**

2 Initialize seeds of endmember bundles using manual endmember selection method  
 3 Calculate correlation coefficients of reflectance values between seeds and the nearby pixels of the seeds  
 4 **for**  $i \leftarrow 1$  to total number of endmember bundles **do**  
 5     **for**  $j \leftarrow 1$  to total number of nearby points  $N$  **do**  
 6         **if** correlation coefficient  $R > 0.99$  and reflectance value  $\in [0,1]$  **then**  
 7             Add pixel  $j$  into the bundle  $i$   
 8         **endif**

```

9      endfor
10 endfor
11 for  $i \leftarrow 1$  to total number of endmember bundles do
12     if correlation coefficient of any two points  $j, k$  in the bundle  $\geq 0$  and sum = 1 then
13       Add linear combination of points  $j, k$  into the bundle  $i$ 
14     endif
15 endfor

Endmember bundle unmixing
16 for  $i \leftarrow 1$  to total number of endmember bundles do
17   Minimizing (19) with constraints (20) (21) using Linear Programming
18   Return  $\mathbf{p}_{min}$ 
19 end for

20 Repeat steps 16-19 by replacing  $\chi_i \leftarrow (1 - \chi_i)$ 
21 Return  $\mathbf{p}_{max}$ 

22 Dikin's affine linear programming algorithm
23 Repeat steps 16-19 by Minimizing (22) with constraints (23)
24 Return  $\mathbf{p}_{min}$  and  $\mathbf{p}_{max}$  for Dikin's method

25 for  $i \leftarrow 1$  to total number of endmember bundles do
26   Compute mean fraction image for each bundle using (24)
27   Compute perturbed mean fraction image for each bundle using (25)
28 endfor

29 Plot endmember bundles and proportion maps


---



```

Extensions of Endmember Bundles method include (Somers, et al., 2012), where a fully automated approach for constructing endmember bundles are developed; and (Zare, et al., 2012), where in addition to automatically constructing endmember bundles, the number of endmember sets are estimated automatically as well by the use of the Competitive Agglomeration Clustering (Frigui, et al., 1997) algorithm.

#### 2.2.4. Local Unmixing

Local unmixing methods extract endmember elements and group them into endmember sets based on the assumption that pixels that are spatially close to each other are likely to be composed of the same endmember elements.

**Spatially Adaptive Hyperspectral Unmixing.** Spatially Adaptive Hyperspectral Unmixing method (Canham, et al., 2011) considers the variation of endmembers within a global image. In this method, the input hyperspectral image is subdivided into segments, or “local areas”. For each local area, local endmembers are extracted using any of the endmember extraction methods proposed by (Rogge, et al., 2007; Plaza, et al., 2004; Plaza, et al., 2002; Zare, et al., 2007). After obtaining the endmembers, linear unmixing is performed with the local endmembers pixel by pixel. Then, spatially and spectrally near endmembers are clustered into groups and all endmembers in the groups are used in obtaining a global proportion map.

**Temporal unmixing analysis based on local unmixing.** Temporal unmixing analysis method (Goenaga, et al., 2012) splits the input hyperspectral image scene into segments (“tiles”) and assumes that a single spectral signature is an adequate representation of an endmember within a tile. Endmember spectral signatures are extracted from each tile and clustered to form endmember classes in order to account for endmember spectral variability across the scene (similar to Spatially Adaptive Hyperspectral Unmixing method mentioned above).

Proportion values in this method are estimated using all endmembers extracted from the scene by solving the constrained linear least squares problem, as follows,

$$\mathbf{p} = \arg \min \|\mathbf{X} - \mathbf{PE}\|^2, \text{ with constraints } p_{ij} > 0, \sum_i p_{ij} \leq 1. \quad 26$$

The proportion of an endmember class is computed as the sum of proportion values for all endmembers that belongs to the endmember class.

### 2.3. Endmember as Statistical Distributions

In this representation, each endmember is represented using a full distribution.

The Normal Compositional Model (NCM) is the most commonly discussed in the literature. In the Normal Compositional Model, endmembers are represented using multivariate Gaussian distributions. A data point can be written as

$$\mathbf{x}_i | \mathbf{E}_i, \mathbf{p}_i \sim \mathcal{N}(\cdot | \sum_{m=1}^M p_{mi} \boldsymbol{\mu}_m, \sum_{m=1}^M p_{mi}^2 \boldsymbol{\Sigma}_m), \quad 27$$

where  $\mathbf{x}_i$  is the  $i^{th}$  pixel,  $M$  is the total number of endmembers,  $\boldsymbol{\mu}_m$  and  $\boldsymbol{\Sigma}_m$  are the mean and variance of the Gaussian distribution for the  $m^{th}$  endmember. The data likelihood can be written as:

$$f(\mathbf{x}_i | \mathbf{E}_i, \mathbf{p}_i) \sim \mathcal{N}(\cdot | \mathbf{p}_i \mathbf{E}, \sum_{m=1}^M p_{mi}^2 \boldsymbol{\Sigma}_m). \quad 28$$

Also, Gamma distribution is investigated in the literature (Dobigeon, et al., 2008; Dobigeon, et al., 2009).

#### 2.3.1. Gaussian Distribution

**EM algorithm using Normal Compositional Model.** Stein and Eismann (Stein, 2003; Eismann, 2006) proposed an Expectation-Maximization (EM) algorithm under Normal Compositional Model to find proportion values and endmember spectra. The mean of endmember Gaussian distribution is initialized with endmembers obtained by an endmember determination technique. The covariance matrices  $\boldsymbol{\Sigma}_m$  are obtained using the

covariance matrices of the clusters of points that are the closest to the endmembers. Proportion values are obtained by selecting the proportion values that maximize the data likelihood, as shown in (28).

**Bayesian estimation using Normal Compositional Model.** Eches (Eches, et al., 2010) proposed a method to estimate proportions using a Bayesian algorithm under the Normal Compositional Model. In this method, endmembers are assumed to be random in order to model uncertainties. Each pixel of the hyperspectral image is represented using a linear combination of such Gaussian-distributed random endmembers. The means of the endmember distributions are derived based on the N-FINDR or VCA algorithms (Winter, 1999; Nascimento, et al., 2005; Eches, et al., 2010).

The algorithm represents both parameter priors and endmember variances in distributions. A prior distribution is placed on endmember variance,  $\sigma^2$ . The endmember variance is assumed to be distributed according to an inverse Gamma distribution,

$$\sigma^2 | \delta \sim \mathcal{IG}(\nu, \delta), \quad 29$$

where  $\nu, g$  are two adjustable hyperparameters. Eches assumed that the prior of  $\delta$  is non-informative Jeffreys' prior defined by

$$f(\delta) \propto \frac{1}{\delta} \mathbf{1}_R + \delta \quad 30$$

The conditional distribution on the hyperparameter  $\delta$  of endmember variance  $\sigma^2$  is distributed according to a Gamma distribution,

$$\delta | \sigma^2 \sim \mathcal{G}\left(1, \frac{1}{\sigma^2}\right), \quad 31$$

where  $\mathcal{G}(a, b)$  is the Gamma distribution with shape parameter  $a$  and scale parameter  $b$ .

In this method, a data point  $i$  can be written as

$$\mathbf{x}_i | \boldsymbol{\alpha}_i, \sigma^2 \sim \mathcal{N}(\cdot | \sum_{m=1}^M \alpha_{mi} \boldsymbol{\mu}_m, \sum_{m=1}^M \alpha_{mi}^2 \sigma^2 \mathbf{I}) \quad 32$$

where  $\mu_m$  are known endmember means,  $\sigma^2$  is endmember variance,  $\alpha$  are abundances (i.e. proportions). If the image has  $N$  pixels,  $i = 1, \dots, N$ .

After using Bayes rule, the proportion posterior can be calculated. Generating  $\alpha$  according to Bayes' theorem gives

$$f(\alpha|y, \sigma^2) \propto \frac{1}{\sigma^2 c(\alpha)^{\frac{N}{2}}} \exp\left(-\frac{\|y - \mu(\alpha)\|^2}{2\sigma^2 c(\alpha)}\right) \mathbf{1}_S(\alpha). \quad 33$$

where  $c(\alpha) = \sum_{r=1}^R \alpha_r^2$ ,  $r = 1, \dots, R$  is the  $r^{th}$  endmember,  $R$  is the total number of endmembers.  $\mathbf{1}_S(\alpha)$  is an indicator function that is defined as

$$\mathbf{1}_S(\alpha) = \begin{cases} 1, & \text{if } \alpha \in S \\ 0, & \text{otherwise} \end{cases} \quad 34$$

where  $S$  represents the simplex defined by abundances satisfying the non-negativity and sum-to-one constraint,

$$S = \{\alpha | \alpha_r \geq 0, \forall r = 1, \dots, R-1, \sum_{r=1}^{R-1} \alpha_r \leq 1\}. \quad 35$$

The model above assumes that all endmember spectra have the same variance,  $\sigma^2$ . If the model is extended to where endmembers have different variances, i.e.  $\sigma^2 = [\sigma_1^2, \dots, \sigma_R^2]^T$ , then the variance will be generated according to

$$f(\sigma_r^2 | \sigma_{-r}, X, A, \delta) \propto f(X|A, \sigma)f(\sigma_r^2 | \nu, \delta). \quad 36$$

where  $\sigma_{-r} = [\sigma_1^2, \dots, \sigma_{r-1}^2, \sigma_{r+1}^2, \dots, \sigma_R^2]^T$ ,  $A$  is the proportion matrix given by

$$A = \begin{pmatrix} \alpha_{1,1} & \cdots & \alpha_{1,N} \\ \vdots & \ddots & \vdots \\ \alpha_{R,1} & \cdots & \alpha_{R,N} \end{pmatrix}, \quad 37$$

data  $X = EA$ ,  $E = [e_1, \dots, e_R]$  is endmember matrix.  $\nu, \delta$  are hyperparameters.

The conditional distribution of variance  $\sigma_r^2$  is distributed according to a conjugate inverse Gamma distribution,

$$\sigma_r^2 \sim \mathcal{IG}(\alpha_\sigma, \beta_\sigma). \quad 38$$

where  $\alpha_\sigma, \beta_\sigma$  are adjustable parameters. The hyperparameter  $\delta$  is distributed according to a Gamma distribution given by

$$\delta \sim \mathcal{G} \left( R, \sum_{r=1}^R \frac{1}{\sigma_r^2} \right). \quad 39$$

The pseudo-code for this method can be seen in Algorithm 3 and Algorithm 4.

---

Algorithm 3 : Hybrid Gibbs Sampler for NCM (Eches, et al., 2010)

---

**Initialization**

- 1 Sample the hyperparameter  $\delta^{(0)}$  from (30)
- 2 Sample  $\sigma^{2(0)}$  from (29)

**Iterations**

- 3 **for**  $t \leftarrow 1$  to max iterations **do**
  - 4     Sample  $\alpha^{(t)}$  from (33) using Metropolis-within Gibbs
  - 5     Sample  $\sigma^{2(t)}$  from (38)
  - 6     Sample  $\delta^{(t)}$  from (31)
  - 7 **endfor**
- 

---

Algorithm 4 : Bayesian Estimation using NCM (Eches, et al., 2010)

---

**Initialization**

- 2 Sample the hyperparameter  $\delta^{(0)}$  from (30)
  - 3 Sample  $\sigma^{(0)} = [\sigma_1^{2(0)} \dots \sigma_R^{2(0)}]$  from (29)
  - 4 **for**  $j \leftarrow 1$  to number of pixels  $N$  **do**
  - 5     Sample  $\alpha^{(0)}$  according to a uniform distribution
  - 6 **endforIterations**
  - 7 **for**  $p \leftarrow 1$  to number of pixels  $N$  **do**
  - 8     Sample  $\alpha_p^{(t)}$  from (36) using Metropolis-within-Gibbs
  - 9     **for**  $r \leftarrow 1$  to number of endmembers  $R$  **do**
  - 10         Sample  $\sigma_r^{2(t)}$  from (36) using Metropolis-within-Gibbs
  - 11         Sample  $\delta^{(t)}$  from (31)
  - 12     **endfor**
  - 13 **endfor**
- 

Algorithm 3 shows the process for Metropolis-within-Gibbs sampler to generate abundances  $\alpha^{(t)}$ , endmember variance  $\sigma^{2(t)}$  and hyperparameter  $\delta^{(t)}$ . Algorithm 4 shows the algorithm with extension to where endmembers have different variances.

**Sampling Piece-wise Convex Endmember Extraction (S-PCUE).** The S-PCUE algorithm (Zare, et al., 2012) also uses MCMC methods. S-PCUE iteratively samples the parameters of interest using a Metropolis-within-Gibbs sampling method. It is assumed that the covariance values of endmember distributions are known and estimates the endmember mean values.

The proportion values for all date points have a prior Dirichlet distribution,

$$p_j | z_j = r \sim D_M(\alpha_{1,r}, \dots, \alpha_{M,r}), \quad 40$$

where  $D_M(\bullet)$  is an  $M$ -factor Dirichlet distribution with density function

$$D_M(p_j | z_j = r) = \frac{\Gamma(\sum_{m=1}^M \alpha_{m,r})}{\prod_{m=1}^M \Gamma(\alpha_{m,r})} \prod_{m=1}^M p_{jm}^{\alpha_{m,r}-1} \quad 41$$

According to the convex combination of Gaussian distributions, the likelihood can be written as  $f(x_j | E, p_j) = N(p_j E, \sum_{m=1}^M p_{jm}^2 S_m)$  where  $E$  is an  $M \times D$  matrix of endmember mean values.

The pseudo code for updating proportions in S-PCUE is shown in Algorithm 5.

---

Algorithm 5 : Sampling-Update Proportions (Zare, et al., 2012)

---

```

1 Initialize Proportions by sampling from the Dirichlet Prior (41)
2 Likelihood  $\leftarrow f(x_j | E, p_j^{initial})$ 
3 for  $j \leftarrow 1$  to number of data points  $N$  do
4     Sample new proportions samples according to the Dirichlet distribution (40)
5     LikelihoodNew  $\leftarrow f(x_j | E, p_j^{new})$ 
6     Acceptance rate  $a_p \leftarrow \min \left\{ \frac{f(x_j | E, p_j^{new})}{f(x_j | E, p_j^{old})}, 1 \right\}$ 
7     Generate a random number  $r$  from  $[0, 1]$ 
8     if  $r < a_p$  then
9         Likelihood  $\leftarrow$  LikelihoodNew
10        Proportions  $p \leftarrow$  new proportion sample values
11    else
12        Likelihood  $\leftarrow$  LikelihoodOld
13        Proportions  $p \leftarrow$  old proportion sample values

```

```
14      endif  
15      until converge (or until maxIteration)  
16 endfor
```

---

### 3. HYPERSPECTRAL UNMIXING USING QUADRATIC PROGRAMMING WITH GAUSSIAN DISTRIBUTIONS

As mentioned in Section 2.3.1, the Normal Compositional Model (NCM) is the most commonly discussed in the literature. In the Normal Compositional Model, endmembers are represented using multivariate Gaussian distributions. In this thesis, NCM models are used as a comparison to the Beta Compositional Models to perform unmixing.

The purpose of unmixing algorithms is to find the best proportion values under known endmember values that will satisfy the two proportion constraints mentioned in Section 1.2.1: non-negativity and sum-to-one. Under NCM model, two approaches of unmixing are discussed: quadratic programming (QP) approach and Metropolis-Hastings (MH) sampling approach. These two approaches will be discussed and compared with the two approaches under BCM model in later chapters.

#### 3.1. Model Description

In this thesis, it is assumed that hyperspectral unmixing follows the Linear Mixing Model (LMM). Remember that as mentioned in Section 1.2.1, the LMM model can be written as (Keshava, 2003):

$$\mathbf{x}_i = \sum_{j=1}^M p_{ij} \mathbf{e}_j + \boldsymbol{\varepsilon}_i, j = 1, \dots, M \quad 42$$

where  $\mathbf{x}_i$  is the spectral measurement (actual image data point spectrum value) at pixel  $i$ ,  $M$  is the total number of endmembers,  $\mathbf{e}_j$  is the  $j^{th}$  endmember (which is a  $1 \times D$  vector,  $D$  is the total number of spectral bands),  $p_{ij}$  is the proportion (scalar) of endmember  $j$  at

pixel  $i$ , and  $\boldsymbol{\varepsilon}_i$  is an error term for the  $i^{th}$  pixel. The proportions of this model satisfy the following constraints:

$$p_{ij} \geq 0, j = 1, \dots M$$

43

$$\sum_{j=1}^M p_{ij} = 1$$

Equation (42) can be re-written as follows in matrix form:

$$\boldsymbol{\varepsilon} = \mathbf{X} - \mathbf{P}\mathbf{E}$$

44

Where  $\mathbf{X}$  is a  $N \times D$  matrix of the spectral measurements (actual image data point spectrum value),  $N$  is the total number of data points,  $D$  is the total number of spectral bands,  $M$  is the total number of endmembers,  $\mathbf{E}$  is the endmember matrix (which is an  $M \times D$  matrix),  $\mathbf{P}$  is the proportion matrix at all pixels for all endmembers ( $N \times M$ ), and  $\boldsymbol{\varepsilon}$  is an error term ( $N \times D$ ).

The desired proportion values are achieved when the error term  $\boldsymbol{\varepsilon}$  is minimized. Thus, equation (44) is treated as the objective function and quadratic programming (QP) (Nocedal, et al., 2006) is applied to minimize the objective function.

The general form of QP can be written as follows (Nocedal, et al., 2006):

$$\text{Minimize } f(x) = \frac{1}{2} x^T H x + F^T x$$

45

while subject to constraints  $A \cdot x \leq b$ ,  $Aeq \cdot x = beq$ , and bounds  $lb \leq x \leq ub$ . Comparing to equations (43) and (44), error term  $\boldsymbol{\varepsilon}$  is minimized, so the objective function can be written as:

$$\text{Minimize } J = \|\mathbf{X} - \mathbf{P}\mathbf{E}\|^2 = \mathbf{X}^T \mathbf{X} - 2\mathbf{X}^T \mathbf{P}\mathbf{E} + \mathbf{P}^T \mathbf{E}^T \mathbf{P}$$

46

The optimization method takes the derivative of the above objective function over  $\mathbf{P}$  and set it to zero. The first term  $\mathbf{X}^T \mathbf{X}$  does not concern  $\mathbf{P}$ , so Equation (46) can be written as:

$$\text{Minimize } J = (P^T E^T - 2X^T)EP$$

47

Comparing to the generalized form for QP in Equation (45), substitute “ $x$ ” with  $P$ , the rest of the parameters  $H$  and  $F$  can be written as:

$$\begin{aligned} H &= 2E^T E \\ F &= (-2X^T E)^T \end{aligned}$$

48

There are no equality or lower and upper bound constraints, so  $A_{eq}$ ,  $b_{eq}$ ,  $lb$  and  $ub$  are set to empty sets. There are, however, strict requirements on the non-negativity and sum-to-one characteristics of proportion values. The non-negativity constraint can be written as (43). The sum-to-one constraint can be regarded as both  $1 \cdot \sum_{j=1}^M p_{ij} \leq 1$  and  $-1 \cdot \sum_{j=1}^M p_{ij} \leq -1$  have to be satisfied. Therefore, comparing to the constraint  $A \cdot P \leq b$  in the generalized QP form,  $A$  and  $b$  can be written in the following matrix form,

$$\begin{aligned} A_1 &= \begin{bmatrix} -1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & -1 \end{bmatrix}_{M \times M}, \quad A_2 = \begin{bmatrix} 1 & \cdots & 1 \\ -1 & \cdots & -1 \end{bmatrix}_{2 \times M}, \\ A &= \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}_{(M+2) \times M}; \\ b_1 &= [0 \quad \dots \quad 0]_{M \times 1}^T, \quad b_2 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}_{2 \times 1}, \\ b &= \begin{bmatrix} b_1 \\ b_2 \end{bmatrix}_{(M+2) \times 1}. \end{aligned}$$

49

Substituting  $H$  and  $F$  into the generalized QP form, adding  $A$  and  $b$  inequality constraints as mentioned above, the proportions  $P$  can be determined using a quadratic programming solver, such as the built-in function *quadprog* in MATLAB and *QP\_Solver* package in Computational Geometry Algorithms Library (CGAL) (CGAL Project, 2013).

## 4. HYPERSPECTRAL UNMIXING USING SAMPLING METHOD WITH GAUSSIAN DISTRIBUTIONS

Another approach, Metropolis-Hastings (MH) sampling approach, is discussed under NCM model. While the QP approach discussed in Chapter 3 obtains proportions by minimizing the difference between true data values and the data values computed from estimated proportions and endmembers, MH sampling approach draws samples for proportions from a Dirichlet distribution and update proportion values based on the data likelihood.

### 4.1. Model Description

Hyperspectral unmixing with Gaussian distributions can follow the S-PCUE model (Zare, et al., 2012). In this model, each endmember is represented using Gaussian distributions in order to account for endmember variability, with a mean value of  $\mathbf{e}_m$  and an isotropic diagonal covariance  $S_m$  for the  $m^{th}$  endmember distribution (Zare, et al., 2012),  $\mathbf{e}_m$  is a  $1 \times D$  vector, where  $D$  is the number of bands,  $S_m$  is a small fixed number and it is same for all the endmembers in the scene. The distribution of the  $m^{th}$  endmember can then be written as  $\hat{\mathbf{e}}_m \sim N(\mathbf{e}_m, S_m)$ .

The mean value  $\mathbf{e}_m$  and the covariance  $S_m$  for the  $m^{th}$  endmember distribution are known in the algorithm. The proportion values for all the data points are given a prior. A Dirichlet distribution  $D_K$  of order  $K$  and parameter  $\alpha_i$ ,  $i = 1, \dots, K$  (as shown below in Equation (50)) is put on the proportion values,

$$D_M(p_j) = \frac{\Gamma(\sum_{i=1}^M \alpha_i)}{\prod_{i=1}^M \Gamma(\alpha_i)} \prod_{i=1}^M p_{ji}^{\alpha_i}, \quad i = 1, \dots, M, j = 1, \dots, N. \quad 50$$

The distribution on the prior of proportion of the  $j^{th}$  data point can then be written as  $p_j \sim D_M(\alpha_1, \dots, \alpha_M)$ . Note that  $N$  is the number of data points in the input image data, and  $M$  is the number of endmembers in the image scene. Dirichlet distribution is used here as the Dirichlet distribution satisfies the “non-negative” and “sum-to-one” constraints on the proportions.

For the parameter  $\alpha_i$  in the Dirichlet distribution,  $\alpha$  is set to be **1** to make it symmetrical and uniform.

#### 4.2. Metropolis-Hastings (MH) Sampling Approach

Metropolis-Hastings (MH) approach (Chib, et al., 1995; Bishop, 2006) is used to sample proportion values for each endmember across all the data points. Metropolis-Hastings (MH) algorithm is a Markov chain Monte Carlo (MCMC) method for obtaining random samples from a probability distribution. In this case, the MH algorithm is used to obtain proportion sample values from the Dirichlet distribution.

An acceptance rate term is defined as:

$$a_p = \min \left\{ \frac{f(\mathbf{x}_j | \mathbf{E}, \mathbf{p}_j^{new})}{f(\mathbf{x}_j | \mathbf{E}, \mathbf{p}_j^{old})}, 1 \right\} \quad 51$$

Here  $f(\mathbf{x}_j | \mathbf{E}, \mathbf{p}_j)$  is the likelihood of the data point given the known endmember and sampled proportion values. As mentioned in Section 2.3.1, the likelihood can be written

as  $f(\mathbf{x}_j|\mathbf{E}, \mathbf{p}_j) = N(p_j \mathbf{E}, \sum_{m=1}^M p_{jm}^2 \mathbf{S}_m)$  where  $\mathbf{E}$  is an  $M \times D$  matrix of endmember mean values.

If  $f(\mathbf{x}_j|\mathbf{E}, \mathbf{p}_j^{new}) \geq f(\mathbf{x}_j|\mathbf{E}, \mathbf{p}_j^{old})$ , the likelihood yielded from the new proportion sample  $\mathbf{p}_{new}$  is better than the likelihood yielded from the old proportion sample  $\mathbf{p}_{old}$ , which would make  $a_p$  always equal to 1. A random number  $\gamma_p$  is generated from a uniform distribution between zero and one. In this case,  $\gamma_p = rand(1)$  will always be less or equal to  $a_p (= 1)$ , and  $\mathbf{p}_{new}$  is accepted as a new proportion sample 100% percent of the time. However, when  $f(\mathbf{x}_j|\mathbf{E}, \mathbf{p}_j^{new}) < f(\mathbf{x}_j|\mathbf{E}, \mathbf{p}_j^{old})$ , the new proportion sample  $\mathbf{p}_{new}$  makes the likelihood worse than the likelihood yielded from the old proportion sample  $\mathbf{p}_{old}$ ,  $a_p = \frac{f(\mathbf{x}_j|\mathbf{E}, \mathbf{p}_j^{new})}{f(\mathbf{x}_j|\mathbf{E}, \mathbf{p}_j^{old})}$ . If  $\gamma_p = rand(1) < a_p$ ,  $\mathbf{p}_{new}$  is accepted and saved. If  $\gamma_p = rand(1) > a_p$ , then  $\mathbf{p}_{new}$  is not accepted, and  $\mathbf{p}_{old}$  is kept as proportion sample. In this way, when the new proportion sample makes the likelihood better than the old proportion sample, the new proportion sample value is always saved as a good  $\mathbf{p}$  value; but if the new proportion sample produces a worse likelihood, the new proportion sample value is still sometimes accepted, but with a probability of  $a_p = \frac{f(\mathbf{x}_j|\mathbf{E}, \mathbf{p}_j^{new})}{f(\mathbf{x}_j|\mathbf{E}, \mathbf{p}_j^{old})}$ , so that it is possible to learn the whole distribution shape of the proportion values.

The sampling and updating process runs through a pre-set number of iterations till convergence. A log likelihood trace is used to track the progress. For every iteration, the best value of *LogLikelihoodTrace*, *LogLikelihoodTraceBest*, is updated. If in the next iteration, *LogLikelihoodTrace* is larger than the current

*LogLikelihoodTraceBest*, the latter term is replaced with the former value and the corresponding  $\mathbf{P}$  values are saved. If not, the current best value will be kept and the corresponding  $\mathbf{P}$  values will be saved, until the trace term is updated and replaced again.

In hyperspectral unmixing with Gaussian distribution, endmembers are assumed to be known. That is to say, the mean and covariance of the endmember Gaussian distributions are known. The pseudo code for hyperspectral unmixing with Gaussian distribution algorithm is shown as follows:

---

**Algorithm 6: Hyperspectral unmixing with Gaussian distribution**

---

```

1 Initialize proportions with Dirichlet prior (50)
2  $Likelihood \leftarrow f(x_j|E_{mean}, p_j^{initial})$ 
3 for  $j \leftarrow 1$  to number of data points  $N$  do
4     Sample new proportions samples according to Dirichlet
5      $LikelihoodNew \leftarrow f(x_j|E_{mean}, p_j^{new})$ 
6     Acceptance rate  $a_p \leftarrow \min \left\{ \frac{f(x_j|\mathbf{E}, \mathbf{p}_j^{new})}{f(x_j|\mathbf{E}, \mathbf{p}_j^{old})}, 1 \right\}$ 
7     Generate a random number  $\gamma_p$  from  $[0, 1]$ 
8     if  $\gamma_p < a_p$  then
9          $Likelihood \leftarrow LikelihoodNew$ 
10        Proportions  $\mathbf{p} \leftarrow \mathbf{p}_{new}$ 
11    else
12         $Likelihood \leftarrow LikelihoodOld$ 
13        Proportions  $\mathbf{p} \leftarrow \mathbf{p}_{old}$ 
14    endif
15    until max Iteration
16 endfor

```

---

## 5. HYPERSPECTRAL UNMIXING USING QUADRATIC PROGRAMMING METHOD WITH BETA DISTRIBUTIONS

Two approaches are proposed under the Beta Compositional Models to perform unmixing: quadratic programming (QP) approach (Chapter 5) and Metropolis-Hastings (MH) sampling approach (Chapter 6). These two approaches are compared with the two approaches under NCM model in Chapters 3 and 4.

### 5.1. Model Description

Since the Gaussian distribution is defined on the interval of minus infinity to infinity, Gaussian endmembers allow samples outside the interval of [0, 1]. However, in reality, the reflectance values of endmember spectra usually vary between zero and one. Beta distribution, on the other hand, is defined over the interval of [0, 1]. Therefore, in this thesis, Beta distribution is assumed for endmembers in order to make it physically realistic. Beta Compositional Model is introduced as the mixing model in this case.

In Beta Compositional Model (BCM), each endmember at each spectral band is represented using a univariate Beta distribution with parameters  $\alpha$  and  $\beta$  in order to account for endmember variability. The matrix  $e$  is an  $M \times D$  matrix, where  $M$  is the number of endmembers and  $D$  is the number of bands. Beta parameters  $\alpha$  and  $\beta$  are both  $M \times D$  matrices, meaning there will be a set of  $\alpha_{jd}$  and  $\beta_{jd}$  value ( $\alpha_{jd} \geq 0$  and  $\beta_{jd} \geq 0$ ) for each  $e_{jd}$ . The distribution of  $e_{jd}$  can be written as:

$$e_{jd} \sim \mathcal{B}(\cdot | \alpha_{jd}, \beta_{jd}) \quad 52$$

where  $e_{jd}$  is the  $j^{th}$  endmember reflectance value on  $d^{th}$  spectral band.

The data points in BCM are represented as the linear combination of endmembers (Linear Mixing Model),  $x_{id} \sim \mathcal{F}(\bullet | \mathbf{p}_i, \boldsymbol{\alpha}_d, \boldsymbol{\beta}_d)$  where  $x_{id}$  is the spectral measurement (actual image data point spectrum value) at pixel  $i$  on band  $d$ ,  $M$  is the total number of endmembers,  $\mathbf{p}_i$  is an  $M \times 1$  vector for proportion values at pixel  $i$ .  $\boldsymbol{\alpha}_d$  and  $\boldsymbol{\beta}_d$  are  $M \times 1$  vectors containing  $\alpha$  and  $\beta$  parameter values for all  $M$  endmembers on band  $d$ . Similar to Gaussian distribution model case, the proportion values have to satisfy:

$$\begin{aligned} p_{ij} &\geq 0, j = 1, \dots M \\ \sum_{j=1}^M p_{ij} &= 1 \end{aligned} \quad 53$$

However, the resulting distribution of the convex combination of Betas is highly complex. According to (Pham-Gia, et al., 1998; Jóhannesson, et al., 1995), the convolution of only two Beta distributions is,

$$\mathcal{F}_2(x|p, a, b) = \begin{cases} f_1(x|p, a, b) & \text{if } 0 \leq x \leq p_1 \\ f_2(x|p, a, b) & \text{if } p_1 \leq x \leq p_2 \\ f_3(x|p, a, b) & \text{if } p_2 \leq x \leq 1 \end{cases} \quad 54$$

where  $p_1 \leq p_2$  and  $f_1, f_2, f_3$  are

$$f_1(x) = \frac{x^{\alpha_1+\alpha_2-1}(p_1-x)^{\beta_1-1}B(\alpha_1, \alpha_2)}{p_1^{\alpha_1+\beta_1-1}p_2^{\alpha_2+\beta_2-1}B(\alpha_1, \beta_1)B(\alpha_2, \beta_2)} F_1\left(\frac{-x}{p_1-x}, \frac{x}{p_2}, \alpha_2, 1-\beta_1, 1-\beta_2, \alpha_1 + \alpha_2\right) \quad 55$$

$$f_2(x) = \frac{(x-p_1)^{\alpha_2-1}(1-x)^{\beta_2-1}}{p_2^{\alpha_2+\beta_2-1}B(\alpha_2, \beta_2)} F_1\left(\frac{-p_1}{x-p_1}, \frac{p_1}{1-x}, \beta_1, 1-\alpha_1, 1-\beta_2, \alpha_1 + \beta_1\right) \quad 56$$

$$f_3(x) = \frac{1-x^{\beta_1+\beta_2-1}(x-p_1)^{\beta_1-1}B(\beta_1, \beta_2)}{p_1^{\beta_1}p_2^{\alpha_2-\beta_2-1}B(\alpha_1, \beta_1)B(\alpha_2, \beta_2)} F_1\left(\frac{1-x}{p_1}, \frac{x-1}{x-p_1}, \beta_1, 1-\alpha_1, 1-\alpha_2, \beta_1 + \beta_2\right) \quad 57$$

$F_1(\bullet)$  is the first Appell function (Pham-Gia, et al., 1998),

$$F_1(a, b_1, b_2, c; x, y) = \sum_{m,n=0}^{\infty} \frac{(a)_{m+n} (b_1)_m (b_2)_n}{(c)_{m+n} m! n!} x^m y^n, |x| < 1, |y| < 1 \quad 58$$

in which

$$(q)_n = \frac{\Gamma(q+n)}{\Gamma(q)} = q(q+1)\dots(q+n-1). \quad 59$$

The computation is far too complicated even for the convolution of only two Beta distributions. In order to reduce the computational complexity, a single Beta distribution,  $\mathcal{B}(\cdot | e, f)$ , is proposed to approximate the convex combination of Betas (Pham-Gia, et al., 1998; Jóhannesson, et al., 1995) for each pixel on each band. Note that the parameters  $e$  and  $f$  typically have different values for each pixel on each band. The parameters  $e$  and  $f$  of the approximated Beta distribution are obtained by equating the mean and variance of the approximated single Beta distribution to that of a combination of Beta distributions. Given a set of data  $x_d$  that is distributed according to the same convex combination of Beta distributions (i.e., the same proportion vectors), the parameters  $f$  and  $e$  for the approximated single Beta can be computed using (60) and (61) (Jóhannesson, et al., 1995):

$$f = (F - S(1 + F)^2)S^{-1}(1 + F)^{-3} \quad 60$$

$$e = Ff \quad 61$$

where

$$F = E(1 - E)^{-1}, \quad 62$$

$E$  is the expected value of the data used and  $S$  is the variance of the data used.

In this specific case, in order to perform spectral unmixing, a set of spectral or spatial neighbors are found for every pixel  $i$ . The neighbors are found using spectral

neighbors as described in Section 5.2, or using spatial neighbors as described in Section 7.1. The neighbor pixels are regarded as the “source data” to fit one beta distribution. In this case,  $\mathbf{F}$ ,  $\mathbf{E}$ , and  $\mathbf{S}$  are all  $N \times D$  matrices where  $N$  is the number of pixels and  $D$  is the number of bands. The  $i^{th}$  row of  $\mathbf{E}$  matrix, which is a  $1 \times D$  vector, is the expected value of the set of neighbor pixels of pixel  $i$ , on  $D$  bands. The  $i^{th}$  row of  $\mathbf{S}$  matrix, which is a  $1 \times D$  vector, is the variance of the set of neighbor pixels of pixel  $i$ , on  $D$  bands. The parameters  $f_{id}$  and  $e_{id}$  are approximated based on the set of neighbors for each pixel on each band. Given the  $f_{id}$  and  $e_{id}$  values approximated from the neighbor data, and given known Beta endmember parameters  $\boldsymbol{\alpha}$  and  $\boldsymbol{\beta}$ , the proportion values can be estimated by determining the proportions that minimize the difference between the two sides of (63) and (64),

$$\text{Mean: } \frac{e_{id}}{e_{id}+f_{id}} = \sum_{j=1}^M p_{ij} \frac{\alpha_{jd}}{\alpha_{jd}+\beta_{jd}} ; \quad 63$$

$$\text{Variance: } \frac{e_{id}f_{id}}{(e_{id}+f_{id})^2(e_{id}+f_{id}+1)} = \sum_{j=1}^M p_{ij}^2 \frac{\alpha_{jd}\beta_{jd}}{(\alpha_{jd}+\beta_{jd})^2(\alpha_{jd}+\beta_{jd}+1)}. \quad 64$$

Now, the proportion values  $p_{ij}$  can then be determined using either the quadratic programming approach (Section 5.2) or the Metropolis-Hastings sampling method (Section 6.2).

## 5.2. Quadratic Programming Unmixing with Beta Distribution

In the quadratic programming unmixing approach, the difference between the mean of the real data and the mean from the approximated Beta distribution is minimized (Zare, et al., 2013). The objective function can be written as follows:

$$J_{id} = \left\| \frac{e_{id}}{e_{id}+f_{id}} - \sum_{j=1}^M p_{ij} \frac{\alpha_{jd}}{\alpha_{jd}+\beta_{jd}} \right\|^2 \quad 65$$

It is assumed that the pixels with similar spectral values have similar compositions, thus similar proportion values. Therefore, the  $K$  nearest neighbors for pixel  $x_i$  (i.e. the  $K$  pixels with spectral values closest to the value at pixel  $x_i$ ) are found. These pixels are regarded as the “source data” and one beta distribution with parameters  $f_{id}$  and  $e_{id}$  from equations (60) and (61) are fit to the pixel points. The parameters  $f_{id}$  and  $e_{id}$  are estimated using a Maximum Likelihood approximation of these Beta parameters given the source data. Specifically, in our implementation, we use MATLAB’s *betafit* function (Statistics Toolbox, 2013). The  $\alpha_j$  and  $\beta_j$  in (65) are assumed to be known. Then, a quadratic programming method was used to minimize (65) and solve for  $p_{ij}$  given the constraints in (53).

The pseudo code for quadratic programming unmixing based on BCM is shown in Algorithm 7.

---

**Algorithm 7: Beta Compositional Model-Quadratic Programming**

---

```

1 Inputs:  $x_i$ , for  $i = 1, 2, \dots, N$ ;  $K$ -nearest neighbors
2 for  $i \leftarrow 1$  to number of data points  $N$  do
3     Identify  $K$  nearest spectral neighbors for pixel  $x_i$ 
4     Estimate  $f_{id}$  and  $e_{id}$  (betafit)
5     Return proportion values using quadratic programming to minimize the objective
        function (65) subject to the constraints in (53)
6 endfor

```

---

The advantage of this algorithm is that it assumes the Beta Compositional Model and represents endmembers using Beta distributions. It can successfully ensure the range of endmember values to be between zero and one and make it physically realistic. However, the performance depends on the selection of the number of neighbors,  $K$ . This algorithm relies on the assumption that the  $K$  pixels with similar spectral value have similar composition, which may not always be true. Also, only the mean of the

approximated Beta and the mean of the convex combination of Betas are used in the objective function, variances are not included. This may lead to really small variance when the spectral values are really close for the  $K$  nearest neighbors, while it is not necessarily the case across all pixels in the scene.

## 6. HYPERSPECTRAL UNMIXING USING SAMPLING METHOD WITH BETA DISTRIBUTION

Another approach, Metropolis-Hastings (MH) sampling approach, is discussed under BCM model. MH sampling approach draws samples for proportions from a Dirichlet distribution and update proportion values based on the data likelihood.

### 6.1. Model Description

MH sampling method with Beta distribution follows Beta Compositional Model (BCM), as discussed in Section 5.1.

### 6.2. Sampling Unmixing Method with Beta Distribution

A Metropolis-Hastings sampling approach is defined to use both the mean and variance of the real data and data from the approximated Beta distribution (Zare, et al., 2013). In this approach, the mean and variance for the  $d^{th}$  band of a pixel  $x_i$  are:

$$E_{id} = \sum_{j=1}^M p_{ij} \frac{\alpha_{jd}}{\alpha_{jd} + \beta_{jd}} + \sigma_{mean} \quad 66$$

$$S_{id} = \sum_{j=1}^M p_{ij}^2 \frac{\alpha_{jd}\beta_{jd}}{(\alpha_{jd} + \beta_{jd})^2 (\alpha_{jd} + \beta_{jd} + 1)} + \sigma_{var} \quad 67$$

The likelihood function can be written as follows and is maximized:

$$\begin{aligned} & E_{id}, S_{id} | \theta \sim N \left( E_{id} \left| \sum_{j=1}^M p_{ij} \frac{\alpha_{jd}}{\alpha_{jd} + \beta_{jd}}, \sigma_{mean} \right. \right) \times N \left( S_{id} \left| \sum_{j=1}^M p_{ij}^2 \frac{\alpha_{jd}\beta_{jd}}{(\alpha_{jd} + \beta_{jd})^2 (\alpha_{jd} + \beta_{jd} + 1)}, \sigma_{var} \right. \right) \\ & = \frac{1}{\sqrt{2\pi}\sigma_{mean}} \exp \left\{ -\frac{\left( \frac{E_{id} - \sum_{j=1}^M p_{ij} \frac{\alpha_{jd}}{\alpha_{jd} + \beta_{jd}}}{2\sigma_{mean}^2} \right)^2}{2} \right\} \times \frac{1}{\sqrt{2\pi}\sigma_{var}} \exp \left\{ -\frac{\left( \frac{S_{id} - \sum_{j=1}^M p_{ij}^2 \frac{\alpha_{jd}\beta_{jd}}{(\alpha_{jd} + \beta_{jd})^2 (\alpha_{jd} + \beta_{jd} + 1)}}{2\sigma_{var}^2} \right)^2}{2} \right\} \end{aligned} \quad 68$$

where  $\Theta = (\alpha, \beta, \sigma_{mean}, \sigma_{var})$ ,  $\sigma_{mean}$  and  $\sigma_{var}$  are parameters to determine the relative importance placed on the mean and variance in the likelihood.

Similar to the sampling method with Gaussian distribution (section 4.2), Metropolis-Hastings sampling is used in sampling proportion values for each endmember across all the data points using the BCM model and the likelihood in (68). Proportion samples are initialized by drawing samples from a symmetric Dirichlet distribution with  $\alpha$  parameters all equal to one. An acceptance rate term is defined as:

$$a_p = \min \left\{ \frac{f(x_j | \Theta, \mathbf{p}_j^{new})}{f(x_j | \Theta, \mathbf{p}_j^{old})}, 1 \right\} \quad 69$$

The Metropolis-Hastings sampler estimates a full distribution of possible parameter values given a complex data likelihood and prior distribution. In (69),  $f(x_j | \Theta, \mathbf{p}_j)$  is the likelihood of the data point given the known endmember and sampled proportion value. A Dirichlet distribution is used to generate samples of proportion values for evaluation in the likelihood function to be maximized. New proportion sample value is accepted according to  $a_p$  so that it is possible to keep the best proportion samples as well as learning the whole distribution shape of the proportion values. If  $f(x_j | \mathbf{E}, \mathbf{p}_j^{new}) \geq f(x_j | \mathbf{E}, \mathbf{p}_j^{old})$ , that means the likelihood yielded from the new proportion sample  $\mathbf{p}_{new}$  is better than the likelihood yielded from the old proportion sample  $\mathbf{p}_{old}$ , which would make  $\mathbf{p}_{new}$  accepted as a new proportion sample one hundred percent of the time. However, when  $f(x_j | \mathbf{E}, \mathbf{p}_j^{new}) < f(x_j | \mathbf{E}, \mathbf{p}_j^{old})$ , the new proportion sample  $\mathbf{p}_{new}$  makes the likelihood worse than the likelihood yielded from the old proportion sample  $\mathbf{p}_{old}$ ,

$a_p = \frac{f(\mathbf{x}_j | \mathbf{E}, \mathbf{p}_j^{new})}{f(\mathbf{x}_j | \mathbf{E}, \mathbf{p}_j^{old})}$ , new proportion sample value is still sometimes accepted, but with a

probability of  $a_p = \frac{f(\mathbf{x}_j | \mathbf{E}, \mathbf{p}_j^{new})}{f(\mathbf{x}_j | \mathbf{E}, \mathbf{p}_j^{old})}$ , so that we can learn the whole distribution shape of

the proportion values.

The pseudo code for sampling unmixing based on BCM is shown in Algorithm 8.

---

Algorithm 8: Beta Compositional Model-Sampling

---

```

1 Inputs:  $x_i$ , for  $i = 1, 2, \dots, N$ ;  $K$ -nearest neighbors
2 for  $i \leftarrow 1$  to number of data points  $N$  do
3     Identify  $K$  nearest spectral neighbors for pixel  $x_i$ 
4     Estimate  $f_{id}$  and  $e_{id}$  (betafit)
5     for  $i \leftarrow 1$  to number of iterations do
6         Sample new proportions samples according to Dirichlet
7          $LikelihoodNew \leftarrow f(\mathbf{x}_j | \mathbf{E}, \mathbf{p}_j^{new})$ 
8         Acceptance rate  $a_p \leftarrow \min \left\{ \frac{f(\mathbf{x}_j | \mathbf{E}, \mathbf{p}_j^{new})}{f(\mathbf{x}_j | \mathbf{E}, \mathbf{p}_j^{old})}, 1 \right\}$ 
9         Generate a random number  $\gamma_p$  from  $[0, 1]$ 
10        if  $\gamma_p < a_p$  then
11             $Likelihood \leftarrow LikelihoodNew$ 
12            Proportions  $\mathbf{p} \leftarrow \mathbf{p}_{new}$ 
13        else
14             $Likelihood \leftarrow LikelihoodOld$ 
15            Proportions  $\mathbf{p} \leftarrow \mathbf{p}_{old}$ 
16        endif
17        until max Iteration
18    endfor
19 endfor
```

---

## 7. HYPERSPECTRAL UNMIXING BASED ON FLICM WITH BETA DISTRIBUTION

Fuzzy Local Information C-Means Clustering (FLICM) Algorithm performs clustering on image data that incorporates local spatial information (Krinidis, et al., 2010). A hyperspectral unmixing algorithm based on FLICM under BCM model is proposed to incorporate spatial information in unmixing. Both quadratic programming (QP) approach and Metropolis-Hastings (MH) sampling approach are discussed under BCM-FLICM model.

### 7.1. Model Description

Unmixing with Beta distribution follows Beta Compositional Model (BCM), the same as section 5.1.

Both of the above BCM based unmixing approaches relies on the assumption that the  $K$  pixels with similar spectral value have similar compositions and thus can be used to represent pixel  $x_i$ . However, in real hyperspectral image, it is more likely that the pixels that are spatial nearest neighbors share similar compositions. Therefore, a new unmixing algorithm is proposed to take the  $K$  nearest spatial neighbors to perform unmixing.

Spatial-based Fuzzy Local Information C-Means (FLICM) Clustering Algorithm is applied here on the image scene to produce clusters of data. The  $K$  nearest spatial neighbors are selected to be used to estimate proportions for a pixel only if the neighboring pixels satisfy both conditions: 1) belong to the same spatial neighborhood; and 2) belong to the same FLICM cluster. In this way, both spatial and spectral factors are taken into account.

Fuzzy Local Information C-Means Clustering (FLICM) Algorithm performs clustering on image data. FLICM incorporates local spatial information and is free of prior parameter selection (Krinidis, et al., 2010). In the algorithm, a “fuzzy factor”  $G_{ki}$  is introduced to control the influence of neighborhood pixels on the center pixel based on the distance between the neighborhood pixel and the center.  $G_{ki}$  is defined as:

$$G_{ki} = \sum_{j \in N_i, i \neq j} \frac{1}{d_{ij} + 1} (1 - u_{kj})^m \|x_j - v_k\|^2 \quad 70$$

where  $i$  is the center pixel number,  $k$  is the reference cluster and  $j$  is the neighbor pixel number. A total of  $N_i$  numbers of pixels around the center pixel are defined as the “neighbor” pixels.  $d_{ij}$  measures the Euclidean distance between the  $i^{th}$  and  $j^{th}$  pixel. The  $u_{kj}$  value is the membership of pixel  $j$  in cluster  $k$ ,  $m$  is the weight on each fuzzy membership,  $v_k$  is the center pixel of cluster  $k$ , and  $x_j$  is the image data point value for  $j^{th}$  pixel.

The objective function of the Fuzzy C-means (FCM) clustering algorithm (Dunn, 1973; Bezdek, 1981) is:

$$J_{FCM} = \sum_{i=1}^N \sum_{j=1}^c u_{ji}^m d^2(x_i, v_j) \quad 71$$

where  $N$  is the number of data points,  $c$  is the number of clusters,  $2 \leq c \leq N$ ,  $u_{ji}$  is the membership of pixel  $i$  in cluster  $j$ ,  $m$  is the weight on each fuzzy membership,  $d^2(x_i, v_j)$  measures the Euclidean distance between the  $i^{th}$  pixel and cluster center pixel  $v_j$ . In addition to FCM, FLICM incorporates the fuzzy factor  $G_{ki}$  and rewrite the objective function as follows:

$$J_{FLICM} = \sum_{i=1}^N \sum_{k=1}^c [u_{ki}^m d^2(x_i, v_k) + G_{ki}] \quad 72$$

In order to minimize  $J_{FLICM}$ , we take derivative of  $J_{FLICM}$  with respect to  $u_{ki}$  and  $v_k$  and set the derivatives to zero,  $u_{ki}$  and  $v_k$  can be updated as follows:

$$u_{ki} = \frac{1}{\sum_{j=1}^c \left[ \frac{d^2(x_i, v_k) + G_{ki}}{d^2(x_i, v_j) + G_{ji}} \right]^{1/m-1}} \quad 73$$

$$v_k = \frac{\sum_{i=1}^N u_{ki}^m x_i}{\sum_{i=1}^N u_{ki}^m} \quad 74$$

where  $u_{ki}$  measures the degree of membership a pixel  $i$  in cluster  $k$  and  $v_k$  is the cluster center for cluster  $k$ . After the algorithm converges, a “defuzzification” step is taken to assign crisp class numbers  $C_i$  to each pixel.  $C_i$  is defined as:

$$C_i = arg_k \{ \max\{u_{ki}\} \}, k = 1, 2, \dots, c \quad 75$$

The pseudo code for FLICM clustering algorithm is shown in Algorithm 9.

---

**Algorithm 9: FLICM clustering algorithm**

---

- 1 Inputs: data  $x_i, i = 1, \dots, N$ ; number of clusters  $c$ , fuzzification parameter  $m$  and stopping criteria  $\epsilon$
  - 2 Initialize a random fuzzy partition matrix
  - 3 **for**  $t \leftarrow 1$  to number of iterations **do**
  - 4     Calculate cluster prototypes using (74)
  - 5     Calculate membership  $U_t$  using (73)
  - 6     **if**  $\max(U_t - U_{t+1}) > \epsilon$  **then**
  - 7         Repeat Steps 4-5
  - 8     **endif**
  - 9 **endfor**
  - 10 Defuzzify and assign class number to pixel  $i$  using (75)
-

After the image data is clustered and assigned, the  $K$  nearest spatial neighbors for each pixel are identified following the steps shown in Algorithm 10.

---

**Algorithm 10: FLICM/BCM-Identify K nearest spatial neighbors**

---

```

1 Initialization
2 for  $i \leftarrow 1$  to number of data points  $N$  do
3     Identify the cluster  $k$  that pixel  $x_i$  belongs to using Algorithm 9
4     Put a window of size  $W \times W$  around pixel  $x_i$ 
5     for  $j \leftarrow$  all pixels in the window do
6         Identify the cluster  $k_j$  that pixel  $x_j$  belongs to using Algorithm 9
7         if  $k_j = k$ 
8             Identify pixel  $x_j$  as one of nearest spatial neighbors of pixel  $x_i$ 
9         endif
10    endfor
11 endfor

```

---

## 7.2. Quadratic Programming and MH Sampling Unmixing Approaches with Beta Distribution

The quadratic programming and Metropolis-Hastings sampling approaches of BCM-FLICM model follow similar steps as with BCM-spectral model in chapter 5 and 6, except that spatial neighbors of each pixel are identified using FLICM clustering algorithms and such spatial neighbors are used to estimate the parameters  $e$  and  $f$  of the approximated Beta distribution. The pseudo codes for QP and MH approaches unmixing under BCM-FLICM model are as follows:

---

**Algorithm 11: BCM-FLICM: Quadratic Programming Approach**

---

```

1 Initialization
2 for  $i \leftarrow 1$  to number of data points  $N$  do
3     Identify  $K$  nearest spatial neighbors for pixel  $x_i$  using Algorithm 10
4     Estimate  $f_{id}$  and  $e_{id}$  (betafit)
5     Return proportion values using quadratic programming of minimizing objective
       function (65) st. constraints in (53)
6 endfor

```

---

---

Algorithm 12: BCM-FLICM: Metropolis-Hastings Sampling Approach

---

```
1 Initialization
2 for  $j \leftarrow 1$  to number of data points  $N$  do
3     Identify  $K$  nearest spatial neighbors for pixel  $x_j$  using Algorithm 10
4     Estimate  $f_{id}$  and  $e_{id}$  (betafit)
5     for  $i \leftarrow 1$  to number of iterations do
6         Sample new proportions samples according to Dirichlet
7          $LikelihoodNew \leftarrow f(x_j | E, p_j^{new})$ 
8         Acceptance rate  $a_p \leftarrow \min \left\{ \frac{f(x_j | E, p_j^{new})}{f(x_j | E, p_j^{old})}, 1 \right\}$ 
9         Generate a random number  $\gamma_p$  from  $[0, 1]$ 
10        if  $\gamma_p < a_p$  then
11             $Likelihood \leftarrow LikelihoodNew$ 
12            Proportions  $p \leftarrow p_{new}$ 
13        else
14             $Likelihood \leftarrow LikelihoodOld$ 
15            Proportions  $p \leftarrow p_{old}$ 
16        endif
17        until max Iteration
18    endfor
19 endfor
```

---

## 8. HYPERSPECTRAL UNMIXING BASED ON SUPERPIXELS WITH BETA DISTRIBUTION

### 8.1. Superpixels

Superpixels are segments obtained from certain image processing process such as clustering. Compared to pixel-grid, superpixels would be more natural and efficient to study and undertake further processing (Ren, 2004).

Superpixels provide great computational efficiency as the number of components in image processing is reduced largely as compared to the pixel-by-pixel study. Furthermore, as the image of interest is pre-segmented by certain superpixel segmentation algorithms, the spectral and spatial information of each pixel in the image is considered even before unmixing, which provides ideal support for incorporating spectral and spatial information.

In this thesis, two superpixel segmentation methods, graph-based superpixel segmentation method and entropy rate superpixel segmentation method, are used to pre-segment the hyperspectral image before unmixing based on BCM.

### 8.2. Graph-Based Superpixel Segmentation Model

Graph-based superpixel model (Felzenszwalb, et al., 2004) returns superpixels after segmentation on a graph-based image representation. The image is represented using an undirected graph  $G = (V, E)$ , where  $v_i \in V$  are vertices in the graph (the pixels in the image to be segmented) and  $(v_i, v_j) \in E$  are edges (pairs of neighboring pixels).

Two differences concerning superpixels are introduced: Internal difference and difference between. Internal difference within one segment, or one “superpixel”,

measures the largest weight in the minimum spanning tree of the image graph, and difference between two segments measures the minimum weight edge connecting the two. Here, edge weight function  $w((v_i, v_j))$  is defined as the absolute difference between intensity values of neighboring pixels, as in (76),

$$w((v_i, v_j)) = |I(p_i) - I(p_j)|. \quad 76$$

where  $i, j$  are the pixel numbers,  $I$  is the intensity value of pixel  $i$  and  $j$ . Generally, a Gaussian filter with variance  $\sigma$  is used to smooth the image before computing the edge weights.

Internal difference of a superpixel component is defined as the largest weight in the minimum spanning tree of the component,  $MST(C, E)$ . It can be written as:

$$Int(C) = \max_{e \in MST(C, E)} w(e) \quad 77$$

Difference between the components can be written as:

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w((v_i, v_j)) \quad 78$$

where  $C_1$  and  $C_2$  are different superpixel segments,  $MST$  is the minimum spanning tree of graph  $G$ , and  $w((v_i, v_j))$  is the absolute intensity difference between pixels  $i$  and  $j$ .

Minimum internal difference  $MInt(C_1, C_2)$  is defined as:

$$MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2)) \quad 79$$

where  $\tau$  is a threshold function.  $\tau$  is proportional to the inverse of the size of the superpixel,

$$\tau(C) = k/|C| \quad 80$$

where  $|C|$  is the size of a pixel and  $k$  is a constant.

A pairwise region comparison predicate is needed to determine if a boundary exist between a pair of components. If there exist(s) pair(s) of superpixels that have no evidence of boundary between them after segmentation  $S$ , such segmentation is defined as “too fine”. If there exists a proper refinement of segmentation  $S$  that is “not too fine”, it is defined as “too coarse”. The desired superpixel segmentation result follows “neither-too-coarse, nor-too-fine” rule. The pairwise region comparison predicate between two superpixels  $D(C_1, C_2)$  is defined as the decision criterion for segmentation given  $MInt(C_1, C_2)$ :

$$D(C_1, C_2) = \begin{cases} \text{true, if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{false, otherwise} \end{cases}. \quad 81$$

Low weights should be put on edges connecting pixels within the same superpixel, and higher weights should be put on edges between pixels with different superpixels. If predicate  $D(C_1, C_2)$  returns true, that means the difference between superpixels is larger than the minimum internal difference of the superpixel, there must exist a boundary between superpixels  $C_1, C_2$ ; if predicate  $D(C_1, C_2)$  returns false, then there is no boundary between the two, which means components  $C_1$  and  $C_2$  are actually and effectively the same superpixel.

The pseudo code for the graph-based superpixel algorithm is:

---

#### Algorithm 13: Graph-based superpixel segmentation algorithm

---

```

1 Calculating edge weight for every edge in the graph using (76)
2 Sort edges E into  $\pi = (o_1, o_2, \dots o_M)$  by non-decreasing edge weight
3 for  $m \leftarrow 1$  to number of edges  $M$  do
4   if  $C_i^{m-1} \neq C_j^{m-1}$  and  $w(o_m) \leq MInt(C_i^{m-1}, C_j^{m-1})$  then
5      $S^i = \text{Merge } C_i^{m-1} \text{ and } C_j^{m-1}$ 
6   else then
7      $S^i = S^{i-1}$ 
8   endif
9 endfor
10 Return segmentation result  $S=S^M$ 
```

---

The running time of the graph-based superpixel segmentation algorithm is  $O(N \log N)$ , where  $N$  is the number of pixels in the image. The algorithm returns superpixel labels for each pixel in the image, thus the nearest neighbors within the same superpixel class can be collected and used for BCM QP and MH approaches, similar to section 7.2.

### 8.3. Entropy Rate Superpixel Model

A random walk consists of a succession of random steps, usually in Markov chains or Markov processes (Pearson, 1905). Entropy is defined as the uncertainty of a random variable  $X$ , as shown in (82),

$$H(X) = -\sum_{x \in X} p_X(x) \log p_X(x). \quad 82$$

Entropy rate based superpixel method (Liu, et al., 2011) represents the hyperspectral image as undirected graph  $G = (V, E)$ . In this algorithm, entropy rate of the random walk on the constructed graph is used as a criterion to obtain “compact and homogeneous clusters” of an input image. Entropy rate is defined as the uncertainty of a stochastic process  $\mathbf{X} = \{X_t | t \in T\}$  where  $T$  is an index set. For a discrete random process, the entropy rate is defined as (83), which measures the uncertainty of the random process after past observations,

$$H(\mathbf{X}) = \lim_{t \rightarrow \infty} H(X_t | X_{t-1}, X_{t-2}, \dots, X_1). \quad 83$$

The entropy rate of the random walk is computed using

$$\begin{aligned} H(\mathbf{X}) &= H(X_2 | X_1) = \sum_i u_i H(X_2 | X_1 = v_i) = -\sum_i u_i \sum_j p_{i,j} \log p_{i,j} = \\ &= -\sum_i \frac{w_i}{w_T} \sum_j \frac{w_{i,j}}{w_i} \log \frac{w_{i,j}}{w_i} = \sum_i \sum_j \frac{w_{i,j}}{w_T} \log \frac{w_{i,j}}{w_T} + \sum_j \frac{w_i}{w_T} \log \frac{w_i}{w_T} \end{aligned} \quad 84$$

where  $w_i = \sum_{k:e_{i,k} \in E} w_{i,k}$  is the sum of incident weights of the vertex  $v_i$ ,  $w_{i,k}$  is edge weights denoting the pairwise similarities,  $w_T = \sum_{i=1}^{|V|} w_i$  is the normalization constant,  $\mu$  is the stationary distribution given by

$$\mu = (\mu_1, \mu_2, \dots, \mu_{|V|})^T = \left( \frac{w_1}{w_T}, \frac{w_2}{w_T}, \dots, \frac{w_{|V|}}{w_T} \right)^T, \quad 85$$

$w_1/w_T$  is the transition probability  $p_{i,j}$  defined in a random walk model (Thomas, et al., 1991).

The set functions for the transition probabilities  $p_{i,j}$  is given as:

$$p_{i,j}(A) = \begin{cases} w_{i,j}/w_i, & \text{if } i \neq j \text{ and } e_{i,j} \in A \\ 0, & \text{if } i \neq j \text{ and } e_{i,j} \notin A \\ 1 - \left( \sum_{j: e_{i,j} \in A} w_{i,j} \right)/w_i, & \text{if } i = j \end{cases} \quad 86$$

Therefore, the entropy rate of the random walk on graph  $G = (V, A)$  can be written as a set function:

$$H(A) = -\sum_i \mu_i \sum_j p_{i,j}(A) \log p_{i,j}(A). \quad 87$$

A balancing function is added to encourage that the clusters after segmentation have similar size. Let  $A$  be the selected edge set,  $N_A$  is the number of connected components in the graph, and  $Z_A(i)$  is the distribution of the cluster membership for the  $i^{th}$  component.  $Z_A$  is equal to the ratio of partitioning and the full volume formed by the components, as shown in (88),

$$p_{z_A}(i) = \frac{|S_i|}{|V|}, i = \{1, \dots, N_A\} \quad 88$$

The balancing term is then given by

$$B(A) = H(Z_A) - N_A = -\sum_I p_{z_A}(i) \log(p_{z_A}(i)) - N_A. \quad 89$$

$H(Z_A)$  term favors clusters with similar sizes while  $N_A$  term favors fewer number of clusters. The balancing term will therefore provide a balanced preference between similar sizes of clusters and number of clusters.

The optimal segmentation is achieved when the following objective function is maximized:

$$J_E = H(A) + \lambda B(A) = H(A) + \lambda(H(Z_A) - N_A) = -\sum_i \mu_i \sum_j p_{i,j}(A) \log p_{i,j}(A) - \lambda \sum_I p_{z_A}(i) \log(p_{z_A}(i)) - \lambda N_A, \quad 90$$

with constraints

$$A \subseteq E \text{ and } N_A \geq K. \quad 91$$

$\lambda \geq 0$  is a weighting term,  $K$  is the number of clusters.

The objective function is maximized using a greedy heuristic algorithm (Nemhauser, et al., 1978). The goal of the greedy algorithm is to search for the edge set  $A$  that maximizes the above objective function. The algorithm starts with  $A = \emptyset$ . At each iteration, the edge that yields the largest gain is added to the set. The algorithm stops when the number of connected components reaches a pre-set number  $K$ , i.e. when  $N_A = K$ . The pseudo code for the greedy algorithm for maximizing the superpixel objective function is shown in Algorithm 14.

---

**Algorithm 14: Entropy rate superpixel segmentation algorithm: greedy algorithm**

---

```

1 Input: undirected graph  $G = (V, E)$ ; number of clusters  $K$ ; weighting term  $\lambda$ 
2 Initialize  $A \leftarrow \emptyset$ , temporary set  $U \leftarrow E$ 
3 while  $U \neq \emptyset$  do
4      $\hat{a} \leftarrow \arg \max J_E$ 
5     if  $A \cup \{\hat{a}\} \in I$ 
6          $A \leftarrow A \cup \{\hat{a}\}$ 
7     endif
8      $U \leftarrow U - \{\hat{a}\}$ 
9 endfor
10 Return segmentation result  $A$ 

```

---

Entropy rate based superpixel method is able to obtain compact, homogeneous and balanced superpixel segmentation results. The algorithm returns edge set  $A$  for each clusters. By examining the vertices included in the edge set, we can obtain the cluster information of each pixel in the image. The nearest neighbors within the same superpixel class can then be collected and used for BCM QP and MH approaches, similar to section 7.2.

#### 8.4. QP and MH Sampling Approaches with Beta Distribution

The quadratic programming and Metropolis-Hastings sampling approaches follows similar steps as with BCM-spectral model in chapter 5 and 6, except that the spatial neighbors of each pixel here is identified using the clustering results from graph-based and entropy rate-based superpixel segmentation algorithms. The pseudo code is similar to Algorithm 11 and Algorithm 12.

## 9. HYPERSPECTRAL UNMIXING BASED ON SPATIAL K-MEANS WITH BETA DISTRIBUTION

### 9.1. Model Description

BCM-Spatial- $K$ -means unmixing approach mimics BCM-Spectral approach except for the identification of spatial neighbors. The  $K$ -means algorithm is applied to cluster the input image data with location information appended and scaled. The scaling parameter depends on the importance to be placed on the location information versus the spectral values. The image data with scaled location information can be written as (101),

$$\hat{x}_L = [x_1 \quad x_2 \quad \cdots \quad x_D \quad s \cdot row_i \quad s \cdot col_i]^T \quad 92$$

where  $D$  is the total number of bands,  $row_i$  and  $col_i$  are the location information (row and column index values) for pixel  $x_i$ , and  $s$  is the scaling parameter.

After the image data is clustered, the nearest spatial neighbors of a center pixel  $i$  are identified. Using the spatial neighbors, quadratic programming and Metropolis-Hastings sampling algorithms are identical to those described in Chapters 5 and 6.

## 10. DATA SETS AND RESULTS

### 10.1. Data Sets

A two-dimensional toy data example, simulated hyperspectral image data examples and real image data sets were used to compare the performance of quadratic programming unmixing with standard endmember spectral, unmixing assuming the Normal Compositional Model using a sampling approach, unmixing assuming the Beta Compositional Model using a quadratic programming and a sampling approach, and spatial-spectral unmixing assuming the Beta Compositional Model using a quadratic programming and a sampling approach.

#### 10.1.1. Toy Data

A two-dimensional, three-endmember toy data was generated initially to conduct the unmixing algorithms. In this example, two sets of toydata were generated: Gaussian toy data and Beta toy data.

For the Gaussian toy data, each endmember is represented with a Gaussian distribution with “true” mean values of (0.3 0.3), (0.5 0.7) and (0.7 0.3). Around each endmember mean, a diagonal covariance with all values equal to 0.003 is added. In this two-dimensional case, the resulting simplex is a triangle with the above mean values as vertices. The values are chosen in order to make sure that the data points randomly generated are enclosed in the [0, 1] region in both bands.

For the Beta toy data, the endmembers are assumed to be distributed according to Beta distributions and the parameters  $\alpha$  and  $\beta$  of the Beta distribution are selected to match the mean and variance of the Gaussian toydata.

The transformation of Beta parameters and Gaussian parameters satisfies the following steps. Beta distribution can be written as

$$\mathcal{B}(x; \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1} (1-x)^{\beta-1}, \quad 93$$

where  $\Gamma(z)$  is the gamma function

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt. \quad 94$$

The mean and variance of Beta distribution are known as:

$$E(X) = \frac{\alpha}{\alpha+\beta}, \quad 95$$

$$var(X) = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}. \quad 96$$

Let  $E(X) = \mu$ ,  $var(X) = \sigma^2$ , the parameters  $\alpha$  and  $\beta$  can be computed using

$$\alpha = \left( \frac{1-\mu}{\sigma^2} - \frac{1}{\mu} \right) \mu^2 \text{ and} \quad 97$$

$$\beta = \alpha \left( \frac{1}{\mu} - 1 \right). \quad 98$$

The corresponding  $\alpha$  and  $\beta$  parameters for each of the endmembers calculated from the above Gaussian mean and covariance values are listed in Table 1.

Table 1 Beta toydata endmember parameter settings

	Endmember 1		Endmember 2		Endmember 3	
	Band 1	Band 2	Band 1	Band 2	Band 1	Band 2
$\alpha$	20.7000	48.3000	41.1667	41.1667	48.3000	20.7000
$\beta$	20.7000	48.3000	48.3000	20.7000	20.7000	48.3000

$N$  data points were generated in the above two ways where  $N$  equals to 13 to 500 with an increment of 20. For each data point, proportions are first generated according to a Dirichlet distribution with all parameters set to 1 in order to ensure that the data is uniformly distributed inside the simplex. Then, the endmember samples are randomly generated according to either Gaussian with the above mean and covariance (in Gaussian toy data), or Beta distribution with the above alpha and beta values (in Beta toy data). Zero-mean independent Gaussian random noise with a variance of 0.001 is added to both toy data and the resulting values are recorded and plotted as the final toy data set. Figure 10-1 shows the toy data set when  $N = 13$  and 500, respectively, for both Gaussian toy data and Beta toy data.

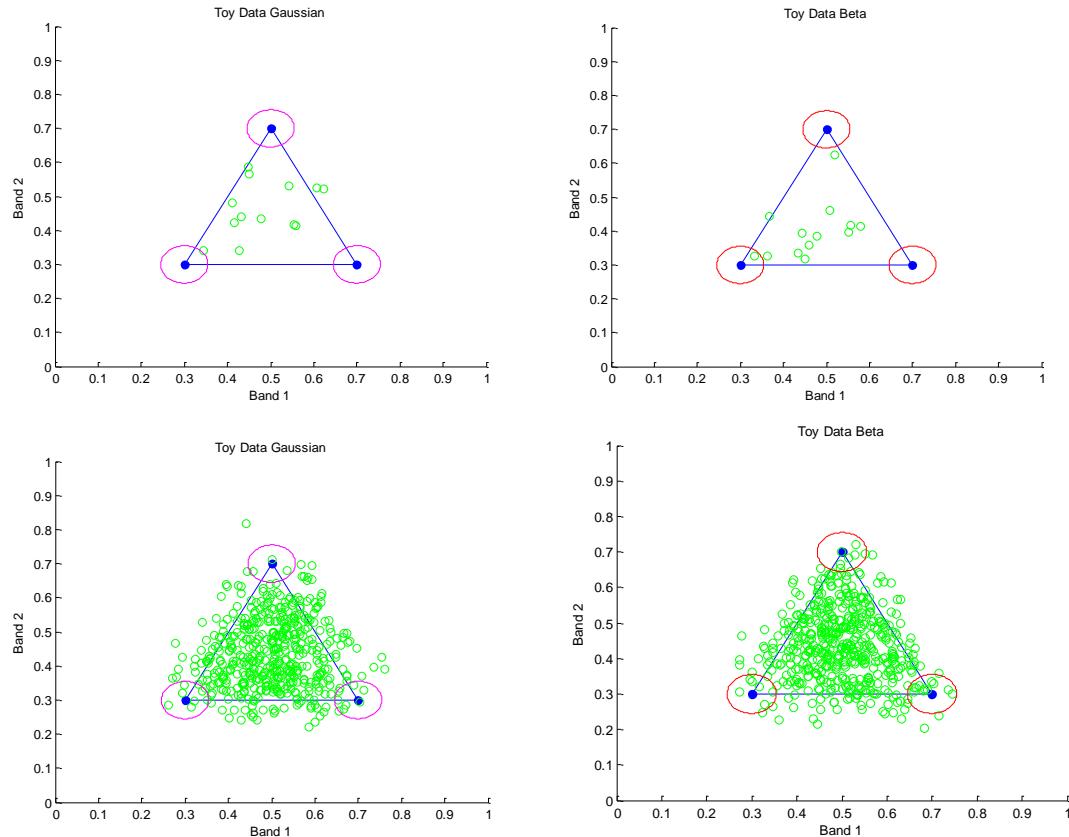


Figure 10-1 Toy data plots for two-dimensional, three-endmember Gaussian toy data and Beta toy data. Green circles: data points. Blue dots: The true mean values of endmembers. Red circles: covariance of endmember distributions. Top:  $N=13$ . Bottom:  $N=500$ . Left: Gaussian toy data. Right: Beta toy data.

### 10.1.2. Simulated Hyperspectral Image Dataset 1

Two simulated hyperspectral image (HSI) data sets are generated. The first simulated HSI data set consists of three endmembers selected from ASTER spectral library: Grass, Water and Soil. In this data set, two sets of simulated HSI data were generated: Gaussian simulated HSI data and Beta simulated HSI data. The endmember spectra have 191 bands from  $0.42\mu\text{m}$  to  $0.80\mu\text{m}$  wavelengths.

The endmember spectra from the spectral library are plotted in Figure 10-2.

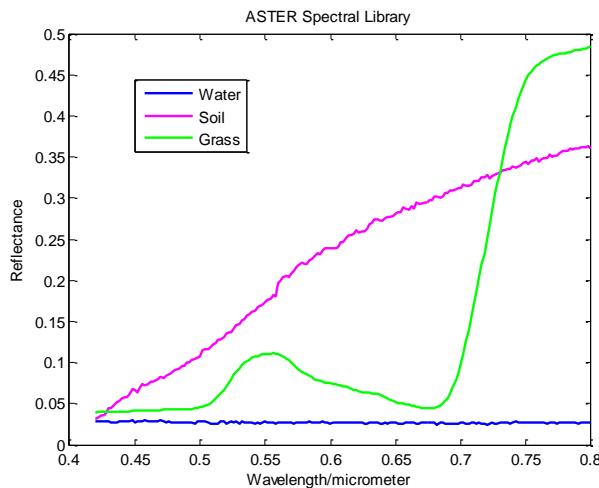


Figure 10-2 Endmember plot of reflectance versus wavelength from ASTER spectral library. Green: Grass. Blue: Water. Yellow: Soil. The wavelength measures from  $0.42\mu\text{m}$  to  $0.80\mu\text{m}$ .

Similar to toy data in Section 10.1.1, a diagonal covariance of 0.0003 was put on endmembers centered on the Grass, Water and Soil endmember values from spectral library. After sampling the proportions and endmembers, a zero-mean Gaussian noise of variance 0.001 was added to the data. The covariance and noise values are chosen to ensure that the data points generated are within the [0, 1] range.

In order to make an effective comparison between the Gaussian and Beta unmixing methods, a skewed Beta distribution is manufactured specifically for the

simulated HSI data. Beta parameters were selected such that the mode (the peak of the distribution) and mean of the Beta distribution are pushed apart such that the value that appears most often in the Beta distribution is far from the mean value. In this case, the Beta Compositional Model is expected to do a better job than the NCM since the mean and the mode of Gaussian distributions are the same.

If the alpha parameter in a Beta distribution is less than 1, an example of the basic shape of the beta is shown in Figure 10-3. However, in this experiment, it is desired to make the beta distribute shape like a bell curve as in Gaussian in order to make the two distributions comparable. As we can see from Equations (63) and (64), the closer the alpha and beta parameters are to 1 ( $\alpha, \beta > 1$ ), the larger the difference between mean and mode. Therefore, in this experiment, the alpha parameters in all Beta distributions across all bands for all endmembers are set to be 1.0001. Figure 10-4 shows a beta distribution plot with  $\alpha = 1.0001$  and  $\beta = 3.0000$ . The blue line in the figure plots the mean of the Beta distribution and the dashed blue line marks the mode of the Beta distribution.

In the unmixing experiments,  $\alpha$  is set to 1.0001. Parameter  $\beta$  is then adjusted according to transformation Equations (95) and (96) to make mean value of the Beta distribution match each and every band of the endmember spectra of Grass, Soil and Water from the spectral library.

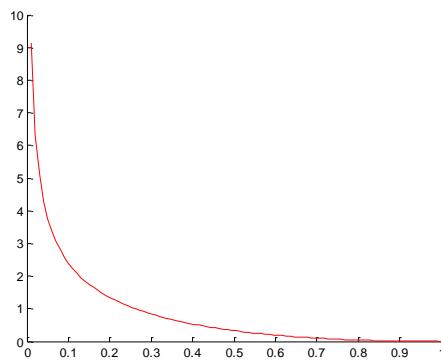


Figure 10-3 Beta distribution plot when alpha is less than 1. Parameters alpha=0.5000, Beta = 3.0000.

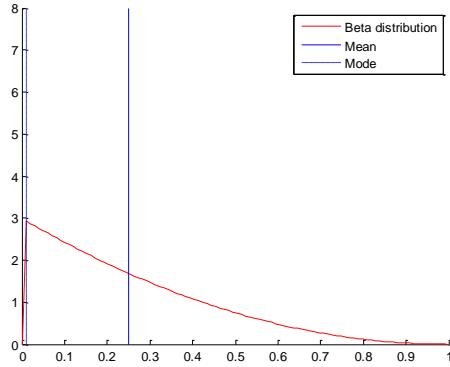


Figure 10-4 Skewed Beta distribution plot. Peak of red line: Mode value of Beta. Value indicated by blue line: Mean value of Beta. Parameters alpha=1.0001, Beta = 3.0000.

#### 10.1.3. Simulated Hyperspectral Image Dataset 2

The second set of simulated HSI data is constructed from four known endmembers: Asphalt, Yellow Curb, Grass and Oak Leaves. The endmember spectra were measured using a handheld ASD spectrometer in Long Beach, MS. The wavelengths range from 400 to 900 nm. The endmember signatures are shown in Figure 10-5. Note that for Grass and Oak Leaves, the basic shapes of the endmember spectral signature are similar, but their variances are different.

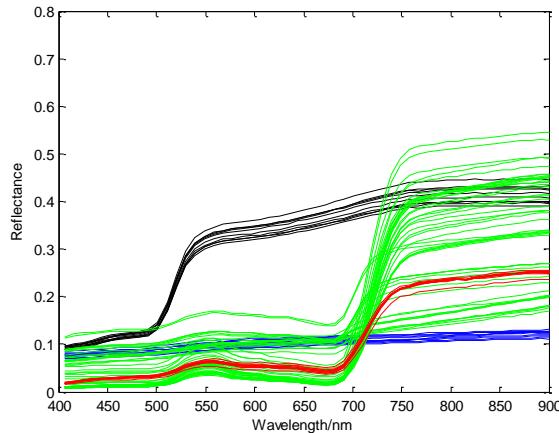


Figure 10-5 Four Endmembers Spectra measured using a handheld ASD spectrometer for Simulated Dataset 2. Blue: Asphalt, Black: Yellow Curb, Red: Grass, Green: Oak Leaves

The simulated data consists of  $100 \times 100$  pixels. In each of the four corners, a square of  $47 \times 47$  pixels is constructed from only one of the four endmembers. The center of  $6 \times 6$  pixels consists of all four endmembers with randomly generated proportions from the Dirichlet distribution. Between every two endmembers, there are four  $47 \times 3$  mixed pixels, which consist of only the nearby two endmembers with a gradient proportions for each  $1 \times 6$  set of pixels on each endmember generated from a symmetric Dirichlet distribution. The  $\alpha$  parameter for the Dirichlet distribution is set to be between 0.1 to 10, with an increment of  $s = \frac{(10 - 0.1)}{(47 - 1)}$ . In Matlab,  $\alpha$  can be written as:

$$\alpha = [0.1: s: 10], \text{ where } s = \frac{(10 - 0.1)}{(47 - 1)}. \quad 99$$

Gaussian noise with variance 0.001 ( $SNR_{dB} = 19.89dB$ ) is added to all pixels.

To test the robustness of the unmixing algorithms, a simulated data set with high noise is generated. All other settings remain the same except for the variance of Gaussian noise, which increases to 0.15 ( $SNR_{dB} = 7.26dB$ ). After adding noise, the data point values of the bottom two endmember blocks (Oak Leaves and Grass) are very similar, only the variances are different. The RGB image of simulated data with low noise and heavy noise are shown in Figure 10-7.

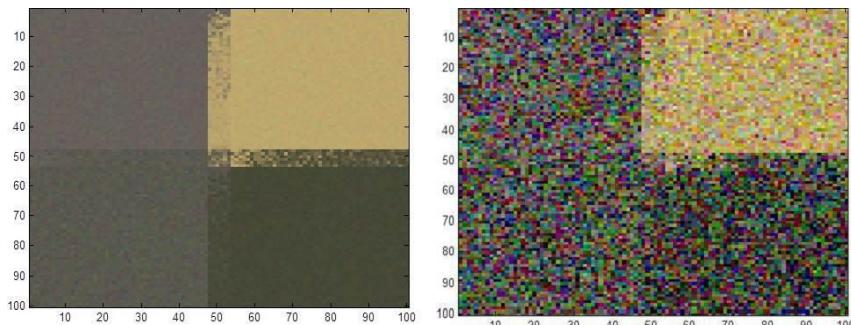


Figure 10-6 RGB image of simulated data with low noise and heavy noise. Top left: Asphalt. Top right: Yellow Curb. Bottom left: Oak Leaves. Bottom right: Grass.

#### 10.1.4. Real Hyperspectral Image Dataset 1

Two real hyperspectral image data taken over Pavia University and Long Beach, MS are used in the experiments. Pavia University (PaviaU) hyperspectral image scene is acquired by the ROSIS sensor during a flight campaign over Pavia, northern Italy. The scene consists of  $610 \times 340$  pixels across 103 bands from wavelengths  $0.43 \mu\text{m}$  to  $0.86\mu\text{m}$ . The ground truth image of PaviaU data consists of 9 classes of materials, as shown in Table 2. Figure 10-7 shows images of PaviaU data. On the left is the ground truth of the PaviaU data, and on the right is a false color image of paviaU data reconstructed from band 66, 33, 13 as the R, G and B bands.

In our experiments, three subsets of PaviaU data are taken and ran on unmixing algorithms.

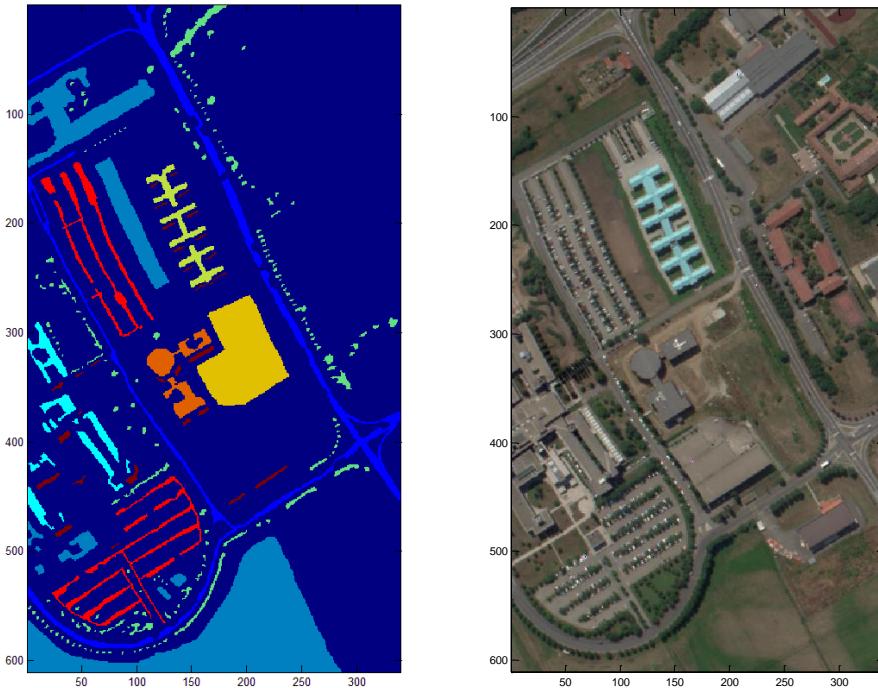


Figure 10-7 PaviaU hyperspectral image data ground truth and RGB image.

Table 2 Groundtruth classes for the PaviaU scene

#	Class	Samples
1	Asphalt	6631
2	Meadows	18649
3	Gravel	2099
4	Trees	3064
5	Painted metal sheets	1345
6	Bare Soil	5029
7	Bitumen	1330
8	Self-Blocking Bricks	3682
9	Shadows	947

#### 10.1.5. Real Hyperspectral Image Dataset 2

Gulfport data set is a hyperspectral image collected over Long Beach, MS at the University of Southern Mississippi-Gulf Park campus. The wavelengths range from 400 to 900 nm. In our experiments, a subset of  $13 \times 19$  pixels is taken from the full imagery. The subset consists of four endmembers: Asphalt, Yellow Curb, Grass and Oak Leaves. This subset was chosen as the true endmember values (measured using an ASD hand-spectrometer) are known. Thus, the true beta endmember parameter values can be computed from the ASD-measured spectra. Figure 10-8 shows the RGB image of the Gulfport data reconstructed using band 26, 14 and 8 as the R, G and B bands.

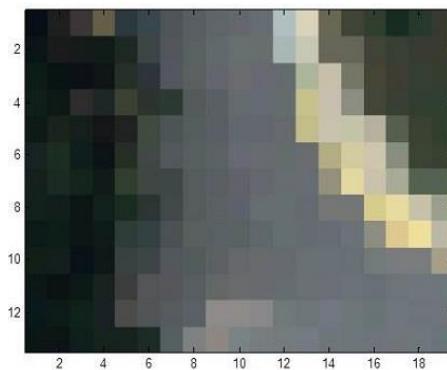


Figure 10-8 Gulfport hyperspectral image data RGB image. Left dark area: Oak Leaves. Middle gray area: Asphalt. Top right light yellow area: Yellow Curb. Top right corner: Grass.

## 10.2. Results

This section presents and compares the unmixing results of toy data, simulated data and real hyperspectral image data by unmixing algorithms described in Chapters 3 to 9 with QP and MH approaches under NCM and BCM models. Unmixing results are presented by proportion maps plotted based on proportion values estimated from the unmixing algorithms, and by comparison of proportion error values,  $PError$ , across different unmixing methods if true proportion values are known. Pixel-wise proportion error values are calculated as follows:

$$PError\_pixel_i = \|Ptrue_i - Pestimate_i\|_2$$

100

where  $i = 1, \dots, N$ ,  $N$  is the total number of pixels,  $Ptrue$  is the true proportion values,  $Pestimate$  is the estimated proportion values, and  $\|\cdot\|_2$  denotes the  $l^2$  norm between the two.

The proportion error mean values are then calculated using  $PError\_pixel_i$  values divided by the number of endmembers  $M$ , representing the proportion error across each pixel for each endmember. These values are reflected as the values before the “ $\pm$ ” sign in the following proportion error tables. The standard deviation values (the values after the “ $\pm$ ” sign) are the standard deviation of proportion error mean values across multiple runs.

### 10.2.1. Toy data

Fully Constrained Least-Square (FCLS) method (Heinz, et al., 2001), QP and MH approaches under NCM and BCM models are ran on the toy data. FCLS method is used as comparison as it is a simple and widely used unmixing algorithm which only uses the

mean of endmember spectra to perform unmixing and does not incorporate any variance from the data. Proportion error,  $PError$ , is defined as the norm between true proportion values and estimated proportion values. Table 3 lists the comparative results of proportion error,  $PError$ , when unmixing toy data using the Fully Constrained Least-Square (FCLS) method (Heinz, et al., 2001), Quadratic programming assuming the NCM, Metropolis-Hasting sampling assuming the NCM, Beta quadratic programming and Beta MH sampling methods.

Table 3  $PError$  when unmixing Toydata

<b>PError</b> <b>N=500</b>	<b>FCLS</b>	<b>Gaussian QP</b>	<b>Gaussian MH</b>	<b>Beta- Spectral QP</b>	<b>Beta- Spectral MH</b>
Gaussian Toydata	0.04± 2.15e-03	8.80e-03± 4.25e-04	8.48e-03± 4.18e-04	8.61e-03± 4.23e-04	8.60e-03± 4.24e-04
Beta Toydata	0.04± 2.18e-03	8.89e-03± 4.45e-04	8.56e-03± 4.36e-04	8.73e-03± 4.42e-04	8.72e-03± 4.43e-04

It shows that Beta quadratic programming and Beta sampling method yields low  $PError$  mean value and a low variance between experiments, and Beta Sampling method has the lowest  $PError$  mean value (as bolded in Table 3). Gaussian methods did a better job in unmixing Gaussian toy data than Beta methods and Beta methods did a better job in unmixing Beta toy data. This result makes sense because Gaussian methods assume Gaussian endmembers while Beta methods assume Beta endmembers, respectively. Also, Beta methods (Beta quadratic programming and Beta MH sampling) yield lower  $PError$  value than Gaussian methods overall.

### 10.2.2. Simulated HSI Dataset 1

Table 4  $PError$  when unmixing Simulated HSI Dataset 1

<b>PError N=500</b>	<b>FCLS</b>	<b>Gaussian QP</b>	<b>Gaussian MH</b>	<b>Beta QP</b>	<b>Beta MH</b>
Gaussian Simulated 1	1.97e-09± 4.99e-24	<b>1.97e-09± 2.91e-24</b>	<b>5.73e-06± 2.99e-07</b>	0.28± 4.43e-16	0.28± 1.76e-04
Beta Simulated 1	0.29± 7.25e-16	0.28± 7.81e-16	0.28± 1.27e-04	<b>1.38e-03± 2.40e-18</b>	<b>1.39e-03± 1.17e-05</b>

Table 4 lists the comparative results of proportion errors,  $PError$ , when unmixing simulated hyperspectral image data by FCLS, Gaussian sampling, Gaussian quadratic programming, Beta quadratic programming and Beta sampling. Similar to the unmixing result of toy data, Gaussian methods did a better job in unmixing Gaussian simulated HSI data and Beta methods did a better job in unmixing Beta simulated HSI data.

### 10.2.3. Simulated HSI Dataset 2

All unmixing methods, including FCLS, Gaussian sampling, Gaussian quadratic programming, Beta quadratic programming, Beta sampling, Beta-FLICM, Beta-Superpixel and Beta-Spatial- $K$ -means methods, are tested on simulated HSI dataset 2 with both low noises and high noises. In order to compare the proposed unmixing methods with one that does not consider endmember variability, FCLS method is also tested on the simulated HSI dataset 2. Here, “Superpixel 1” method refers to the graph-based superpixel segmentation algorithm discussed in Section 8.2, and “Superpixel 2”

method refers to the entropy rate superpixel segmentation algorithm discussed in Section 8.3.

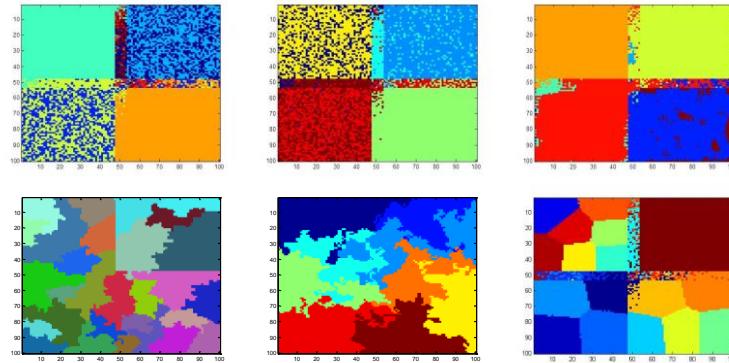


Figure 10-9 Clustering results of simulated dataset 2 by  $K$ -means, Fuzzy C-means, FLICM, Superpixel 1, Superpixel 2, and Spatial- $K$ -means.

Figure 10-9 shows the clustering results of simulated HSI dataset 2 by  $K$ -means, Fuzzy C-means, FLICM, Superpixel 1, Superpixel 2, and Spatial  $K$ -means clustering algorithms. The cluster number is set to 9 for  $K$ -means, Fuzzy C-means, FLICM, Superpixel 2, and Spatial- $K$ -means as to avoid empty clusters. In this experiment, Superpixel 2 unmixing algorithm takes the RGB image of the simulated data, where R, G and B bands correspond to bands 26, 14 and 8. The spatial- $K$ -means clustering result is desirable as the segments form in circular shapes and is distinguishable between different endmembers, achieving a good balance between spectral and spatial values of the data set.

Figure 10-10 and Figure 10-11 shows the unmixing results of simulated HSI dataset 2 by endmembers Asphalt, Yellow Curb, Oak Leaves and Grass, with low noise and high noise. As seen from Figure 10-9, Superpixel 1 clustering algorithm is not able to clearly distinguish between endmembers and yields a poor segmentation result. Thus, BCM-superpixel 1 unmixing results are excluded in the following experiments.

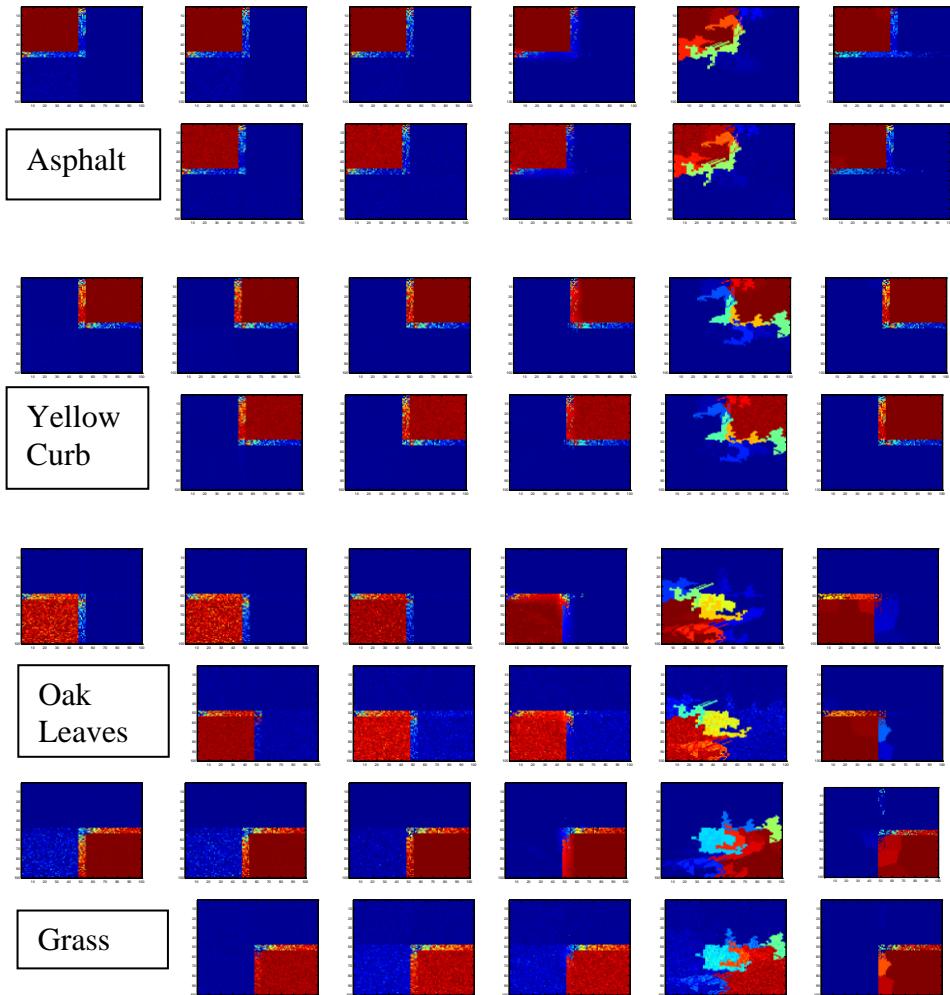


Figure 10-10 Unmixing results of Simulated data set 2 with low noise. Top to Bottom: Asphalt, Yellow Curb, Grass and Oak Leaves. Unmixing methods from left to right (The same method are in the same column. For example Gaussian QP is second column first line and Gaussian MH method is second column second line): FCLS, Gaussian quadratic programming, Gaussian sampling, Beta quadratic programming, Beta sampling, Beta-FLICM quadratic programming, Beta-FLICM sampling, Beta-Superpixel 2 quadratic programming, Beta- Superpixel 2 sampling, Beta-Spatial-K-means quadratic programming and Beta-Spatial-K-means sampling.

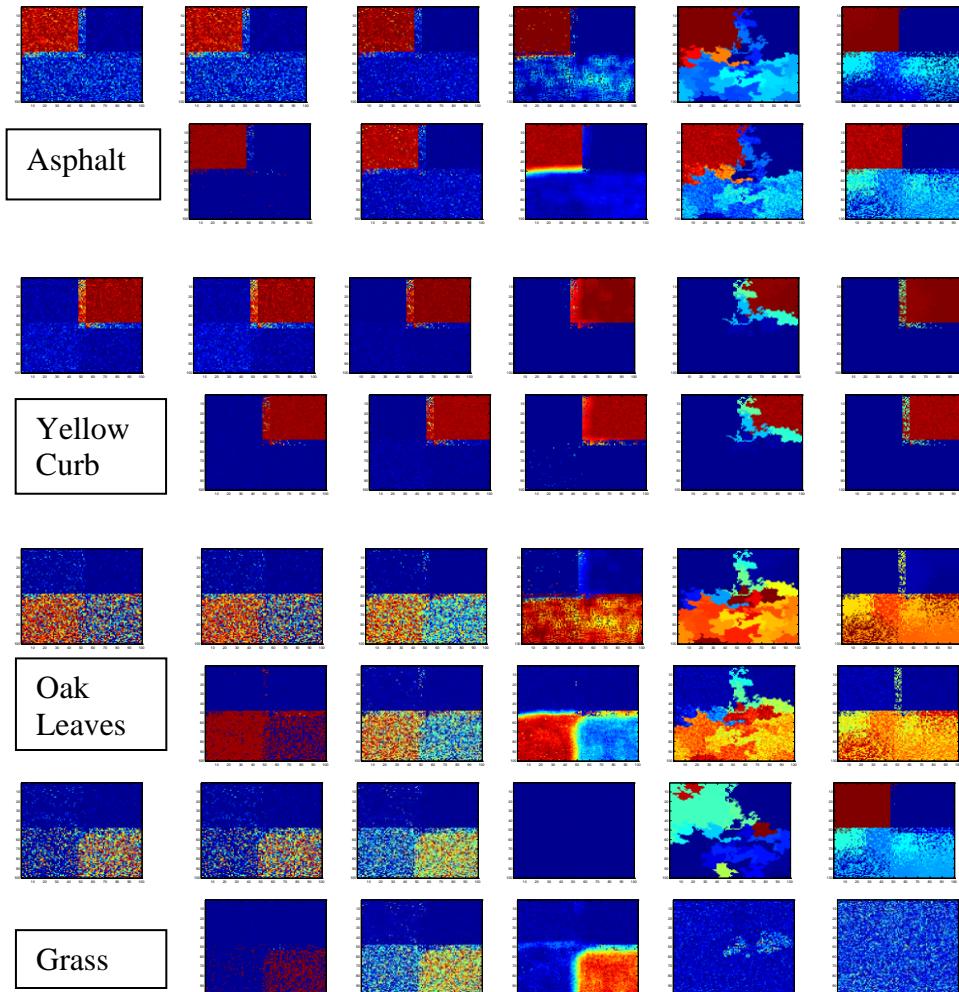


Figure 10-11 Unmixing results of Simulated data set 2 with high noise. Top to Bottom: Asphalt, Yellow Curb, Grass and Oak Leaves. Unmixing methods from left to right (The same method are in the same column. For example Gaussian QP is second column first line and Gaussian MH method is second column second line): FCLS, Gaussian quadratic programming, Gaussian sampling, Beta quadratic programming, Beta sampling, Beta-FLICM quadratic programming, Beta-FLICM sampling, Beta-Superpixel 1 quadratic programming, Beta-Superpixel 1 sampling, Beta-Superpixel 2 quadratic programming, Beta-Superpixel 2 sampling, Beta-Spatial-K-means quadratic programming and Beta-Spatial-K-means sampling.

Table 5 lists the comparative results of proportion errors,  $PError$ , when unmixing simulated hyperspectral image data by FCLS, NCM-QP, NCM-MH, BCM-Spectral (denoted as “Beta”)-QP, BCM-Spectral-MH, BCM-FLICM-QP, BCM-FLICM-MH, BCM-Superpixel 2-QP, BCM-Superpixel 2-MH, BCM-Spatial- $K$ -means QP and BCM-Spatial- $K$ -means MH methods.

Table 5  $PError$  when unmixing Simulated HSI Dataset 2 (Q: Quadratic programming. S: Sampling)

PError N = 100*100		FCLS	Gaussian	Beta	Beta- FLICM	Beta- Entropy rate Superpixel	Beta spatial $K$ means
Low Noise	Q	<b>0.01± 3.56e-18</b>	<b>8.70e-03± 1.78e-18</b>	<b>0.01± 1.34e-35</b>	<b>0.01± 1.69e-16</b>	0.03± 1.15e-09	<b>0.01± 2.30e-06</b>
	P		1.08e-02± 4.16e-03	0.01± 3.71e-05	8.90e-03± 7.86e-05	0.03± 5.00e-09	<b>1.02e-02± 1.07e-04</b>
High Noise	Q	0.06± 1.10e-35	0.06± 1.46e-17	<b>0.05± 1.48e-05</b>	0.03± 2.65e-03	0.15± 4.50e-06	0.12± 4.96e-03
	P		0.08± 2.09e-03	0.05± 1.39e-04	<b>0.03± 3.76e-04</b>	0.14± 2.36e-05	0.13± 1.05e-03

It can be seen from Table 5 that, when simulated dataset 2 has low noise, all methods did a good job in unmixing and producing the correct proportion maps for each endmember. Compared to Gaussian methods, Beta methods yielded a lower error in proportions. BCM-Superpixel algorithms produces proportion maps that are highly dependent on the superpixel segmentation result on the data (see Figure 10-9). The BCM-Superpixel algorithm is able to map the region where each endmember has high proportion values, but the edges of each region are non-linear and blurred. Between the

two superpixel results, entropy rate superpixel gives a better result (the 5<sup>th</sup> and 6<sup>th</sup> columns of Figure 10-10). The BCM-Spatial- $K$ -means unmixing algorithm yields the lowest proportion errors, as shown in Table 5.

When simulated dataset 2 has heavy noise, Beta methods, in general, still yielded lower proportion errors than Gaussian methods. FCLS method, which does not consider endmember variability, produced high proportion errors. Beta-spectral methods (Beta quadratic programming and Beta sampling, the 3<sup>th</sup> and 4<sup>th</sup> columns of Figure 10-11), which consider only spectral neighbors of pixels, did poorly in distinguishing grass and oak leaves endmembers. As indicated in Figure 10-5, grass and oak leaves have similar shapes of endmember spectral signature, but their variances are different. With heavy noise in simulated dataset 2, the spectral values of the bottom two blocks of the image are very similar. Therefore, if no spectral variance is considered, the unmixing results would do poorly. On the other hand, Beta-FLICM methods with spatial information successfully unmixed all four endmembers including grass and oak leaves. Beta-FLICM sampling method yielded the lowest proportion error.

#### 10.2.4. PaviaU Hyperspectral Image Data

##### 10.2.4.1. PaviaU Data Endmember Distribution

In order to investigate whether the Beta distribution is a good fit for endmember spectra, comparisons between hyperspectral data and their representation using the Beta distribution are made.

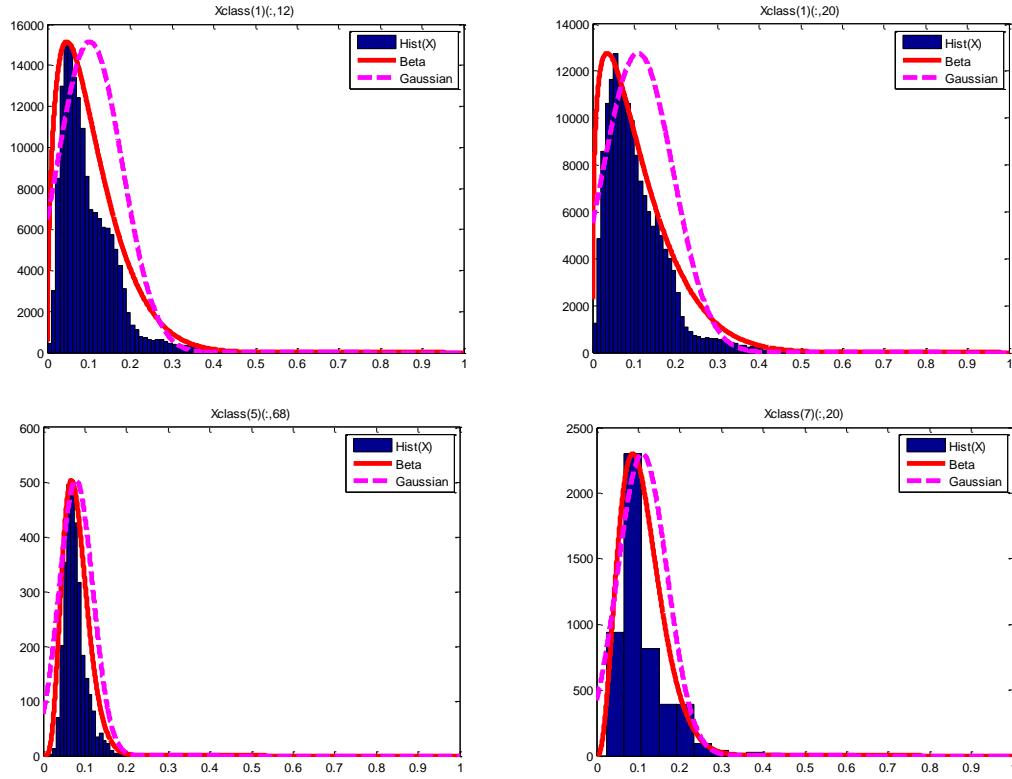


Figure 10-12 PaviaU hyperspectral image data endmember demonstration. Top left: Asphalt at wavelength  $0.48\mu\text{m}$ . Top right: Asphalt at wavelength  $0.51\mu\text{m}$ . Bottom left: Painted Metal Sheets at wavelength  $0.71\mu\text{m}$ . Bottom right: Bitumen at wavelength  $0.51\mu\text{m}$ .

Figure 10-12 shows the histogram of selected endmember and bands of PaviaU data. The red line plots the Beta approximation for the endmember values using a Maximum Likelihood approach. The green dashed line plots the Gaussian approximation of the endmember values using Maximum Likelihood. The endmember and band value is indicated in the title of each subplot. It can be observed from the plots that Beta distribution is able to better represent endmembers in the above data set.

Two quantitative methods are used to compare the results of Beta and Gaussian approximation to the sample data. The first method plots the histogram of the original data of the endmembers and marks the center and element values of each bin. Then, using Beta and Gaussian approximation, the element value (“y” axis value in the above figures)

of the approximated Beta and Gaussian distribution corresponding to the center of the bin (“ $x$ ” axis value in the above figures) can be recorded. The root mean square (RMS) between the true data element value and the approximated Beta and Gaussian value are calculated. If RMS of Beta distribution is smaller than RMS of Gaussian, the beta approximation fits the data better; and if RMS of Beta is larger, Gaussian distribution may better represent the data.

The second measurement method plots the histogram of the original data of the endmembers and uses Kullback-Leibler divergence to compare the Beta approximation and Gaussian approximation versus the original histogram. Kullback-Leibler divergence (Kullback, et al.) measures the difference between two probability distributions  $P$  and  $Q$ . Specifically, the Kullback–Leibler divergence of  $Q$  from  $P$ , denoted  $\text{KL}(P||Q)$ , calculates the average additional amount of information that is required for distribution Q to match distribution P, i.e. the information lost when  $Q$  is used to approximate .  $\text{KL}(P||Q)$  can be written as follows:

$$\begin{aligned}\text{KL}(p||q) &= - \int p(x) \ln q(x) dx - (- \int p(x) \ln p(x) dx) \\ &= - \int p(x) \ln \frac{q(x)}{p(x)} dx\end{aligned}\quad 101$$

Here, the “true” distribution values  $p(x)$  are of  $x$  ranging from 0 to 1 with an increment of 0.0001. Note that the  $x$  refers to the endmember values and the true distribution values are the histogram values for  $x$ . There are two approximated distributions, Beta and Gaussian. The differences between Beta approximation and Gaussian approximation versus the true histogram distribution of endmembers are calculated and a count number is recorded. If the difference between Beta approximation and the original data is smaller than that of Gaussian, Beta approximation is a better fit, count number *Count* pluses one. The algorithm outputs *Count* which measures the

number of times when Beta distribution is a better fit for the true endmember distribution than Gaussian distribution.

In (101), if  $q(x)$  is zero, the equation yields infinity values. In our implementation, two data processing methods are taken: (1) crop out all  $q(x)$  that is zero and use what's left of the data to measure  $KL$  distance (results corresponding to “Crop data” row in Table 6); (2) set the zero values to a very small near-zero number, such as  $10^{-100}$  (results corresponding to “Set data=0 to a small nonzero number” row in Table 6). It ensures that the calculation won't give out error, yet matches the original data as close as possible. In order to define the appropriate “small near-zero number”, a plot is made of the count number ratios of Beta and Gaussian versus  $10^{-m}$ . It can be seen from Figure 10-13 that after  $m$  equals to 3, the ratio is converged and therefore a number smaller than  $10^{-3}$  is good enough to substitute the zero values in the original approximations.

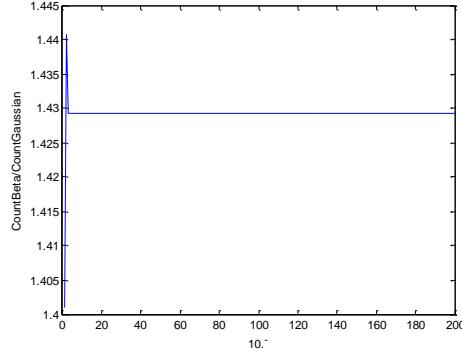


Figure 10-13 Illustration for substituting zero data for KL-Divergence method.

The total number of original endmember data points equals to 1030 (10 endmembers  $\times$  103 bands) and the corresponding RMS and KL-Divergence results are shown in Table 6. It can be seen from the table that RMS determines that Gaussian distribution fits the data slightly better than Beta distribution. However, since RMS breaks the distributions down to small discrete samples, it is more accurate to use KL-

Divergence as a measurement of the true difference between distributions. After using KL-Divergence and cropping the data (or setting data to  $10^{-100}$  when necessary, it gives out identical result), Beta distribution is shown to be matching the true endmember distribution 606 cases out of 1030, evidently better than Gaussian distribution (424 out of 1030). This table shows that Beta distribution is a better fit for the endmembers used in this experiment (endmembers included in this scene are: Asphalt, Meadow, Gravel, Trees, Painted Metal Sheets, Bare Soil, Bitumen and Self-Blocking Bricks).

Table 6 PaviaU hyperspectral image data endmember approximation

		distBeta distGaussian	<=	distBeta distGaussian	>	Total
RMS	based on centers	509		521		1030
		49.42%		50.58%		100%
KL- Divergence	Crop data	329		701		1030
		31.94%		68.06%		100%
KL- Divergence	Crop data	<b>606</b>		424		1030
		<b>58.83%</b>		41.17%		100%
	Set data=0 to a small nonzero number	606		424		1030
		58.83%		41.17%		100%

#### 10.2.4.2. PaviaU Data Unmixing Results

Three subsets of Pavia University data are taken as the source data for unmixing. Figure 10-14 shows the ground truth of the subset 1. This scene contains four materials: shadows, Bitumen, Bare Soil and Trees (Background is included as a separate material when unmixing).

Figure 10-15 shows the proportion maps of all methods for each endmembers. The sampling methods run up to 20,000 iterations and the image is pre-segmented into 100 clusters by Beta FLICM methods.

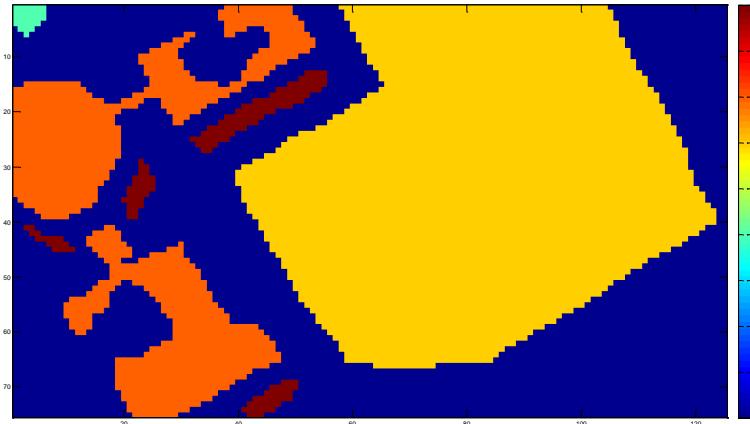
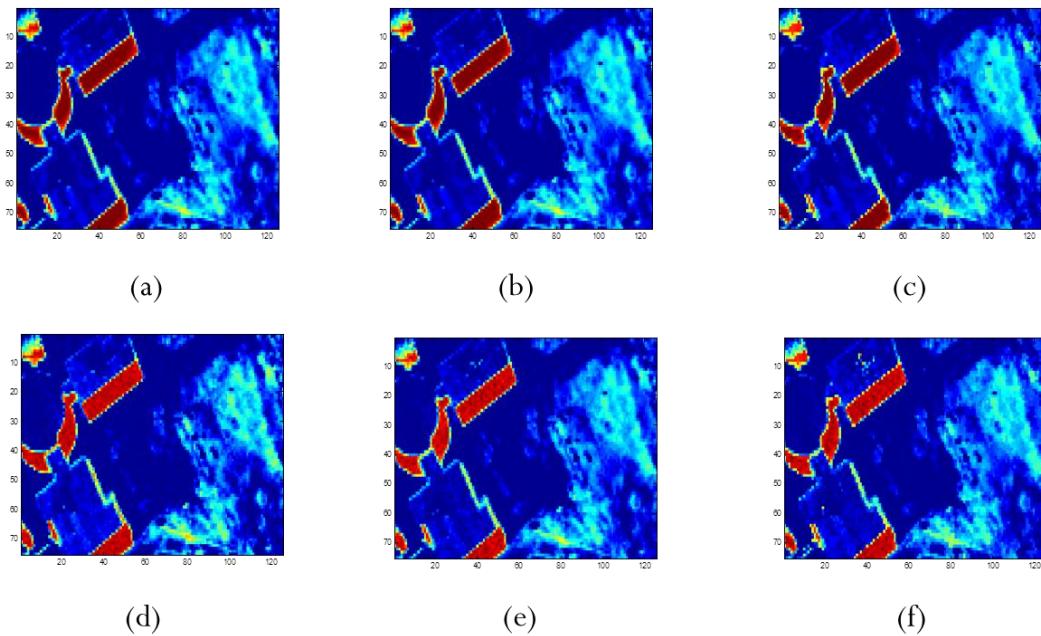
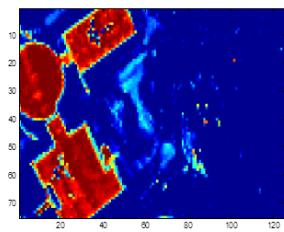
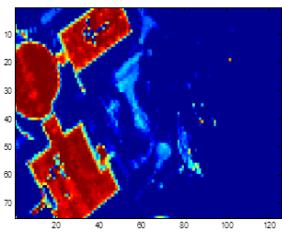


Figure 10-14 Ground Truth of PaviaU data subset 1. Endmember number=5.

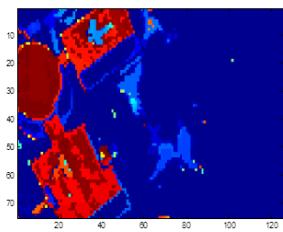




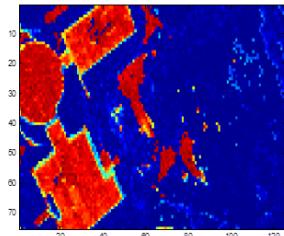
(a)



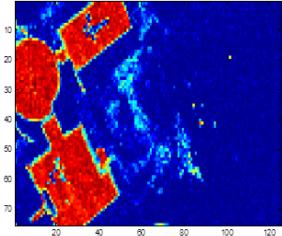
(b)



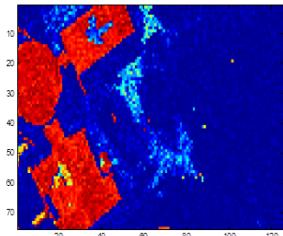
(c)



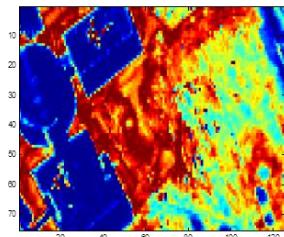
(d)



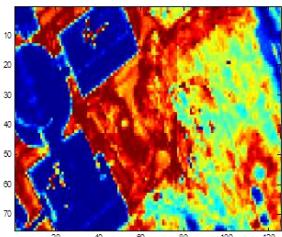
(e)



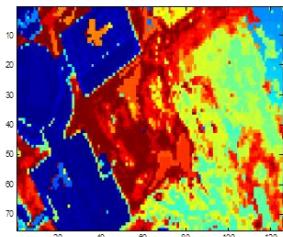
(f)



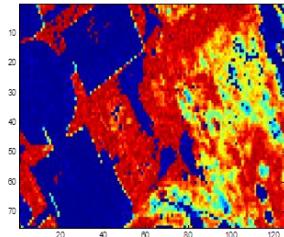
(a)



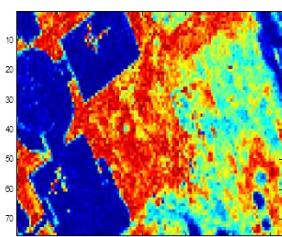
(b)



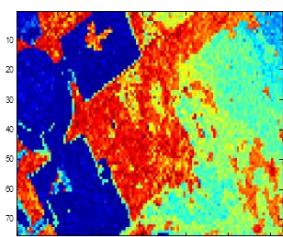
(c)



(d)



(e)



(f)

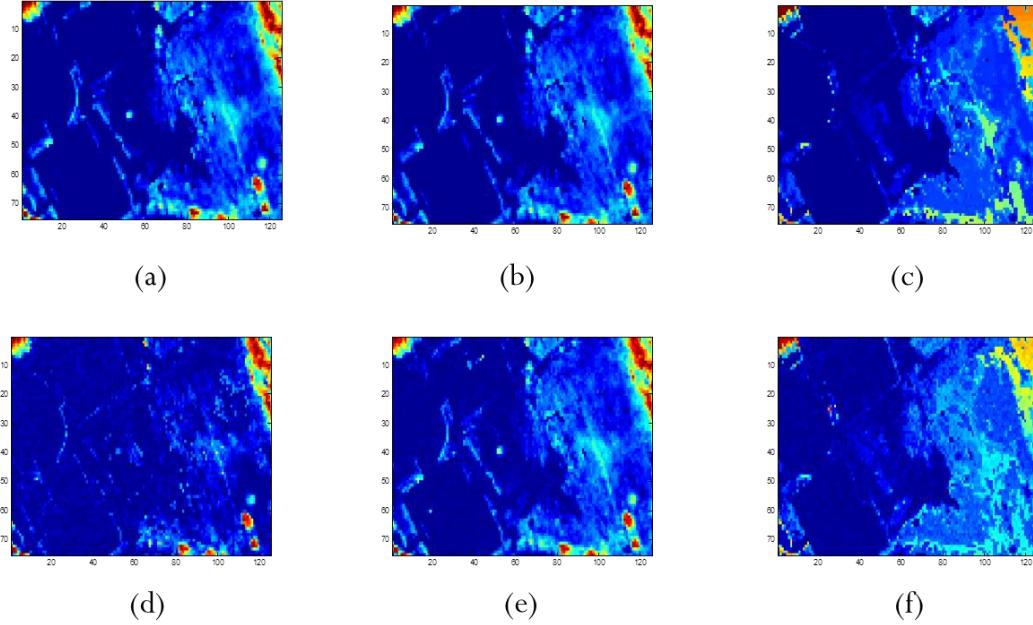


Figure 10-15 Proportion map of shadows, bitumen, bare soil and trees unmixing method on PaviaU data subset, from top to bottom. Each proportion map has images showing the unmixing result for (a) quadratic programming, (b) Beta quadratic programming, (c) Beta FLICM quadratic programming, (d) Gaussian sampling, (e) Beta sampling and (f) Beta FLICM sampling methods.

It can be seen from the proportion maps that the block of Bitumen is identified without “fading” using the proposed Beta-FLICM method as compared to the Gaussian unmixing methods and BCM method. The painting endmember is identified much clearer with an “L” shape. Besides, irrelevant endmember is weaker as compared to results before, meaning less noise. Especially with sampling methods, there is significant improvement in Beta-FLICM results compared to the original Gaussian sampling method. The soil endmember is clearly and correctly unmixed (deep red area), and on the bottom right corner, there are no background dots (deep blue). The reason for the improvements is inferred to be Beta-FLCIM method clusters the surrounding pixels as the same cluster as the center pixel, thus is more able to identify the same endmember in the spatial neighbor, i.e. if the material is “block-shaped”, meaning a large area of the scene consisting of the same endmember as compared to scattered materials. As for trees

endmember, the top right corner, in real-life-color image, appears to be a meadow area. Beta FLICM did not show this area to be deep red (tree) while previous methods did.

Figure 10-16 shows the ground truth of the subset 2. This scene contains four endmembers: shadows, meadows, painted metal sheets and self-blocking blocks (including background when unmixing). Figure 10-17 shows the proportion maps of all methods for each endmembers. The sampling methods run up to 20000 iterations and the image is pre-segmented into 100 clusters by Beta FLICM methods.

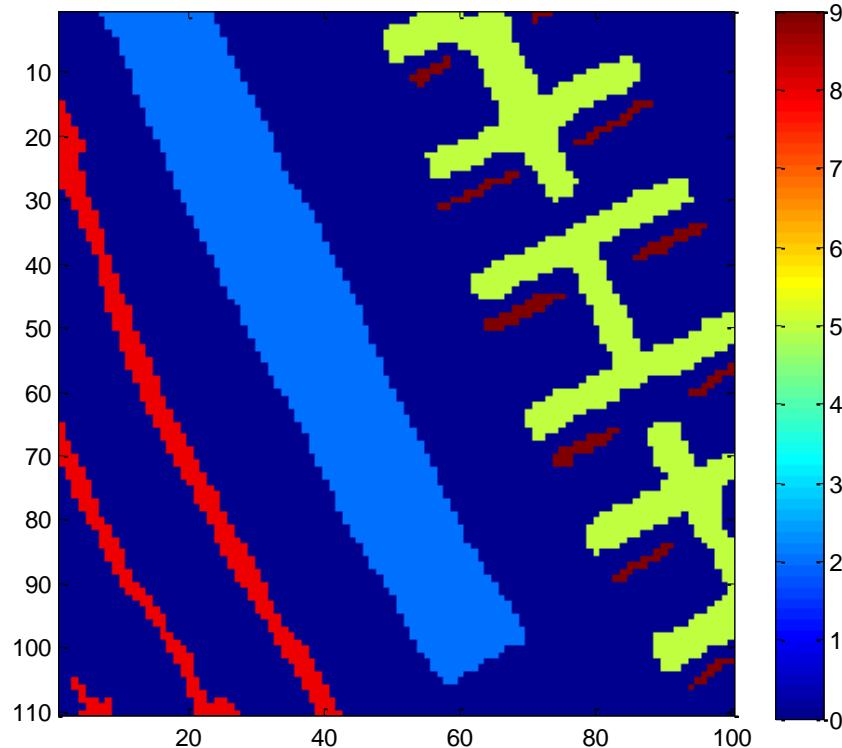
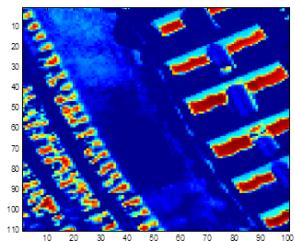
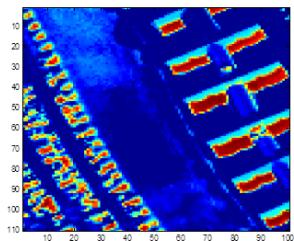


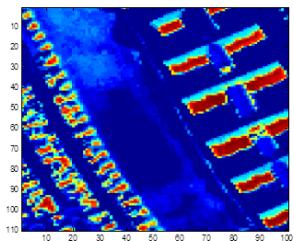
Figure 10-16 Ground Truth of PaviaU data subset 2. Endmember number=5.



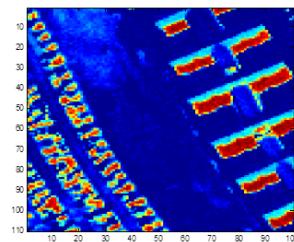
(a)



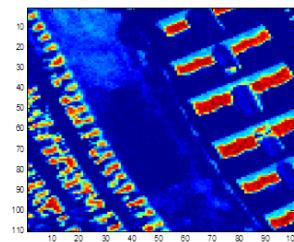
(b)



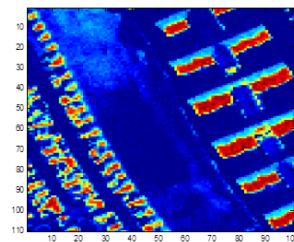
(c)



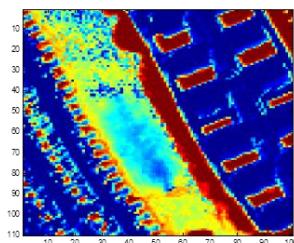
(d)



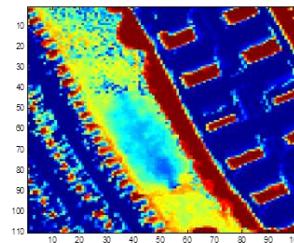
(e)



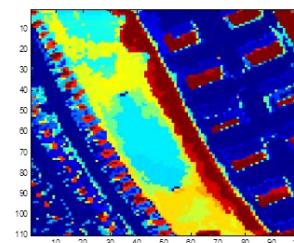
(f)



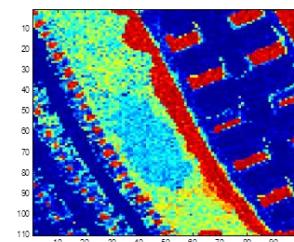
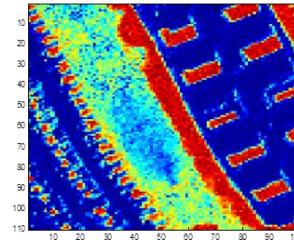
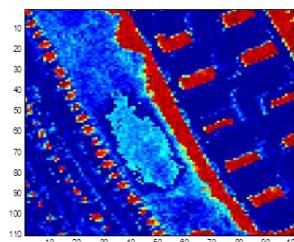
(a)

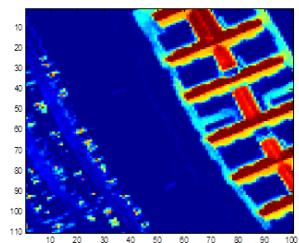


(b)

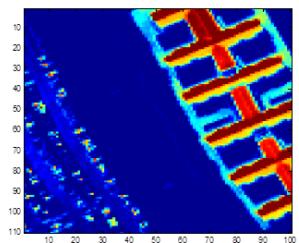


(c)

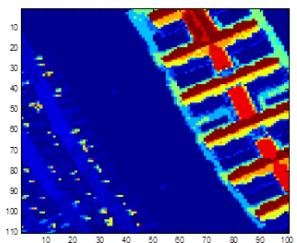




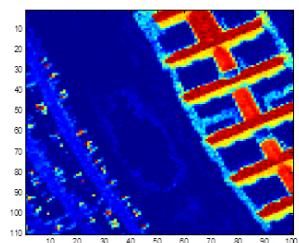
(a)



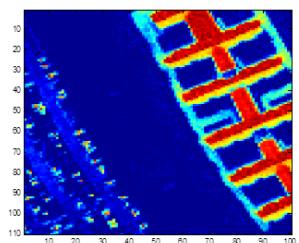
(b)



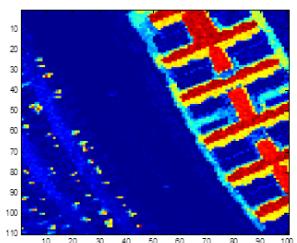
(c)



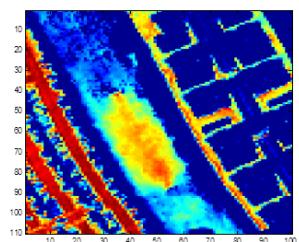
(d)



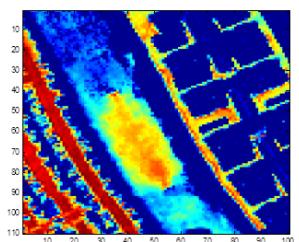
(e)



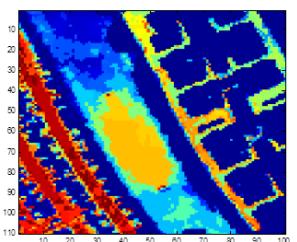
(f)



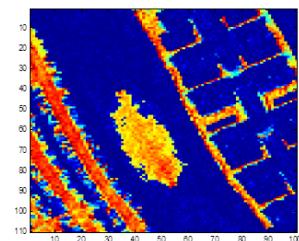
(a)



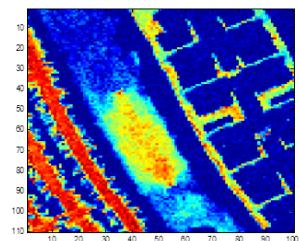
(b)



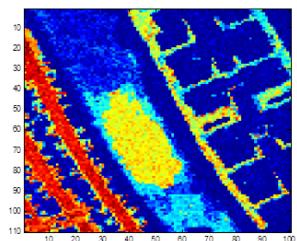
(c)



(d)



(e)



(f)

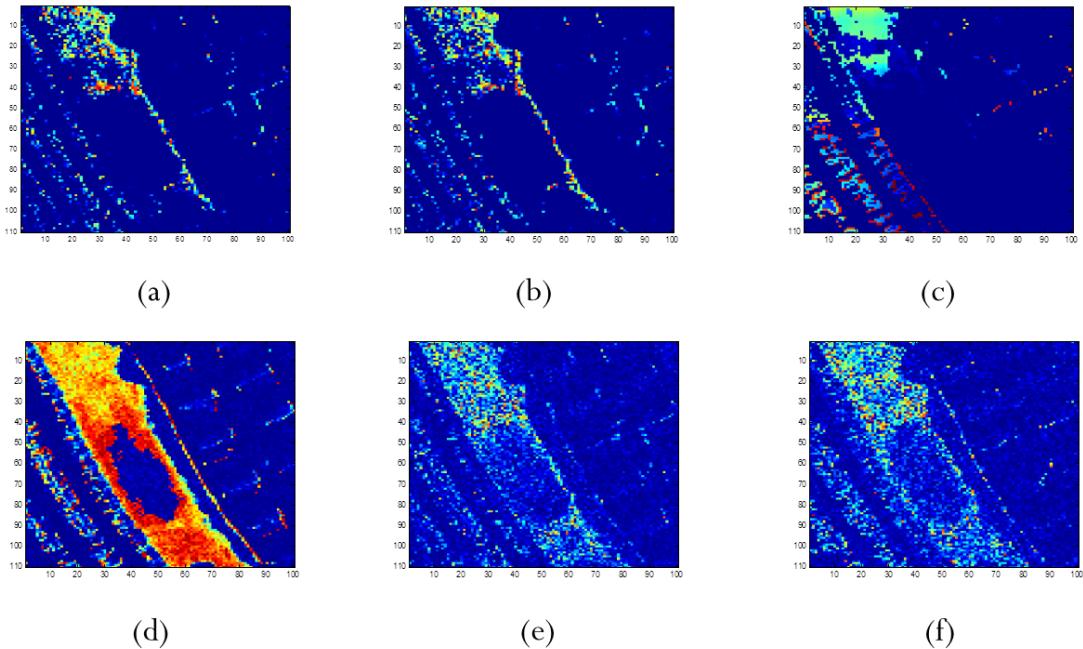


Figure 10-17 Proportion map of shadows, meadows, painted metal sheets, self-blocking blocks and background unmixing method on PaviaU data subset, from top to bottom. Each proportion map has images showing the unmixing result for (a) quadratic programming, (b) Beta quadratic programming, (c) Beta FLICM quadratic programming, (d) Gaussian sampling, (e) Beta sampling and (f) Beta FLICM sampling methods.

For this subset, the unmixing results for Painted Metal Sheets and Self-blocking Blocks are pretty similar. The reason may be that the Painted Metal Sheets in this data set does not have a very high variance. For the self-blocking blocks, Beta-FLICM did a better job in that 1) the bottom left corner, the blocks are shown to be deep red and have more clear-cut boundaries exhibiting the true shape of the blocks; 2) the right gridded area, which should truly be Metal Sheets, are not shown to be blocks with Beta-FLICM while in the other methods they are shown to be red (blocks). As for Meadows endmember, which has a high variance, the proportion map result exhibits significant improvements in that 1) the meadow in the bottom left corner are much clearer in Beta-FLICM unmixing than others ; 2) the middle slanted long section, Beta-FLICM results are smoother and clearer.

Figure 10-16 shows the ground truth of the subset 3. This scene contains four endmembers: shadows, meadows, painted metal sheets and self-blocking blocks (including background when unmixing). Figure 10-17 shows the proportion maps of all methods for each endmembers. The sampling methods run up to 20000 iterations and the image is pre-segmented into 100 clusters by Beta FLICM methods.

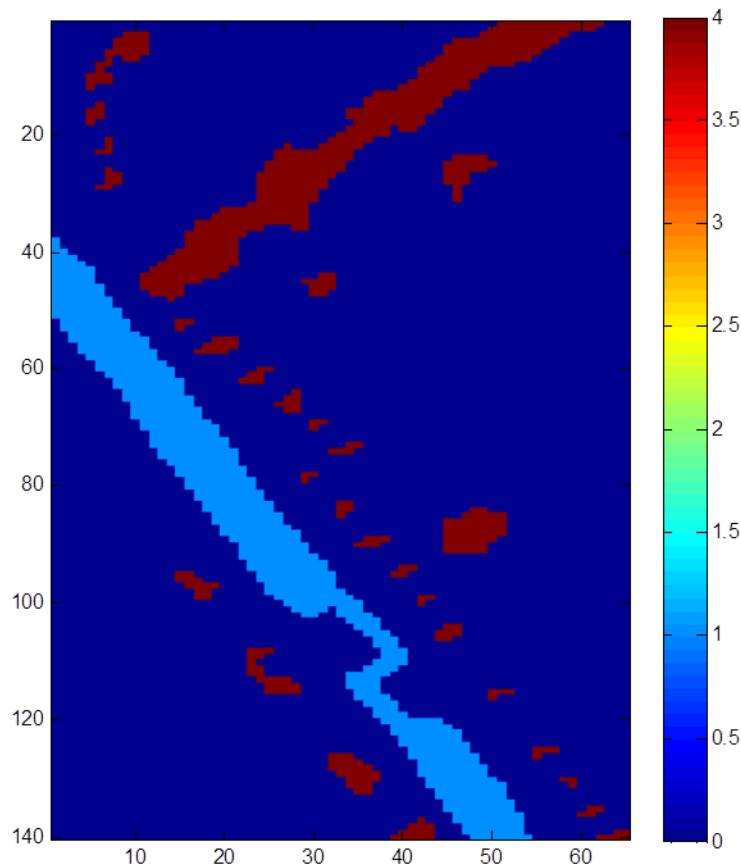
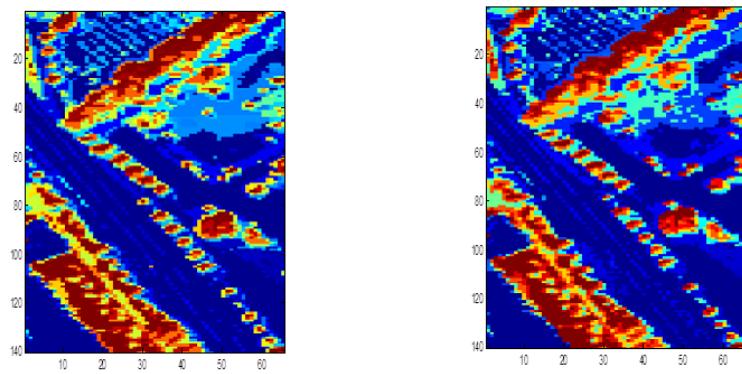
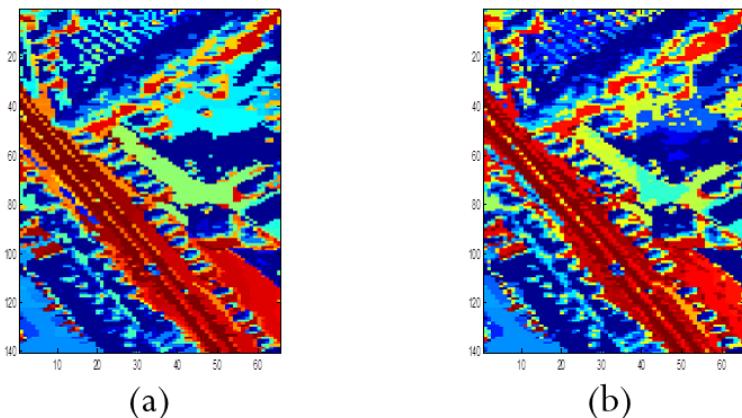


Figure 10-18 Ground Truth of PaviaU data subset 3. Endmember number=3.



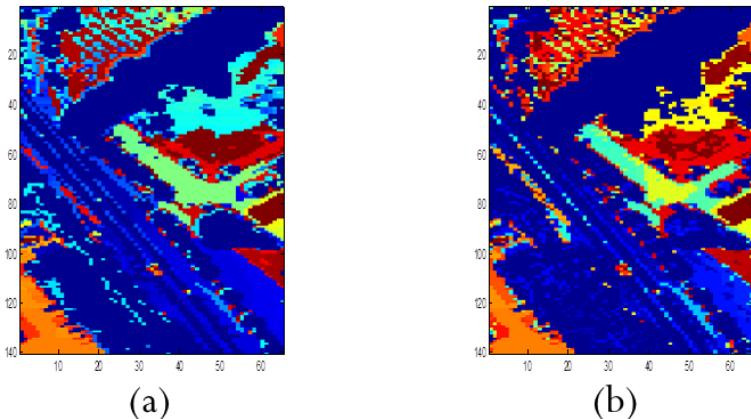
(a)

(b)



(a)

(b)



(a)

(b)

Figure 10-19 Proportion map of trees, asphalt and background unmixing method on PaviaU data subset, from top to bottom. Each proportion map has images showing the unmixing result with data pre-clustered by (a)  $K$ -means clustering and (b) FLICM clustering methods.

The purpose of this subset is to demonstrate that FLICM is a useful, appropriate and effective way to pre-cluster the image data as compared to traditional K-means. Beta FLICM, which uses local information—neighboring pixels, is better than K means, which uses only mean of points/pixels.

For the Tree endmember, there is no blurry area in proportion maps generated by Beta FLICM method, only clear-cut unmixing area of trees. In the upper part of the proportion map, the ground is not emphasized as trees, meaning less noise.

For the Asphalt, there are very significant improvements: the deep red part (appears to be a slanted main road) is much more clear and shown to be “deep red” in the FLICM results. The middle upside-down-T-shaped area (possibly a building) is not misclassified as asphalt. The open ground area in the upper right side of the map is not misclassified as asphalt and much clearer (deep blue). The same applies for the top open ground area.

#### 10.2.5. Gulfport Hyperspectral Image Data

##### 10.2.5.1. Gulfport Data Endmember Distribution

Gulfport data is investigated for the comparison between hyperspectral data and their representation using the Beta distributions and Gaussian distributions.

Quantile-Quantile (Q-Q) plots (Wilk, et al., 1968) are used to visualize the difference between the Beta distribution representation, the Gaussian distribution representation, and the original data (Figure 10-21). A Q-Q plot compares the similarity of two distributions by plotting their inverse cumulative distribution values at regular intervals versus one another. It was found that, for some materials, the Beta distribution provided a better fit to measured hyperspectral data. However, for other materials, it was

found that the Gaussian distribution provided a better fit. For example, 300 pixels of pure grass were extracted from Gulfport hyperspectral image. For each band of this extracted data, the parameters for a Gaussian distribution and a Beta distribution are estimated using a Maximum-Likelihood approach. In, the Q-Q plots comparing the Beta distribution to the histogram of the measured data and comparing the Gaussian distribution to the histogram of the measured data at the 368 nm wavelength is shown. In these plots, the x-value of a point on Q-Q plot corresponds to the quantile of the fitted Beta or Gaussian distribution, and the y-value is the quantile of the sample data. Thus, if the fitted distribution and the sample quantiles are equal, the Q-Q plot will be the 45-degree line  $y = x$ . Furthermore, Figure 10-20 shows the fitted Beta and Gaussian distributions plotted over the histogram of the grass spectral values at wavelength 368nm. Both of these figures demonstrate that, for some materials, the Beta distribution may provide a better fit over the Gaussian distribution. Future work will include determining, for each material, what distribution provides the best representation.

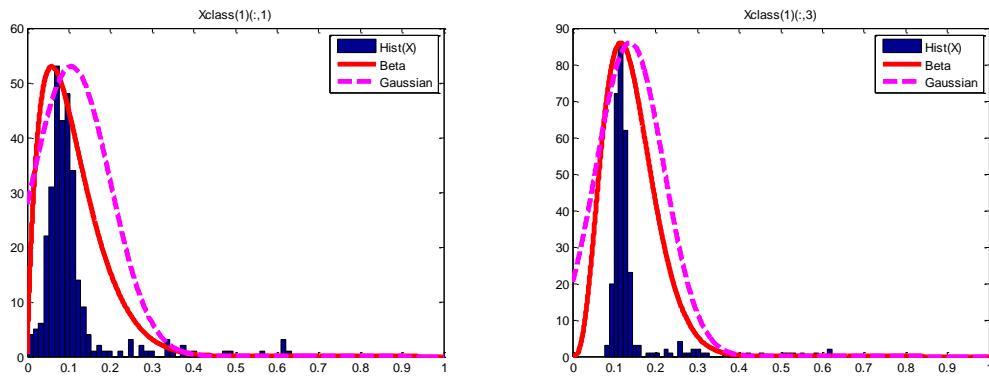


Figure 10-20 Gulfport hyperspectral image data endmember demonstration. Left: Grass at wavelength 368nm. Right: 387nm.

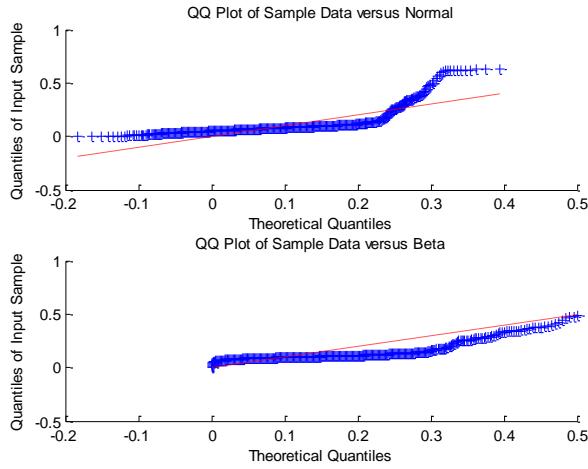


Figure 10-21 Gulfport hyperspectral image data endmember q-q plot with endmember Grass at 368nm.

#### 10.2.5.2. Gulfport Data Unmixing Results

Figure 10-8 shows the RGB image of the Gulfport data used in unmixing experiments. All of the above unmixing methods, including FCLS, Gaussian quadratic programming, Gaussian sampling, Beta quadratic programming, Beta sampling, Beta-FLICM, Beta-Superpixel and Beta-spatial-K-means methods, are tested.

Figure 10-22 shows the cluster result of spatial k-means of Gulfport data when cluster number varies from 15 to 40. In order to find one cluster result that is neither too coarse nor too “fine”, cluster results when cluster number equals 15 is taken into the unmixing process.

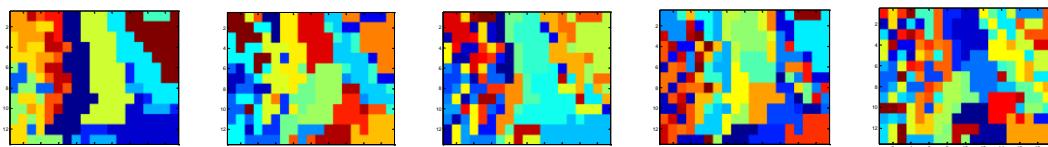


Figure 10-22 Gulfport hyperspectral image data cluster result of spatial K-means when cluster number = 15, 20, 30, 35, 40.

Figure 10-23 shows the segmentation result of entropy rate superpixel segmentation (superpixel 2) of Gulfport data when cluster number varies from 15 to 40.

In order to find one cluster result that is neither too coarse nor too “fine”, segmentation results when cluster number equals 30 is taken into the unmixing process. In this experiment, Superpixel 2 unmixing algorithm takes the RGB image of the Gulfport data, where R, G and B bands correspond to bands 26, 14 and 8.

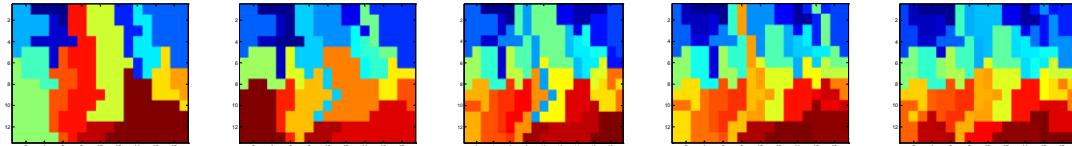
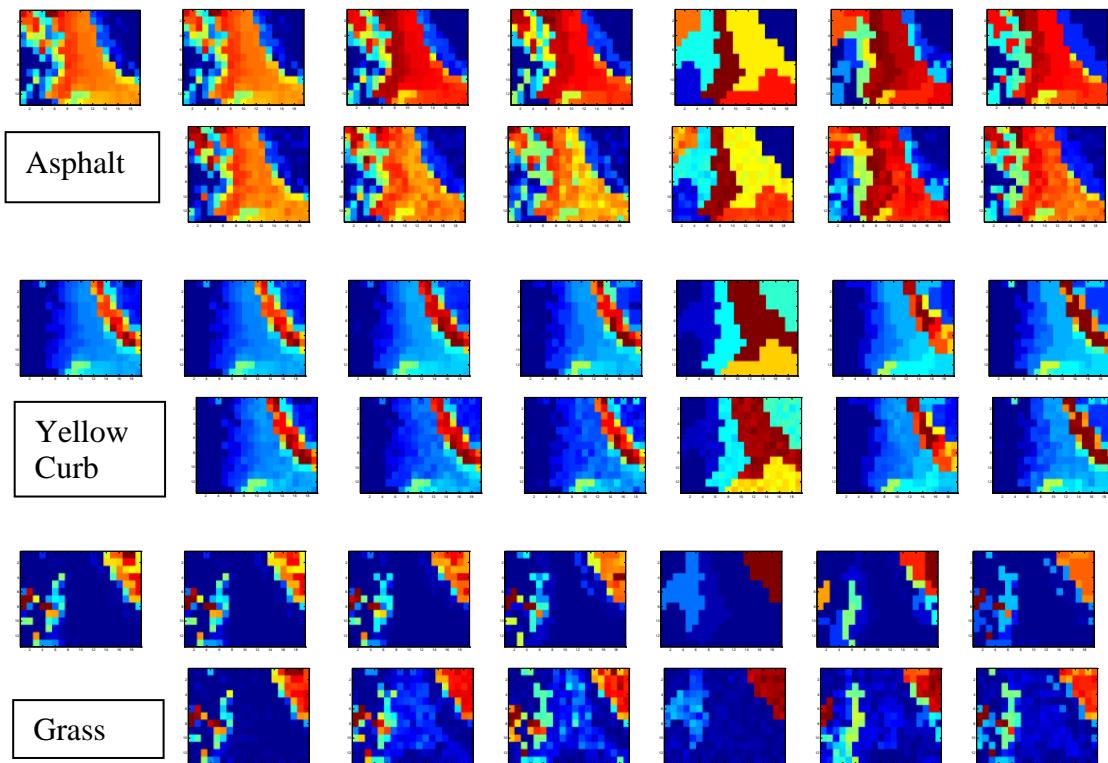


Figure 10-23 Gulfport hyperspectral image data cluster result of superpixel 2 when cluster number = 15, 20, 30, 35, 40.

Figure 10-24 shows the proportion maps of all methods for each endmembers. The sampling methods run up to 20000 iterations. In sampling methods,  $\sigma_{mean} = 0.001$  and  $\sigma_{var} = 100$ .



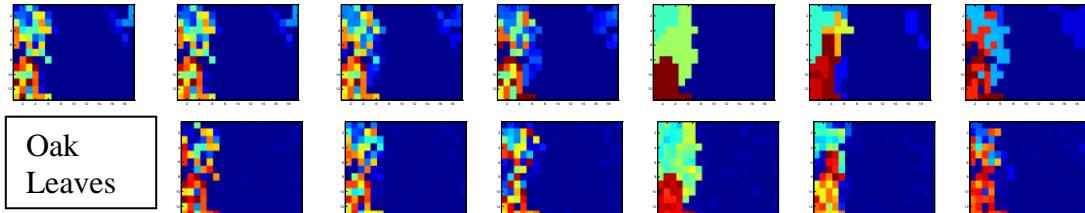


Figure 10-24 Unmixing results of Gulfport data. Top to Bottom: Asphalt, Yellow Curb, Grass and Oak Leaves. Unmixing methods from left to right (The same method are in the same column. For example Gaussian QP is second column first line and Gaussian MH method is second column second line): FCLS, Gaussian quadratic programming, Gaussian sampling, Beta quadratic programming, Beta sampling, Beta-FLICM quadratic programming, Beta-FLICM sampling, Beta-Superpixel 1 quadratic programming, Beta-Superpixel 1 sampling, Beta-Superpixel 2 quadratic programming, Beta- Superpixel 2 sampling, , Beta-Spatial-K-means quadratic programming and Beta-Spatial-K-means sampling.

As shown in Figure 10-24, the pixels corresponding to asphalt presents a deeper color on proportion maps when compared with FCLS method and NCM methods. This indicates that the proportion maps from BCM methods for Asphalt endmember yield larger proportion values for the endmembers, meaning better unmixing results. The proportion maps from Beta-Superpixel 1 shapes similar to superpixel segmentation labels, which is hard to observe and correctly identify endmember regions. On the other hand, Beta-Superpixel 2 and Beta-Spatial-K-means unmixing yield satisfying proportion maps with deep red concentrated on asphalt area.

All methods are able to separate the Yellow Curb endmember from other endmembers but proportion maps from BCM-Spatial methods show a darker and more concentrated color in yellow curb pixels and lighter, smoother color in other pixels. Beta-Superpixel 1 results are again not satisfying as it includes other regions besides the yellow curb region in the proportion maps. Beta-Superpixel 2 is able to unmix the Yellow Curb endmember, but with slight inaccuracies around the borders as shown in the orange-and-yellow-colored pixels around the true yellow curb pixels.

The unmixing results of Grass endmember is not entirely satisfying for FCLS and NCM methods as the proportion maps show deep color on both the grass region and the oak leaves region. As mentioned before, Grass endmember and Oak Leaves endmember

have similar spectral signature shape but with different variance across the samples. Therefore, for unmixing methods that does not consider endmember variability such as FCLS and NCM quadratic programming methods, it is natural to assume (and to prove, as can be seen in proportion maps that grass and oak leaves will not be unmixed correctly and clearly. BCM methods did a better job in unmixing grass endmember and produce proportion maps such that the grass region is highlighted and the background pixels, including the oak leave regions, are in light blue colors. The proposed BCM approaches, on the other hand, produce proportion maps with lighter color in other pixels than the grass pixels, especially the oak leaves pixels. For Oak Leaves endmember, BCM approaches produce deeper color on the oak leaves region in proportion maps, especially BCM-Spatial unmixing algorithms (low noise in the background and deep red color on oak leaves region).

It can also be observed that, between BCM-Spatial quadratic programming (QP) method and BCM-Spatial sampling method, the sampling method produces almost no noise in the background and very concentrated color in oak leaves region, while the QP method still has certain noise in the background (light blue right beside the oak leaves and blue on the top right corner, which is supposed to be grass region). This is probably due to the fact that BCM-Spatial algorithms take into account local spatial information of the image, and spatial k-means provides such information by pre-clustering the image. In addition, BCM-Spatial sampling method considers variance as well. In this case, during unmixing, BCM-Spatial algorithms are able to separate oak leaves endmember not only based on spectral information, but also spectral variance as well as spatial information, thus producing better unmixing results shown in proportion maps.

It can be concluded from the proportion maps that BCM-Spectral and BCM-Spatial unmixing algorithms are able to successfully perform unmixing. BCM methods

performs better than FCLS and NCM methods for Gulfport data set, and BCM-Spatial unmixing methods produce more concentrated endmember and lighter, smoother background in proportion maps. BCM-Spatial- $K$ -means algorithms yield the most accurate and truthful proportion maps among all the unmixing algorithms and approaches.

### 10.3. Varying Parameters in Unmixing Methods

#### 10.3.1. Relationship between $N$ and Proportion Errors

Four unmixing methods illustrated in chapters 3-6 are conducted on the Gaussian and Beta toydata generated in section 10.1.1. The number of endmembers was set to be three (known) for the algorithm. Considering the influence of the number of data points inside the simplex, we ran the algorithm for  $N$  from 13 to 500 with an increment of 20. For each  $N$ , the algorithm was run continuously for 20,000 iterations, and the whole unmixing process is repeatedly run for 10 iterations. The mean value for the error (Euclidean distance) between unmixing results  $P_{\text{estimate}}$  and true proportion values  $P_{\text{true}}$  over these 10 iterations are recorded as the final  $P_{\text{Error}}$  output. The variance over 10 error values are recorded and calculated. Figure 10-25 shows the plots of proportion error versus number of data points  $N$  of Gaussian toydata example using Gaussian quadratic programming, Gaussian sampling, Beta quadratic programming, and Beta Sampling unmixing methods. The variances over 10 iterations are reflected by plotting the error bar for each  $N$ , centered on mean  $P_{\text{Error}}$  values.

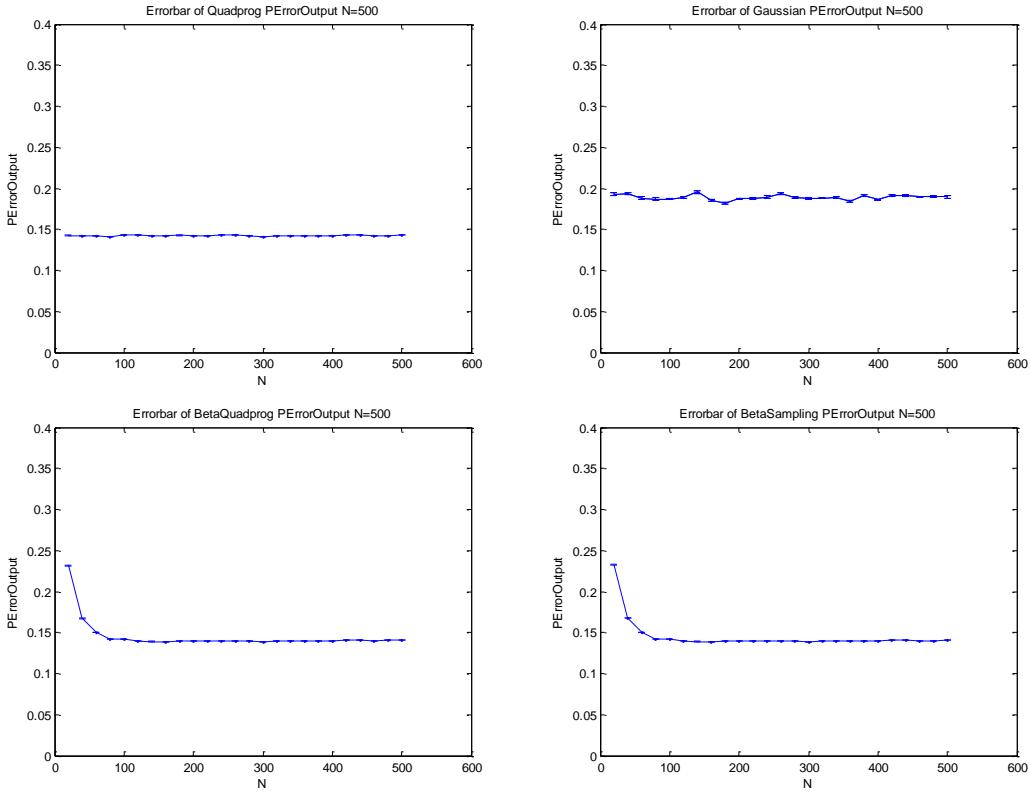


Figure 10-25 Proportion error versus number of data points  $N$  of Gaussian toy data example using Gaussian quadratic programming, Gaussian sampling, Beta quadratic programming, and Beta sampling unmixing methods. Unmixing was run on each data point for 100 iterations and error bar was drawn centered on the mean value and based on the variance of proportion error over 100 iterations. Top left: Gaussian toy data Beta quadratic programming unmixing. Top right: Gaussian toy data Gaussian Sampling unmixing. Bottom left: Gaussian toydata Beta Quadratic Programming unmixing. Bottom right: Gaussian toy data Beta sampling unmixing.

For Beta toy data, every step is the same as above except that the unmixing algorithms are ran on Beta toy data generated as described in section 10.1.1. Figure 10-26 shows the plots of proportion error versus number of data points  $N$  of Gaussian toydata example using Gaussian quadratic programming, Gaussian sampling, Beta quadratic programming, and Beta Sampling unmixing methods. The variances over 10 iterations are reflected by error bar for each  $N$ , centered on mean PError values.

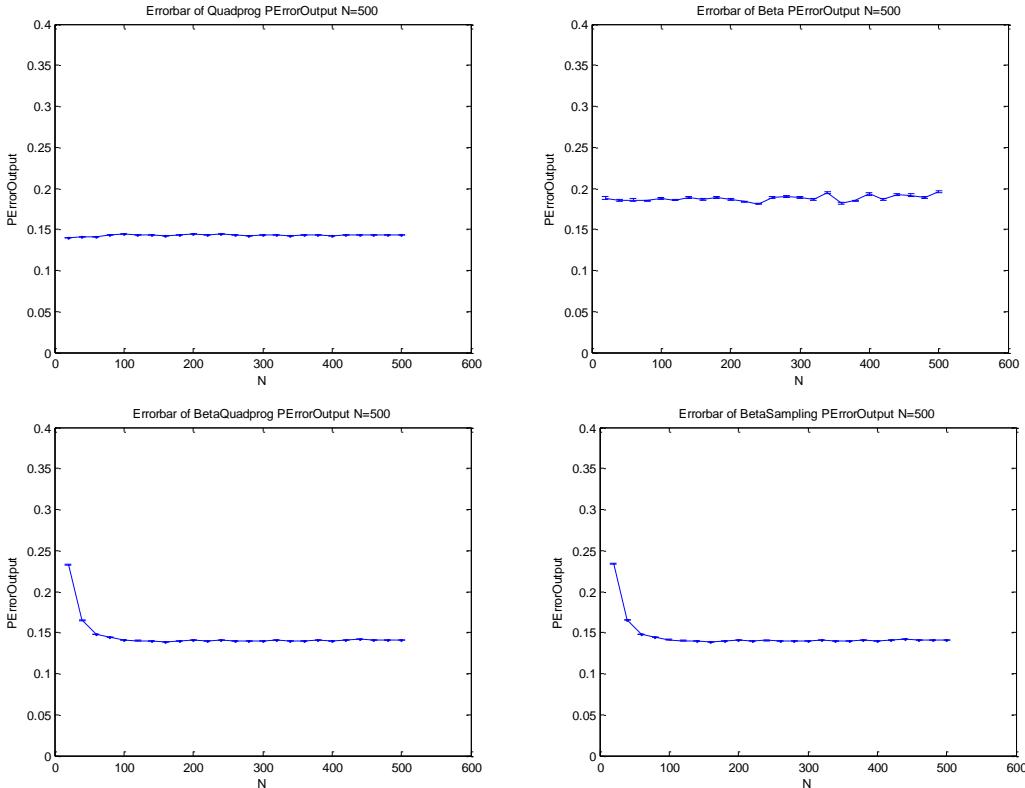


Figure 10-26 Proportion error versus number of data points  $N$  of Beta toy data example using Gaussian quadratic programming, Gaussian sampling, Beta quadratic programming, and Beta sampling unmixing methods. Unmixing was run on each data point for 100 iterations and error bar was drawn centered on the mean value and based on the variance of proportion error over 100 iterations. Top left: Beta toy data Beta quadratic programming unmixing. Top right: Beta toy data Gaussian Sampling unmixing. Bottom left: Beta toydata Beta Quadratic Programming unmixing. Bottom right: Beta toy data Beta sampling unmixing.

In each iteration, endmembers are updated only when the new iteration produces better log likelihood than the last iteration. That is to say, only the “best” proportion sample values are preserved. From the above figures we see there are no significant variation of *PError* Output against  $N$ , which makes sense, as the toy data is generated to be uniformly distributed in the simplex and endmember samples are drawn randomly with Gaussian and Beta distribution so that  $N$  would not have an influence on the proportion error output.

All unmixing methods yield a relatively low proportion error. Error is high for Beta unmixing methods when  $N$  is very small, possibly due to the fact that Beta unmixing methods use  $K$  nearest neighbors to estimate the center pixel ( $K = 12$  in this case). When  $N$  is small ( $N < 100$ ), there is no sufficient information of neighbors to help determine the composition of the center pixel, thus causing high proportion errors.

#### 10.3.2. Relationship between Endmember Covariance and Proportion Errors

Figure 10-27 and Figure 10-28 illustrate the relationship between endmember covariance and proportion errors on Gaussian and Beta toy data. It can be seen that the proportion error stays stable as noise covariance changes for Beta toy data, which shows the robustness of the unmixing algorithms. However, the proportion errors increase when the endmember covariance increase in Gaussian toy data, meaning the Gaussian toy data is more affected by the endmember covariance. Gaussian unmixing method yields a lower proportion error than Beta unmixing methods on Gaussian toy data overall.

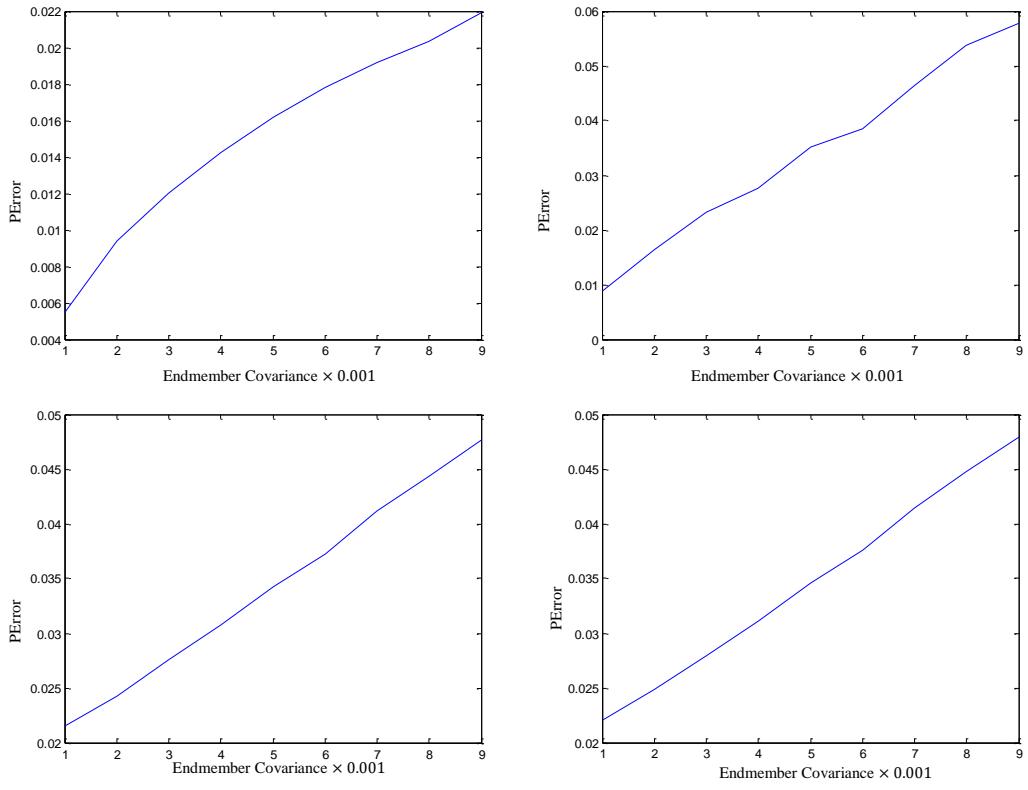


Figure 10-27 Proportion error versus Endmember covariance using Gaussian quadratic programming, Gaussian sampling, Beta quadratic programming, and Beta sampling unmixing methods on Gaussian toy data. Top left: Gaussian toy data Beta quadratic programming unmixing. Top right: Gaussian toy data Gaussian Sampling unmixing. Bottom left: Gaussian toy data Beta Quadratic Programming unmixing. Bottom right: Gaussian toy data Beta sampling unmixing. The  $x$  axis (Endmember covariance) is on  $\times 0.001$  scale.

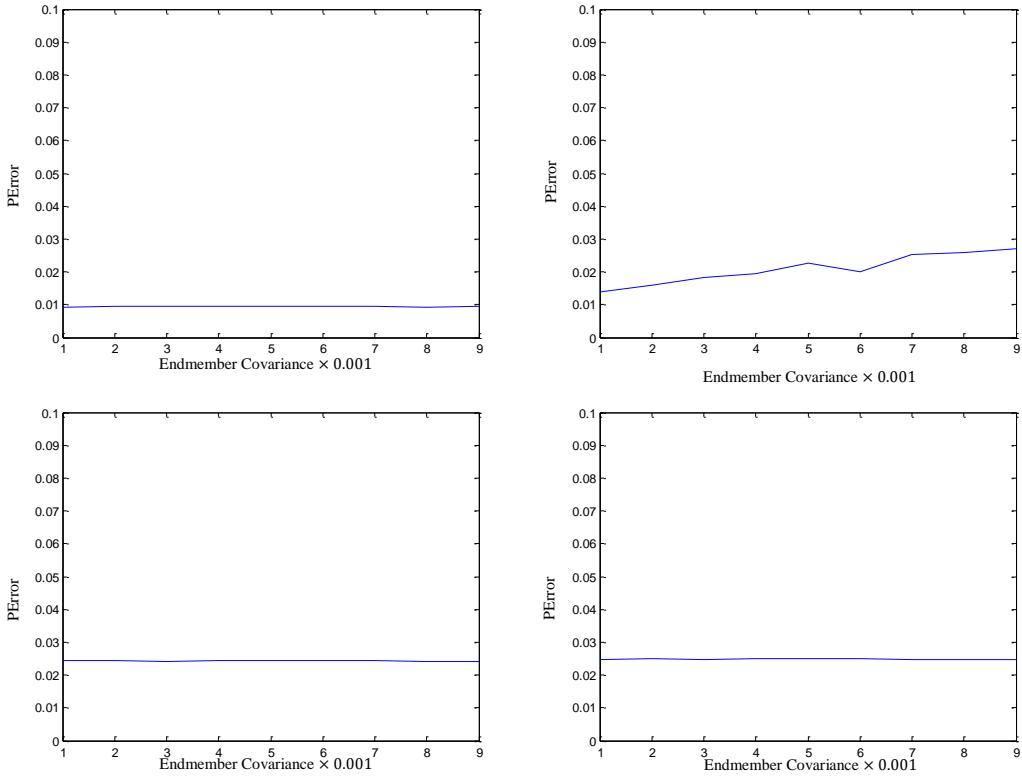


Figure 10-28 Proportion error versus Endmember covariance using Gaussian quadratic programming, Gaussian sampling, Beta quadratic programming, and Beta sampling unmixing methods on Beta toy data. Top left: Beta toy data Beta quadratic programming unmixing. Top right: Beta toy data Gaussian Sampling unmixing. Bottom left: Beta toydata Beta Quadratic Programming unmixing. Bottom right: Beta toy data Beta sampling unmixing. The  $x$  axis (Endmember covariance) is on  $\times 0.001$  scale.

### 10.3.3. Relationship between Noise Covariance and Proportion Errors

Figure 10-29 and Figure 10-30 illustrate the relationship between noise covariance and proportion errors on Gaussian and Beta toy data. It can be seen that the proportion error stays stable as noise covariance changes, which shows the robustness of the unmixing algorithms. Beta sampling method yields a low proportion error overall.

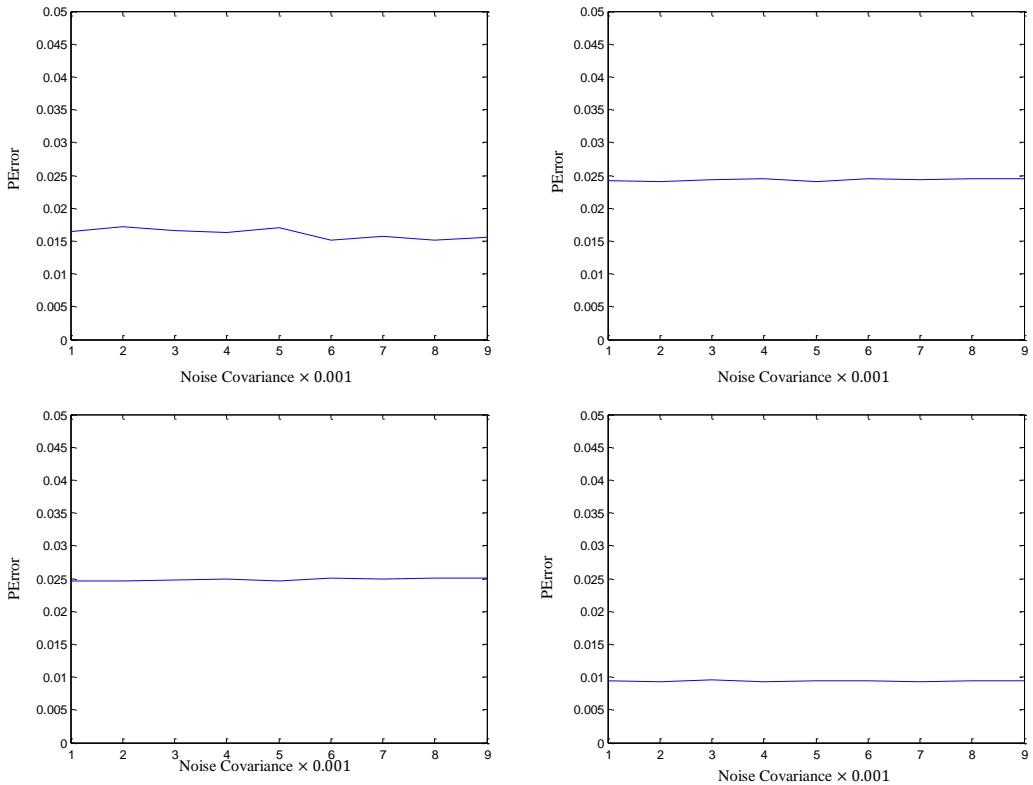


Figure 10-29 Proportion error versus noise covariance using Gaussian Sampling, Beta Quadratic Programming, Beta Sampling and Quadratic Programming unmixing methods on Gaussian toy data. Top left: Gaussian toydata Gaussian Sampling unmixing. Top right: Gaussian toydata Beta Quadratic programming unmixing. Bottom left: Gaussian toydata Beta Quadratic Programming unmixing. Bottom right: Gaussian toydata Quadratic Programming unmixing. The X axis (noise covariance) is \*0.001 value.

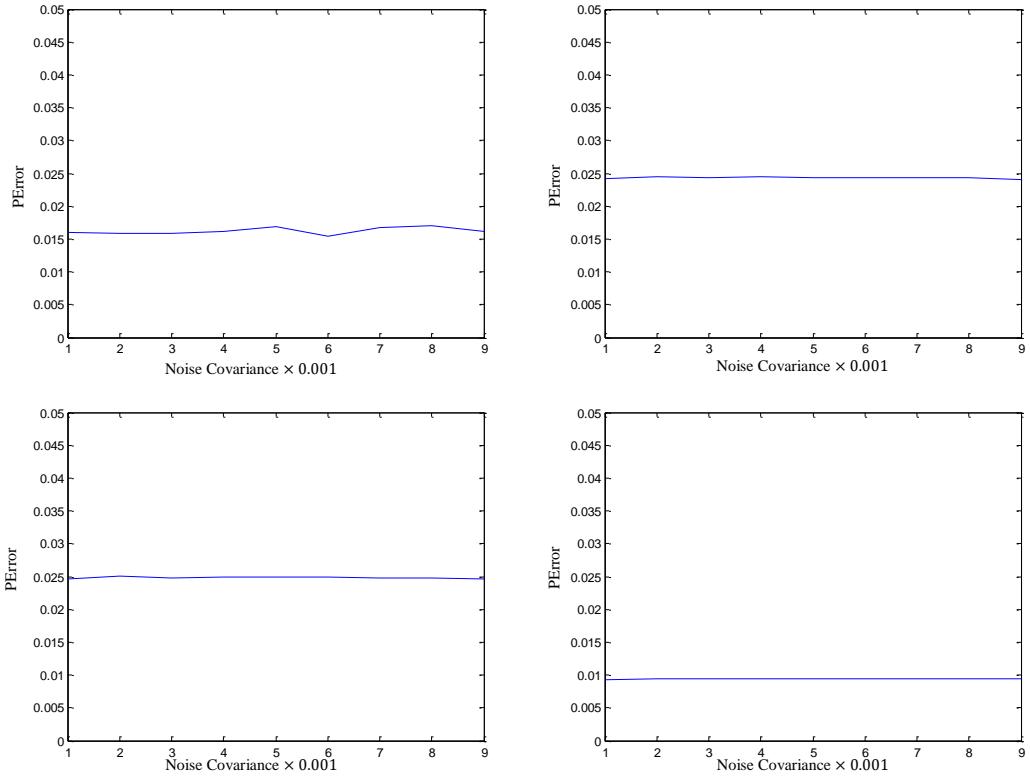


Figure 10-30 Proportion error versus noise covariance using Gaussian Sampling, Beta Quadratic Programming, Beta Sampling and Quadratic Programming unmixing methods on Beta toy data. Top left: Beta toydata Gaussian Sampling unmixing. Top right: Beta toydata Beta Quadratic programming unmixing. Bottom left: Beta toydata Beta Quadratic Programming unmixing. Bottom right: Beta toydata Quadratic Programming unmixing. The X axis (noise covariance) is \*0.001 value.

#### 10.3.4. Relationship between Cluster Number $K$ and Proportion Errors

Figure 10-31 illustrate the relationship between cluster number  $K$  and proportion errors,  $PError$ , on simulated data set 2 using BCM-Spatial- $K$ -means unmixing algorithm. In the experiment,  $K = 5:195$  with an increment of 5. It can be seen that the proportion error is high when  $K$  is small (when  $K = 5$ , for example), and decreases as  $K$  gets larger. When  $K$  is larger than 80 (80 clusters in the image),  $PError$  stays stable. This result is as expected as when cluster number is small, each component after segmentation is large and the “spatial neighbors” of a pixel in each cluster may include

other materials or have different composition. On the other hand, if  $K$  is large, each cluster is small enough to include only the spatial neighbors of a pixel that have similar composition. Large  $K$  leads to a small proportion error as BCM-Spatial- $K$ -means unmixing is based on the assumption that the spatial neighbors of a certain pixel have similar composition and the algorithm uses spatial neighbors to unmix each pixel.

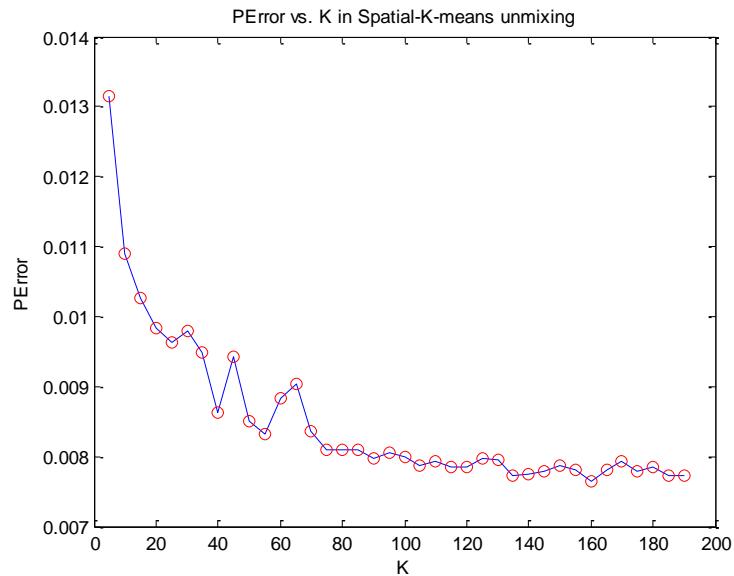


Figure 10-31 Proportion error versus cluster number  $K$  using BCM-Spatial- $K$ -means unmixing on Simulated data set 2.  $K = 5:195$  with an increment of 5.

### 10.3.5. Exploration of band selection in Superpixel 2 method

As mentioned in Section 10.2.5, current Superpixel 2 unmixing algorithm takes the RGB image of the Gulfport data, where R, G and B bands are three bands selected from the fifty-three bands of Gulfport data. This section will explore the relationship between proportion errors and the selection of bands as the input data of Superpixel 2 unmixing algorithm.

A random function is written to generate three bands in random order from all spectral bands in the input image. Then, for each band combination, BCM-Superpixel 2 unmixing algorithm is ran multiple times (in this experiment, three runs), and proportion errors are calculated against the known true proportion values.

Table 7 Proportion Error and Band Selection

Bands			Proportion Errors
36	1	30	0.06±2.81e-09
50	15	44	0.06±5.00e-09
7	40	29	0.05±7.81e-09
1	2	17	0.06±2.81e-09
32	40	22	0.06±5.00e-09
26	14	8	0.06±2.31e-06
<b>Summary</b>			<b>0.06±6.63e-06</b>

The proportion error “summary” term in Table 7 is calculated based on the proportion values over all the above band selections. The mean value is the mean of all proportion error mean values, and the variance is the variance of all proportion error mean values. It can be seen that the variance, 6.63e-06, is very small, meaning the random selection of bands does not have a heavy impact on the proportion error output. Future work may include the extension of BCM-Superpixel 2 algorithm to more than three bands.

#### 10.4. Computational Complexity

This section discusses and compares the computational complexity between QP (quadratic programming) and MH (Metropolis-Hastings sampling) approaches.

In QP approach, a variety of solving methods can be used to solve the quadratic programming problem. For positive definite  $Q$ , the ellipsoid method for solving QP problems can solve in polynomial time.(Vavasis, 2009)

In MH approach, the most computational consuming step is where the loglikelihood of data is calculated. Recall the likelihood can be calculated using (68).

Taking the log of the likelihood, the loglikelihood term is

$$\text{loglikelihood} = -\frac{\left(\frac{\alpha_{jd}}{E_{id} - \sum_{j=1}^M p_{ij} \alpha_{jd} + \beta_{jd}}\right)^2}{2\sigma_{mean}^2} - \frac{\left(\frac{\alpha_{jd}\beta_{jd}}{S_{id} - \sum_{j=1}^M p_{ij}^2 \left(\frac{\alpha_{jd}\beta_{jd}}{\left(\alpha_{jd} + \beta_{jd}\right)^2 (\alpha_{jd} + \beta_{jd} + 1)}\right)}\right)^2}{2\sigma_{var}^2} \quad 102$$

The computational complexity of a single iteration for each pixel is  $O(DM)$ ,  $M$  is the number of endmembers,  $D$  is the number of bands. The computational complexity of  $I$  iterations across  $N$  data points is  $O(INDM)$ , where  $N$  is the number of pixels and  $I$  is the number of iterations. The computational complexity discussed here includes the computational complexity for computing the loglikelihood as this is the most significant term in the computation process. There are other steps that are being excluded.

## 11. CONCLUSIONS AND FUTURE WORK

### 11.1. Conclusions

This thesis proposes hyperspectral unmixing methods incorporating spectral variability and spatial information based on the Beta Compositional model. Quadratic programming and Metropolis-Hastings sampling approaches are discussed for each hyperspectral unmixing methods. Experimental results are presented on toydata, simulated hyperspectral data and real hyperspectral image data, with comparison to hyperspectral unmixing methods in the literature (such as unmixing methods based on the Normal Composition Model). The relationship of parameters, such as the number of data points, variance of endmembers and noise versus the error in proportions are explored.

As an improvement to existing BCM unmixing methods, spatial information is incorporated to account for variance and help determine the proportion values for each pixel in the scene. It is assumed that the nearest spatial neighbor of a pixel can best represent the composition of the pixel, which makes sense in real hyperspectral images than using spectral neighbors as in existing BCM unmixing method. FLICM, superpixel and Spatial-K-Means algorithms are used to obtain spatial information. Results show that both BCM-Spectral and BCM-Spatial approaches are able to successfully perform unmixing on both simulated data and real hyperspectral imagery. BCM-spatial algorithms show good performance even when the image has high noise.

Among BCM-Spatial unmixing methods, BCM-Spatial-*K*-means algorithms show a better result in unmixing. Spatial-*K*-means algorithm can easily cluster the input data set incorporating spatial information, and it does not need to specify a neighboring “window size” as required in FLICM algorithms. BCM-Superpixel methods tend to produce “segmented” proportion maps with crossovers between materials, which is

undesirable. BCM-Superpixel 2 performs slightly better than BCM-Superpixel 1, and it is possibly due to the balance term in the entropy rate-based superpixel segmentation method, which balances the size and number of clusters, making sure that the clusters are smooth, compact and homogeneous. BCM-FLICM algorithm has an improved result when applying to data that has a “block” of one-endmember regions. The reason for the improvement may be that FLICM can label and define the “block” as one cluster and the clustering result will benefit the unmixing results. BCM-Spectral algorithms may find points with similar spectral values but not necessarily of the same composition, but BCM-Spatial algorithms can make use of the spatial neighbors as well as the spectral information of the data.

When compared with methods that does not consider spectral variability, such as FCLS, it is clear that both BCM-Spectral unmixing and BCM-Spatial have a better performance, especially on data that consists endmembers that has similar spectral signature shape with different variance. When compared with unmixing algorithms under Normal Composition Model, the BCM unmixing algorithms yields lower error on data that consists of endmember to which Beta distribution is a better fit.

## 11.2. Future Work

This section discusses future work into hyperspectral unmixing. There is much room for developing new unmixing algorithms, such as considering cross-band variability and using a full covariance (as opposed to the current diagonal covariance). Computational efficiency can be considered as another factor to improve.

Currently, the prior knowledge of endmembers comes from the relatively pure pixels that are hand-picked from a hyperspectral image. Further investigation may be

devoted to the selection and the acquisition of endmember information, such as the number of endmember distributions and the covariance parameters for each endmember distribution.

Moreover, since the spatial neighbors are deemed helpful to hyperspectral unmixing through this research, methods of incorporating spatial information other than using index values can be studied, such as the elevation information in LiDAR systems.

As discussed in Section 10.3.5, current BCM-Superpixel 2 algorithms are applied to three bands from the real hyperspectral image data. Future work may include the extension of BCM-Superpixel 2 unmixing algorithms to multiple and full hyperspectral bands.

Other future work may include the investigation into developing BCM-based non-linear mixing models. Scattering between endmembers and intimate mixtures in the scene can lead to nonlinear mixing model representation and may benefit this study on hyperspectral unmixing algorithms as well.

## BIBLIOGRAPHY

**Adams J. B., Smith M. O. and Johnson P. E.** Spectral mixture modeling: A new analysis of rock and soil types at the Viking Lander 1 site. [Journal] // Journal of Geophysical. - 1986. - Vol. 91. - p. 8098–8112.

**Akbari H. [et al.]** Hyperspectral imaging and quantitative analysis for prostate cancer detection [Journal]. - [s.l.] : Journal of Biomedical Optics, 2012. - 7 : Vol. 17.

**Asner G. P., Bustamante M. M. C. and Townsend A. R.** Scale dependence of biophysical structure in deforested areas bordering the Tapajo's National Forest, Central Amazon [Journal] // Remote Sensing of Environment. - [s.l.] : Elsevier, 2003. - 4 : Vol. 87. - pp. 507-520.

**Bajjouk T., Populus J. and Guillaumont B.** Quantification of Subpixel Cover Fractions Using Principal Component Analysis and a Linear Programming Method: Application to the Coastal Zone of Roscoff (France) [Journal] // Remote Sensing of Environment. - 1998. - Vol. 64. - pp. 153-165.

**Bateson A. and Curtiss B.** A method for manual endmember selection and spectral unmixing [Journal] // Remote Sensing of Environment. - 1996. - 3 : Vol. 55. - pp. 229-243.

**Bateson C. A., Asner G. P. and Wessman C. A** Endmember bundles: a new approach to incorporating endmember variability into spectral mixture analysis [Journal] // Geoscience and Remote Sensing, IEEE Transactions on . - 2000. - 2 : Vol. 38. - pp. 1083-1094.

**Berman M. [et al.]** ICE: a statistical approach to identifying endmembers in hyperspectral images [Journal] // IEEE Trans. Geosci. and Remote Sens.. - 2004. - 10 : Vol. 42. - pp. 2085–2095.

**Bezdek J. C.** Pattern Recognition with Fuzzy Objective Function Algorithms [Book]. - Norwell : Kluwer Academic Publishers, 1981.

**Bioucas-Dias J. M. and Figueiredo M. A. T.** Alternating Direction Algorithms for Constrained Sparse Regression: Application to Hyperspectral Unmixing [Journal] // Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop. - 2010. - Vol. 1. - pp. 1-4.

**Bioucas-Dias J. M. [et al.]** Hyperspectral Unmixing Overview: Geometrical, Statistical, and Sparse Regression-Based Approaches [Journal] // Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of. - Apr. 2012. - 2 : Vol. 5. - pp. 354-379.

**Bishop C. M.** Pattern Recognition and Machine Learning (Information Science and Statistics [Book]. - [s.l.] : Springer-Verlag New York, Inc., 2006.

**Boardman J. W., Kruse F. A. and Green R. O.** Mapping Target Signatures via Partial Unmixing of AVIRIS Data [Journal] // in Summaries of the Fifth Annual JPL Airborne Earth Science Workshop. - Washington, D.C. : JPL Publication 95-1, 1995. - Vol. 1. - pp. 23-26.

**Boardman J. W.** Geometric Mixture Analysis of Imaging Spectrometry Data [Journal] // Geoscience and Remote Sensing Symposium, 1994. IGARSS '94. Surface and Atmospheric Remote Sensing: Technologies, Data Analysis and Interpretation., International . - Aug. 1994. - Vol. 4. - pp. 2369-2371.

**Bovolo F., Bruzzone L. and Carlin L.** A Novel Technique for Subpixel Image Classification Based on Support Vector Machine [Journal] // Image Processing, IEEE Transactions on . - 2010. - 11 : Vol. 19. - pp. 2983-2999.

**Brown M., Lewis H. G. and Gunn S. R.** Support vector machines for spectral unmixing [Journal] // Geoscience and Remote Sensing Symposium, 1999. IGARSS '99 Proceedings. IEEE 1999 International . - 1999. - Vol. 2. - pp. 1363-1365.

**Canham K. [et al.]** Spatially Adaptive Hyperspectral Unmixing [Journal] // Geoscience and Remote Sensing, IEEE Transactions on . - 2011. - 11 : Vol. 49. - pp. 4248-4262.

**Casella G. and George E. I.** Explaining the Gibbs Sampler [Journal] // The American Statistician. - 1992. - 3 : Vol. 46. - pp. 167-174.

**CGAL Project** The CGAL User and Reference Manual [Journal] / ed. CGAL Editorial Board 4.3 edition. - 2013.

**Chang C.-I.** An Information-Theoretic Approach to Spectral Variability, Similarity, and Discrimination for Hyperspectral Image Analysis [Journal] // IEEE TRANSACTIONS ON INFORMATION THEORY. - Aug 2000. - 5 : Vol. 46. - pp. 1927-1932.

**Chaudhry F. [et al.]** Pixel Purity Index-Based Algorithms for Endmember Extraction from Hypersepcetal Imagery [Book Section] // Recent Advances in Hyperspectral Signal and Image Processing / ed. Chang Chein-I. - Kerala : Transworld Research Network, 2006. - 81-7895-218-1.

**Chib S. and Greenberg E.** Understanding the Metropolis-Hastings algorithm [Journal] // The American Statistician. - 1995. - 4 : Vol. 49. - pp. 327–335.

**Cho M. A. [et al.]** Spectral variability within species and its effects on Savanna tree species discrimination [Journal] // Geoscience and Remote Sensing Symposium,2009 IEEE International,IGARSS 2009. - 2009. - Vol. 2. - pp. II-190,II-193.

**Combe J.-Ph. [et al.]** Analysis of OMEGA/Mars Express data hyperspectral data using a Multiple-Endmember Linear Spectral Unmixing Model (MELSUM): Methodology and first results [Journal] // Planetary and Space Science. - 2008. - 7 : Vol. 56. - pp. 951-975.

**Comon P.** Independent component analysis, A new concept? [Journal] // Signal Processing. - 1994. - 3 : Vol. 36. - pp. 287-314.

**Corp L. A. [et al.]** Forecasting corn yield with imaging spectroscopy [Journal] // Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International. - 2010. - pp. 1819-1822.

**Dennison P. E. and Roberts D. A.** Endmember selection for multiple endmember spectral mixture analysis using endmember average RMSE [Journal] // Remote Sensing of Environment. - 2003. - 2-3 : Vol. 87. - pp. 123-135.

**Dennison P. E., Halligan K. Q. and Roberts D. A.** A comparison of error metrics and constraints for multiple endmember spectral mixture analysis and spectral angle mapper [Journal] // Remote Sensing of Environment. - 2004. - 3 : Vol. 93. - pp. 359-367.

**Dikin I. I.** Iterative solution of problems of linear and quadratic programming [Journal] // Soviet Mathematics Doklady. - 1967. - Vol. 8. - pp. 674-675.

**Dobigeon N. [et al.]** Joint Bayesian endmember extraction and linear unmixing for hyperspectral imagery [Journal] // Signal Processing, IEEE Transactions on. - 2009. - 11 : Vol. 57. - pp. 4355-4368.

**Dobigeon N., Tourneret J.-Y. and Chang C.-I.** Semi-supervised linear spectral using a hierarchical Bayesian model for hyperspectral imagery [Journal] // Signal Processing, IEEE Transactions on. - 2008. - 7 : Vol. 56. - pp. 2684-2696.

**Donoho D. L. and Elad M.** Optimally Sparse Representation in General (non-Orthogonal) Dictionaries via  $\ell_1$  Minimization [Journal] // Proceedings of the National Academy of Sciences. - 2003. - 5 : Vol. 100. - p. 2197.

**Dunn J. C.** A fuzzy relative of the ISODATA process and its use in detecting compact Well-Separated clusters [Journal] // Journal of Cybernetics. - 1973. - 3 : Vol. 3. - pp. 32-57.

**Eches O. [et al.]** Bayesian Estimation of Linear Mixtures Using the Normal Compositional Model. Application to Hyperspectral Imagery [Journal] // Image Processing, IEEE Transactions on . - 2010. - 6 : Vol. 19. - pp. 1403-1413.

**Eches O., Dobigeon N. and Tourneret J.-Y.** Estimating the Number of Endmembers in Hyperspectral Images Using the Normal Compositional Model and a Hierarchical Bayesian Algorithm [Journal] // Selected Topics in Signal Processing, IEEE Journal of . - 2010. - 3 : Vol. 4. - pp. 582-591.

**Eismann M. T. and Stein D.** Stochastic mixture modeling [Book Section] // Hyperspectral Data Exploitation: Theory and Applications / ed. C.Chang. - New York : Wiley, 2007.

**Felzenszwalb P. F. and Huttenlocher D. P.** Efficient Graph-Based Image Segmentation [Journal] // International Journal of Computer Vision. - 2004. - Vol. 59. - 2.

**Franke J. [et al.]** Hierarchical Multiple Endmember Spectral Mixture Analysis (MESMA) of hyperspectral imagery for urban environments [Journal] // Remote Sensing of Environment. - 2009. - 8 : Vol. 113. - pp. 1712-1723.

**Frigui H. and Krishnapuram R.** Clustering by competitive agglomeration [Journal] // Pattern Recognition. - 1997. - 7 : Vol. 30. - pp. 1109-1119.

**Gamba P.** Human Settlements: A Global Challenge for EO Data Processing and Interpretation [Journal] // Proceedings of the IEEE . - 2013. - 3 : Vol. 101. - pp. 570-581.

**Gillis D. [et al.]** A generalized linear mixing model for hyperspectral imagery [Journal] // Proc. SPIE—Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XIV. - 2008. - 1 : Vol. 6966. - p. 69661B.

**Goenaga M. A [et al.]** Unmixing Analysis of a Time Series of Hyperion Images Over the Guánica Dry Forest in Puerto Rico [Journal] // Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of. - [s.l.] : IEEE, 2012. - Vol. 99. - pp. 1-10.

**Goetz A. F. H. [et al.]** Imaging Spectrometry for Earth Remote Sensing [Journal] // Science, New Series. - 1985. - 4704 : Vol. 228. - pp. 1147-1153.

**Green A. [et al.]** A Transformation for Ordering Multispectral Data in Terms of Image Quality with Implications for Noise Removal [Journal] // Geoscience and Remote Sensing, IEEE Transactions on. - Jan. 1988. - 1 : Vol. 26. - pp. 65-74.

**Green R. O. [et al.]** Imaging Spectroscopy and the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) [Journal] // Remote Sensing of Environment. - 1998. - 3 : Vol. 65. - pp. 227-248.

**Guglietta D. [et al.]** Hyperspectral Airborne Remote Sensing for Multi-Temporal Assessment about Urban Changes in Naples Area [Journal] // Analysis of Multi-temporal Remote Sensing Images, 2007. MultiTemp 2007. International Workshop on the . - 2007. - pp. 1-5.

**Heiden U. [et al.]** Ecological evaluation of urban biotope types using airborne hyperspectral HyMap data [Journal] // Remote Sensing and Data Fusion over Urban Areas, 2003. 2nd GRSS/ISPRS Joint Workshop on. - 2003. - pp. 18-22.

**Heinz D. C. and Chang C.-I.** Fully constrained least squares linear spectral mixture analysis method for material quantification in hyperspectral imagery [Journal] // Geoscience and Remote Sensing, IEEE Transactions on. - 2001. - 3 : Vol. 39. - pp. 529-545.

**Hu Y.-H., Lee H. B. and Scarpace F. L.** Optimal linear spectral unmixing [Journal] // Geoscience and Remote Sensing, IEEE Transactions . - 1999. - 1 : Vol. 37. - pp. 639-644.

Hyperspectral Remote Sensing Scenes [Online]. - [http://www.ehu.es/ccwintco/index.php/Hyperspectral\\_Remote\\_Sensing\\_Scenes](http://www.ehu.es/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes).

**Ifarraguerri A. and Chang C.-I.** Multispectral and hyperspectral image analysis with convex cones [Journal] // IEEE Trans. Geosci. and Remote Sens.. - 1999. - 2 : Vol. 37. - pp. 756-770.

**Iordache M.-D., Bioucas-Dias J. M. and Plaza A.** Sparse Unmixing of Hyperspectral Data [Journal] // Geoscience and Remote Sensing, IEEE Transactions on . - 2011. - 6 : Vol. 49. - pp. 2014-2039.

**Jia X. [et al.]** Controlled spectral unmixing using extended Support Vector Machines [Journal] // Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2010 2nd Workshop on . - 2010. - pp. 1-4.

**Jóhannesson B. and Giria N.** On approximations involving the beta distribution [Journal] // Communications in Statistics - Simulation and Computation. - 1995. - 2 : Vol. 24. - pp. 489-503.

**Jolliffe I. T.** Principle Component Analysis [Book]. - New York : Springer-Verlag, 1986.

**Keshava N. and Mustard J. F.** Spectral Unmixing [Journal] // Signal Processing Magazine, IEEE . - Jan 2002. - 1 : Vol. 19. - pp. 44-57.

**Keshava N.** A survey of spectral unmixing algorithms [Journal] // Lincoln Laboratory J.. - 2003. - 1 : Vol. 14. - pp. 55–77.

**Krezhova D. and Kirova E.** Hyperspectral remote sensing of the impact of environmental stresses on nitrogen fixing soybean plants (*Glycine max L.*) [Journal] // Recent Advances in Space Technologies (RAST), 2011 5th International Conference on . - 2011. - pp. 172-177.

**Krinidis S. and Chatzis V.** A robust fuzzy local information C-Means clustering algorithm [Journal] // Image Processing, IEEE Transactions on. - 2010. - 5 : Vol. 19. - pp. 1328-1337.

**Kullback S. and Leibler R.A.** On Information and Sufficiency [Journal] // Annals of Mathematical Statistics . - 1 : Vol. 22. - pp. 79-86.

**Landgrebe D.** Hyperspectral image data analysis [Journal] // Signal Processing Magazine, IEEE. - 2002. - 1 : Vol. 19. - pp. 17-28.

**Landgrebe D. and Biehl L.** Hyperspectral Analysis Example of Urban Data [Online]. - <https://engineering.purdue.edu/~landgreb/Hyperspectral.Ex.html>.

**Landgrebe D. and Biehl L.** Hyperspectral Images [Online] // MultiSpec. - <https://engineering.purdue.edu/~biehl/MultiSpec/hyperspectral.html>.

**Lawson C. L. and Hanson R. J.** Solving least squares problems [Book]. - [s.l.] : Society for Industrial and Applied Mathematics, 1995.

**Li H. [et al.]** Pixel-Unmixing Moderate-Resolution Remote Sensing Imagery Using Pairwise Coupling Support Vector Machines: A Case Study [Journal] // Geoscience and Remote Sensing, IEEE Transactions on. - 2011. - 11 : Vol. 49. - pp. 4298-4307.

**Li X., Wang L. and Jia X.** Spectral unmixing based on improved extended support vector machines [Journal] // Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International. - July 2012. - pp. 4118-4121.

**Liu M.-Y. [et al.]** Entropy Rate Superpixel Segmentation [Journal] // Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition. - [s.l.] : IEEE Computer Society, 2011. - pp. 2097--2104.

**Martin M. E. [et al.]** Development of an advanced hyperspectral imaging(HSI) system with applications for cancer detection [Journal] // Annals of Biomedical Engineering. - 2006. - 6 : Vol. 34. - pp. 1061-1068.

**Martinez W. L. and Martinez A. R.** Computational Statistics Handbook with MATLAB [Book Section]. - [s.l.] : Chapman & Hall/CRC, 2007.

**Mianji F. A. and Zhang Y.** SVM-Based Unmixing-to-Classification Conversion for Hyperspectral Abundance Quantification [Journal] // Geoscience and Remote Sensing, IEEE Transactions on. - 2011. - 11 : Vol. 49. - pp. 4318-4327.

**Miao L. and Qi H.** Endmember Extraction from Highly Mixed Data Using Minimum Volume Constrained Nonnegative Matrix Factorization [Journal] // Geoscience and Remote Sensing, IEEE Transactions on. - 2007. - 3 : Vol. 45. - pp. 765-777.

**Nascimento J. M. P. and Bioucas-Dias J. M.** Does independent component analysis play a role in unmixing hyperspectral data? [Journal] // Geoscience and Remote Sensing, IEEE Transactions on. - 2005. - 1 : Vol. 43. - pp. 175-187.

**Nascimento J. M. P. and Bioucas-Dias J. M.** Vertex Component Analysis: A Fast Algorithm to Unmix Hyperspectral Data [Journal] // Geoscience and Remote Sensing, IEEE Transactions on. - Apr. 2005. - 4 : Vol. 43. - pp. 898-910.

**Nemhauser G. L., Wolsey L. A. and Fisher M. L.** An analysis of approximations for maximizing submodular set functions [Journal] // Mathematical Programming. - 1978. - Vol. 14. - pp. 265-294.

**Nocedal J. and Wright S.** Numerical Optimization [Book Section]. - Berlin : Springer-Verlag, 2006.

**Pearson K.** The Problem of the Random Walk [Journal] // Nature. - 1905. - Vol. 72. - p. 294.

**Pham-Gia T. and Turkkana N.** Distribution of the linear combination of two general beta variables and applications [Journal] // Communications in Statistics - Theory and Methods. - 1998. - 7 : Vol. 27. - pp. 1851-1869.

**Platt J.** Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods [Journal] // Advances in Large Margin Classifiers / ed. Smola A. J. [et al.]. - Cambridge : MIT Press, 1999.

**Plaza A. [et al.]** A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data [Journal] // Geoscience and Remote Sensing, IEEE Transactions on. - 2004. - 3 : Vol. 42. - pp. 650-663.

**Plaza A. [et al.]** Spatial/spectral endmember extraction by multidimensional morphological operations [Journal] // Geoscience and Remote Sensing, IEEE Transactions on. - 2002. - 9 : Vol. 40. - pp. 2025-2041.

**Qi J. [et al.]** Spatial and temporal dynamics of vegetation in the San Pedro River basin area [Journal] // Agricultural and Forest Meteorology. - Nov 2000. - 1-3 : Vol. 105. - pp. 55-68.

**Rabe A., van der Linden S. and Hostert P** Simplifying Support Vector Machines for Regression analysis of hyperspectral imagery [Journal] // Hyperspectral Image and Signal Processing: Evolution in Remote Sensing, 2009. WHISPERS '09. First Workshop on. - 2009. - pp. 1-4.

**Raksuntorn N. and Du Q.** A New Linear Mixture Model for Hyperspectral Image Analysis [Journal] // Geoscience and Remote Sensing Symposium, 2008. IGARSS 2008. IEEE International. - 2008. - Vol. 3. - pp. 258-261.

**Ren X.** Superpixel: Empirical Studies and Applications [Online]. - 2004. - <http://ttic.uchicago.edu/~xren/research/superpixel/>.

**Roberts D. A. [et al.]** Mapping Chaparral in the Santa Monica Mountains Using Multiple Endmember Spectral Mixture Models [Journal] // Remote Sensing of Environment. - 1998. - Vol. 65. - pp. 267-279.

**Roberts D. A. [et al.]** Mapping the Spectral Variability in Photosynthetic and Non-Photosynthetic Vegetation, Soils and Shade using AVIRIS [Journal] // Summaries 3rd Annual JPL Airborne Geoscience Workshop. - Pasadena, CA. : [s.n.], 1992. - Vols. 1, AVIRIS. - pp. 38-40.

**Roberts D. A. [et al.]** Evaluation of the potential of Hyperion for fire danger assessment by comparison to the Airborne Visible/Infrared Imaging Spectrometer [Journal] // Geoscience and Remote Sensing, IEEE Transactions on . - 2003. - 6 : Vol. 41. - pp. 1297-1310.

**Rogge D. M. [et al.]** Integration of spatial-spectral information for the improved extraction of endmembers [Journal] // Remote Sensing of Environment. - 2007. - 3 : Vol. 110. - pp. 287-303.

**Shimabukuro Y. E. and Smith J. A.** The Least-Squares Mixing Models to Generate Fraction Images Derived From Remote Sensing Multispectral Data [Journal] // Geoscience and Remote Sensing, IEEE Transactions on . - 1991. - 1 : Vol. 29. - pp. 16-20.

**Singer R. B. and McChord T. B.** Mars: Large scale mixing of bright and dark surface materials and implications for analysis of spectral reflectance [Journal] // Proc. Luna and Planetary Sci. Conf.. - 1979. - pp. 1835-1848.

**Smith M. O., Adams J. B. and Sabol D. E.** Spectral Mixture Analysis - New Strategies for the Analysis of Multispectral Data [Book Section] // Imaging Spectrometry — a Tool for Environmental Observations / book auth. Hill Joachim and Mégier Jacques. - [s.l.] : Springer Netherlands, 1994. - Vol. 4.

**Somers B. [et al.]** Automated Extraction of Image-Based Endmember Bundles for Improved Spectral Unmixing [Journal] // Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of . - 2012. - 2 : Vol. 5. - pp. 396-408.

**Song C.** Spectral mixture analysis for subpixel vegetation fractions in the urban environment: How to incorporate endmember variability? [Journal] // Remote Sensing of Environment. - 2005. - 2 : Vol. 95. - pp. 248 – 263.

Spectral Mixture Analysis-Landscape Toolbox Wiki [Online] // The Landscape Toolbox: Tools and Methods for Effective Rangeland Management. -  
[http://wiki.landscapetoolbox.org/doku.php/remote\\_sensing\\_methods:spectral\\_mixture\\_analysis](http://wiki.landscapetoolbox.org/doku.php/remote_sensing_methods:spectral_mixture_analysis).

**Statistics Toolbox Matlab** Matlab Statistics Toolbox [Book Section]. - 2013.

**Stein D.** Application of the normal compositional model to the analysis of hyperspectral imagery [Journal] // Advances in Techniques for Analysis of Remotely Sensed Data, 2003 IEEE Workshop on. - 2003. - pp. 44-51.

**Theseira M. A. [et al.]** Sensitivity of mixture modeling to endmember selection [Journal] // International Journal of Remote Sensing. - 2003. - 7 : Vol. 24. - pp. 1559 – 1575.

**Thomas J. A. and Cover T. M.** Elements of information theory [Book]. - [s.l.] : John Wiley, 1991. - 2nd.

**Tits L., Somers B. and Coppin P.** The Potential and Limitations of a Clustering Approach for the Improved Efficiency of Multiple Endmember Spectral Mixture Analysis in Plant Production System Monitoring [Journal] // Geoscience and Remote Sensing, IEEE Transactions on. - 2012. - 6 : Vol. 50. - pp. 2273-2286.

**Tompkins S. [et al.]** Optimization of endmembers for spectral mixture analysis [Journal] // Remote Sensing of Environment. - 1997. - Vol. 59. - pp. 472 – 489.

**Torres-Madronero M. C. [et al.]** Multi-temporal unmixing analysis of Hyperion images over the Guanica Dry Forest [Conference] // Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2011 3rd Workshop on. - [s.l.] : IEEE, 2011. - pp. 1-4.

**Wan B. and Fu Z.** Study on atmospheric correction and retrieval of chlorophyll-a from East lake using Hyperion hyperspectral image [Journal] // Remote Sensing, Environment and Transportation Engineering (RSETE), 2011 International Conference on. - 2011. - pp. 2534-2537.

**Wang L. and Jia X.** Integration of Soft and Hard Classifications Using Extended Support Vector Machines [Journal] // Geoscience and Remote Sensing Letters, IEEE. - 2009. - 3 : Vol. 6. - pp. 543-547.

**Wilk M. B. and Gnanadesikan R.** Probability Plotting Methods for the Analysis of Data [Journal] // Biometrika. - 1968. - Vol. 55. - pp. 1-17.

**Winter M. E.** N-FINDR: An Algorithm for Fast Autonomous Spectral Endmember Determination in Hyperspectral Data [Conference] // the Thirteenth International Conference on Applied Geologic Remote Sensing. - Vancouver : [s.n.], 1999. - Vol. 3753. - pp. 266-277.

**Wu T.-F., Lin C.-J. and Weng R. C.** Probability Estimates for Multi-class Classification by Pairwise Coupling [Journal] // Journal of Machine Learning Research. - 2004. - pp. 975--1005.

**Youngentob K. N. [et al.]** Mapping two Eucalyptus subgenera using multiple endmember spectral mixture analysis and continuum-removed imaging spectrometry data [Journal] // Remote Sens. Environ. - 2011. - 5 : Vol. 115. - pp. 1115-1128.

**Zare A. [et al.]** Bootstrapping for Piece-wise Convex Endmember Distribution Detection [Journal] // 4th IEEE Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing. - 2012.

**Zare A. [et al.]** Spectral Unmixing Using the Beta Compositional Model [Journal] // Workshop on Hyperspectral Image and Signal Processing. - 2013. - In Press.

**Zare A. [et al.]** Sub-pixel target spectra estimation and detection using functions of multiple instances [Journal] // Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS), 2011 3rd Workshop on . - June 2011. - pp. 1-4. 6-9.

**Zare A. and Gader P.** An Investigation of Likelihoods and Priors for Bayesian Endmember Estimation [Journal] // AIP Conference Proceedings. - 2010. - 1 : Vol. 1305. - pp. 311-318.

**Zare A. and Gader P.** Pattern Recognition using Functions of Multiple Instances [Journal] // Pattern Recognition (ICPR), 2010 20th International Conference on. - Aug 23-26, 2010. - pp. 1092-1095.

**Zare A. and Ho D.** Endmember Variability in Hyperspectral Analysis [Journal] // IEEE Signal Processing Magazine. - In Press.

**Zare A. and Gader P.** Sparsity promoting iterated constrained endmember detection for hyperspectral imagery [Journal] // IEEE Geosci. Remote Sens. Lett.. - 2007. - 3 : Vol. 4. - pp. 446-450.

**Zare A., Gader P. and Casella G.** Sampling Piece-wise Convex Unmixing and Endmember Extraction [Journal] // Geoscience and Remote Sensing, IEEE Transactions on. - 2012. - 99 : Vol. PP. - pp. 1-11.

## VITA

Xiaoxiao Du received her Bachelor of Engineering degree in Zhejiang University, China with Chu Kochen Honors Certificate in June, 2011. She is a Graduate Research Assistant in the Electrical and Computer Engineering department, University of Missouri-Columbia with Dr. Alina Zare. She is also awarded “Outstanding MS Student” in Missouri Honors Award, 2013.