

**MAP-GUIDED HYPERSPECTRAL IMAGE SUPERPIXEL SEGMENTATION  
USING SEMI-SUPERVISED PARTIAL MEMBERSHIP  
LATENT DIRICHLET ALLOCATION**

---

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

---

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

---

by  
HAO SUN  
Dr. Alina Zare, Thesis Advisor  
December 2016

© Copyright by Hao Sun 2016

All Rights Reserved

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

**MAP-GUIDED HYPERSPECTRAL IMAGE SUPERPIXEL SEGMENTATION  
USING SEMI-SUPERVISED PARTIAL MEMBERSHIP  
LATENT DIRICHLET ALLOCATION**

presented by Hao Sun,

a candidate for the degree of Master of Science and hereby certify that, in their opinion, it is worthy of acceptance.

---

Dr. Alina Zare

---

Dr. Zhihai He

---

Dr. Junwu Tu

## **ACKNOWLEDGMENTS**

I sincerely want to thank Dr. Alina Zare as my advisor, for her patient guidance and full support during my master research. I also want to thank my master thesis committee Dr. Zhihai He and Dr. Junwu Tu for all their good suggestions during the defense.

I also really appreciate the patient discussions and suggestions from my lab-mates especially Changzhe Jiao, Chao Chen and Sheng Zou.

My biggest motivation to finish my graduate studies is from my parents and girlfriend. They deserve the most sincere thanks.

## TABLE OF CONTENTS

<b>ACKNOWLEDGMENTS . . . . .</b>	<b>ii</b>
<b>LIST OF TABLES . . . . .</b>	<b>v</b>
<b>LIST OF FIGURES . . . . .</b>	<b>vi</b>
<b>ABSTRACT . . . . .</b>	<b>viii</b>
<b>CHAPTER</b>	
<b>1 Introduction . . . . .</b>	<b>1</b>
1.1 Hyperspectral Imaging . . . . .	1
1.2 Problem Statement . . . . .	4
<b>2 Literature Review . . . . .</b>	<b>7</b>
2.1 Hyperspectral Unmixing Algorithms . . . . .	7
2.1.1 Pure Pixel-Based Endmember Extraction Method . . . . .	8
2.1.2 Minimum Volume-Based Spectral Unmixing Method . . . . .	9
2.1.3 Sparse Endmember Estimation Method . . . . .	11
2.1.4 Semi-Supervised Partial Membership Latent Dirichlet Allocation . .	13
2.2 Hyperspectral Image Segmentation Algorithms . . . . .	17
2.2.1 Working in the spatial domain . . . . .	17
2.2.2 Working in the spectral domain . . . . .	20
2.2.3 Combining spatial-based and spectral-based segmentation . . . . .	24
2.3 Cluster Validity . . . . .	32

2.3.1	The Dunn and Dunn-like Indices . . . . .	32
2.3.2	Davies-Bouldin Index . . . . .	33
2.3.3	Silhouette Index . . . . .	35
<b>3</b>	<b>Proposed Methods . . . . .</b>	<b>37</b>
3.1	Methodology . . . . .	37
3.2	Initial Superpixel Segmentation Based On Map-Guided Hyperspectral SLIC	38
3.3	Hyperspectral Unmixing Based On Semi-Supervised PM-LDA . . . . .	44
3.4	Superpixel Segmentation Based On Proportion Maps . . . . .	46
3.5	Evaluation . . . . .	47
<b>4</b>	<b>Experiment . . . . .</b>	<b>50</b>
4.1	HySens Pavia University Data . . . . .	50
4.1.1	Data Set Description . . . . .	50
4.1.2	Superpixel Segmentation Results . . . . .	51
4.2	MUUFL Gulfport Hyperspectral Data . . . . .	65
4.2.1	Data Set Description . . . . .	65
4.2.2	Superpixel Segmentation Results . . . . .	66
<b>5</b>	<b>Conclusion and Future Work . . . . .</b>	<b>77</b>
	<b>BIBLIOGRAPHY . . . . .</b>	<b>79</b>

## LIST OF TABLES

Table	Page
4.1 Dunn, Davies-Bouldin, Silhouette indices and standard deviation on Pavia University . . . . .	55
4.2 Modified Dunn, Davies-Bouldin indices on Pavia University . . . . .	56
4.3 Dunn, Davies-Bouldin, Silhouette indices and standard deviation on Pavia University when change parameter $k$ . . . . .	57
4.4 Modified Dunn, Davies-Bouldin indices on Pavia University and standard deviation on Pavia University when change parameter $k$ . . . . .	58
4.5 Dunn, Davies-Bouldin, Silhouette indices and standard deviation on Gulfport	70
4.6 Modified Dunn, Davies-Bouldin indices and standard deviation on Gulfport	70
4.7 Dunn, Davies-Bouldin, Silhouette indices and standard deviation on Gulfport when change parameter $k$ . . . . .	72
4.8 Modified Dunn, Davies-Bouldin indices and standard deviation on Gulfport when change parameter $k$ . . . . .	72

## LIST OF FIGURES

Figure	Page
1.1 Pavia University data set [3] (left: RGB images; right: reflectance from all wavelengths) . . . . .	2
2.1 Graphical model of PM-LDA . . . . .	14
3.1 The Flowchart of Proposed Method . . . . .	38
3.2 OpenStreetMap on Pavia University . . . . .	40
3.3 Nodes and Ways from OpenStreetMap on Pavia University . . . . .	41
3.4 Overlap OpenStreetMap information with the image of Pavia University . .	43
4.1 RGB Image of Pavia University . . . . .	52
4.2 Superpixels on Pavia University . . . . .	59
4.3 Estimated proportion maps using semi-supervised PM-LDA on Pavia University . . . . .	60
4.4 Superpixel Segmentation results on Pavia University . . . . .	61
4.5 Zoom in to painted metal sheets area on superpixel segmentation results . .	62
4.6 Zoom in to grey building area on superpixel segmentation results . . . . .	63

4.7	Superpixel segmentation results on Pavia University when change parameter $k$ . . . . .	64
4.8	RGB Image of Gulfport Data . . . . .	65
4.9	Superpixels on Gulfport . . . . .	66
4.10	Estimated proportion maps for semi-supervised PM-LDA on Gulfport . . . . .	67
4.11	Superpixel Segmentation on Gulfport with 6 algorithms . . . . .	73
4.12	Zoom in to the red roof building area on superpixel segmentation results . . . . .	74
4.13	Zoom in to another red roof building area on superpixel segmentation results . . . . .	75
4.14	Superpixel segmentation results on Gulfport when change parameter $k$ . . . . .	76

## ABSTRACT

Many superpixel segmentation algorithms which are suitable for the regular color images like images with three channels: red, green and blue (RGB images) have been developed in the literature. However, because of the high dimensionality of hyperspectral imagery, these regular superpixel segmentation algorithms often do not perform well in hyperspectral imagery. Although there are some authors who have modified some regular superpixel segmentation algorithms to fit the hyperspectral image, many still underperform on complex data.

In this thesis, to solve this problem, we introduce a hyperspectral unmixing based superpixel segmentation that leverages map information. We call this approach map-guided semi-supervised PM-LDA superpixel segmentation. The approach uses auxilliary map information to guide segmentation. The approach also leverages spectral unmixing results to provide improved results compared with segmentation based on raw data.

We test our proposed method on two real hyperspectral data, University of Pavia and MUUFL Gulfport Hyperspectral Data. In these experiments, our proposed method achieves better results compared to other state-of-the-art algorithms. We also develop new cluster validity metrics to evaluate the results.

# **Chapter 1**

## **Introduction**

Many machine learning algorithms have been proposed and applied to hyperspectral imagery. In this thesis, hyperspectral image superpixel segmentation problems are developed. Superpixel segmentation is often used as a preprocessing step for machine learning method.

### **1.1 Hyperspectral Imaging**

Spectral imaging belongs to a domain of spectroscopy and photography, which refers to collecting and processing spectral information over a range of the electromagnetic spectrum. In conventional photography, an image is captured by measuring reflectance on three wavelengths, which are red, green and blue. In spectral imagery, each pixel has its reflectance captured on hundreds of narrow contiguous wavelengths bands. The information from all wavelengths bands can be used to do analysis tasks on the scene, which may be capable with the human eyes or standard RGB imagery.

Hyperspectral and multispectral are two categories in spectral imaging. Hyperspectral

images (HSI) have more bands (often, hundreds) and a higher spectral resolution when compared to multispectral images. Hyperspectral images are often captured by imaging spectrometers, such as HYDICE (Hyperspectral Digital Imagery Collection Experiment) [1] and NASA JPL's AVIRIS (Airborne Visible InfraRed Imaging Spectrometer) [2].

In HSI, hyperspectral sensors collect radiance values from a scene across many wavelengths of the electromagnetic spectrum, and these values are organized into planes forming a three dimensional data cube. Every element in this cube represents the reflectance or radiance values at a particular wavelength at a pixel location. Each spectral vector corresponds to the reflectance value at a particular pixel across all spectral wavelengths. Fig 1.1 illustrates the concept of hyperspectral imaging.

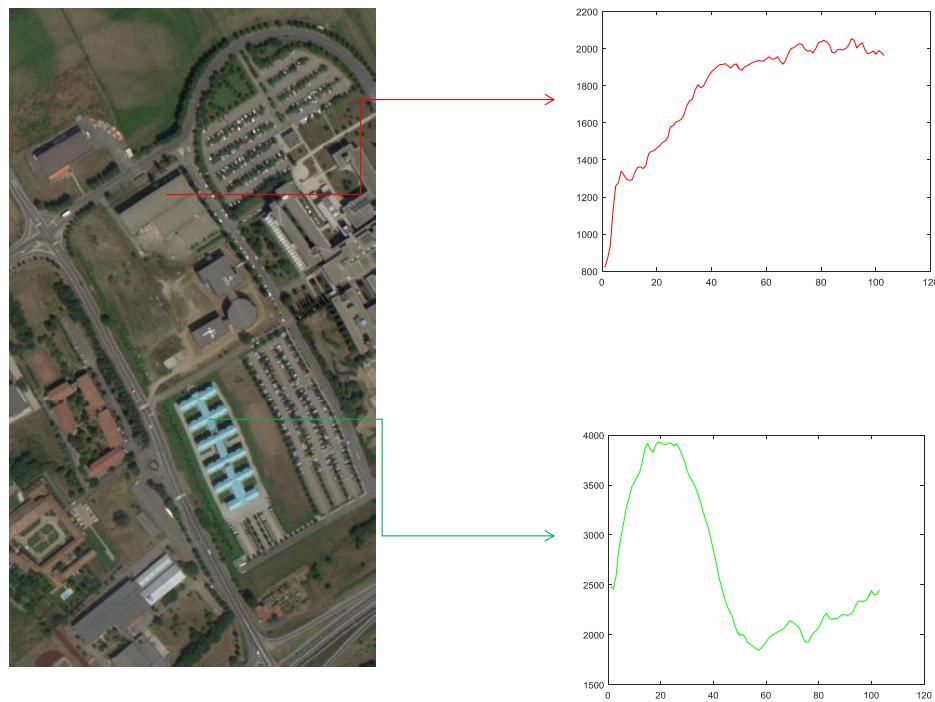


Figure 1.1: Pavia University data set [3] (left: RGB images; right: reflectance from all wavelengths)

HSI has been used for applications as varied as space exploration [4, 5], geology and medical image analysis [6, 7], archeology [8, 9].

The analysis of hyperspectral images often consists of endmember estimation and hyperspectral unmixing [10]. Endmember estimation aims at estimating the pure spectral signatures of materials, which are also called endmembers in the image [11]. It is assumed that each pixel in the hyperspectral image is combined by the spectral signatures of different endmembers weighted by corresponding abundances (also called proportions). With this assumption, hyperspectral unmixing here aims at estimation the proportion of each endmember for each pixel.

Hyperspectral unmixing depends on the assumption of a mixing model which can be categorized into two ways [12, 13]: linear mixing model and non-linear mixing model.

Singer first applied the linear mixing model (LMM) when the scale of mixing is macroscopic and the incident light only interacts with one material [14]. Assuming the scattering of light between different endmembers can be ignored, the reflectance value captured by the sensor is the linear combination of the spectral signature of every endmember with its corresponding proportion values respectively. This means each pixel value in the hyperspectral image is the convex combination of endmembers and proportions. LMM model can be presented mathematically as:

$$\mathbf{x}_i = \sum_{k=1}^M p_{ik} \mathbf{e}_k + \epsilon_i, i = 1, \dots, N, \quad (1.1)$$

where  $\mathbf{x}_i$  is the spectral measurement at pixel  $i$ ,  $N$  is the total number of pixels,  $M$  is the number of endmembers in the image,  $p_{ik}$  is the proportion of endmember  $k$  at pixel  $i$ ,  $\mathbf{e}_k$  is the  $k^{th}$  endmember and  $\epsilon_i$  is the error term for pixel  $i$ . Under this model, two constraints (non-negative and sum-to-one) for the proportion are assumed:

$$p_{ik} \geq 0, k = 1, \dots, M \quad (1.2)$$

$$\sum_{k=1}^M p_{ik} = 1. \quad (1.3)$$

The LMM assumes that all pixels in the image are included in the convex hull. There are many unmixing methods in the literature, including geometrical, statistical, sparse regression-based and spatial-spectral contextual information-based approaches which were reviewed for hyperspectral endmember extraction and unmixing under the linear mixing model [13, 15].

In a more realistic situation, uneven scattering of light between different materials could be captured by the sensor. In this case, non-linear mixing models are more appropriate. The non-linear interaction occurs between multiple layers which the radiation light will be reflected from different materials bouncing from to another materials before it is received by the sensor. On the other hand, there can be intimate interactions when the surface is mixed homogeneously and the interactions of light will be reflected at a molecular level [15].

The work proposed here assumes the LMM.

## 1.2 Problem Statement

Image segmentation is a computer vision task which aims at partitioning an image into homogenous regions or semantically meaningful regions without crossing edges or sharp contours in the image. There are many state-of-the-art image segmentation algorithms.

Most of them have been developed for and only applied to gray scale or RGB images.

A huge amount of hyperspectral data will be collected by the new generation of hyperspectral sensors. Many fields of knowledge will benefit the potential applications of hyperspectral imagery. Applying most traditional image processing method to HSI will be a big challenge due to the increased dimensionality of the data.

Often, to perform the segmentation of a hyperspectral image, dimension reduction is applied, both for reducing the influence of noise and accelerating the segmentation algorithm. Principal component analysis (PCA) based methods have been proposed and widely applied in hyperspectral image segmentation algorithms [16]. The components with significant variance are selected as the features during the segmentation process. However, there are two problems for a PCA-based approach. The first one is determining the number of components is not easy. The reason for this is the variance of the useful components could be similar with the variance of the noise. Another problem is that the components obtained by PCA do not necessarily correspond to any physical meaning. The components returned by PCA could be noisy. An alternative approach to reduce the dimensionality is to use the proportion maps (also called abundance maps). The proportion maps, which are obtained from hyperspectral unmixing, contain the chemical species present in the image [17].

Many unsupervised unmixing algorithms, like vertex component analysis (VCA) [18] and Sparsity Promoting Iterative Constraint Endmembers (SPICE) [19], can be applied to obtain the abundance maps. Also, many state-of-the-art supervised hyperspectral unmixing methods reviewed in [20] rely on the prior knowledge for endmember signatures, either from spectral library or an endmember extraction algorithm. In this case, the hyperspectral unmixing performance is highly impacted by the spectral library or endmember extraction

method selected. However, the amount of reference information is generally limited in practice.

In this thesis, we propose to use a hyperspectral unmixing algorithm called Semi-supervised PM-LDA to obtain the proportion maps of endmembers. A few limited tags extracted from the crowd-sourced media OpenStreetMap can help us to guide the unmixing process and improve results. Using the proportion maps as a reduced dimensionality representation of the imagery, we then perform superpixel segmentation. The details will be discussed in Chapter 3.

# **Chapter 2**

## **Literature Review**

This chapter provides a review of some existing hyperspectral unmixing methods and hyperspectral image segmentation algorithms.

### **2.1 Hyperspectral Unmixing Algorithms**

This section briefly reviews some existing hyperspectral unmixing methods. The aim of unmixing is to estimate the pure spectral signature of every endmember and the proportion values for each pixel. To achieve this, there are generally three categories methods: pure pixel-based endmember extraction methods, minimum volume-based spectral unmixing methods and sparse endmember estimation methods. A representative unmixing algorithm for each method will be discussed in this section. We also will discuss a hyperspectral unmixing algorithm called Partial membership latent Dirichlet allocation (PM-LDA)[21]. We will make a use of extension of this algorithm in Chapter 3 for our proposed method to perform unmixing.

### 2.1.1 Pure Pixel-Based Endmember Extraction Method

Pure pixel-based endmember extraction methods assume that at least one pixel existing in the image is a pure example of each endmember. Algorithms of this type rely on the fact that endmembers in the hyperspectral image can be detected without performing spectral unmixing. Some representative algorithms following the assumption include Pixel Purity Index (PPI)[22], Iterative Error Analysis (IEA)[23], N-FINDR [24] and Vertex Component Analysis (VCA) [18]. VCA is a very widely used approach and have been applied to perform unmixing for hyperspectral images. It is reviewed in this section as a representative algorithm.

VCA assumes the existence of pure pixels for every endmember in the hyperspectral images. This algorithm exploits two facts. The first one is the endmembers are presented as the vertices of a simplex. The second one is the affine transformation of a simplex is still a simplex.

Assuming the linear mixing model for a hyperspectral image, the hyperspectral data is modeled by a convex cone. The projective projection of the convex cone onto a properly chosen hyperplane is a simplex with vertices [18]. Each pixel in the image is projected onto the plane on random direction. In the first iteration, the pixel with the biggest projection is chosen as the first endmember. After identifying the first one, the remaining endmembers are identified by iteratively projecting the data onto a direction orthogonal to the subspace spanned by the endmember determined in the last iteration [25].

The new endmember is identified as the pixel which corresponds to the extreme of the projection. VCA algorithm will iterate until the number of endmembers is exhausted [18]. After all endmembers' signatures are obtained, least squares optimization [26] can be applied to estimate the proportion maps for all endmembers.

## 2.1.2 Minimum Volume-Based Spectral Unmixing Method

Other methods to achieve the unmixing process are the minimum volume-based spectral unmixing methods. They assume the endmember matrix with the enclosed volume is minimized based on the non-negative and sum-to-one proportion constraints mentioned in Chapter 1.

Minimum volume-based spectral unmixing methods do not assume the existence of pure pixels in the image, and they are performed based on optimization. Iterated constraint endmembers (ICE) is a well established approach of this type. ICE is an optimization-based method that tries to minimize the objective function in order to estimate the signatures of endmembers and the corresponding proportion maps [27].

The ICE algorithm optimizes of the Residual Sum of Squared (RSS), which is a measure of discrepancy between the input data and the estimated signatures of the endmembers and the corresponding proportions based on the linear mixing model. It is given below:

$$RSS = \sum_{i=1}^N (\mathbf{x}_i - \sum_{k=1}^M p_{ik} \mathbf{e}_k)^T (\mathbf{x}_i - \sum_{k=1}^M p_{ik} \mathbf{e}_k). \quad (2.1)$$

Here  $\mathbf{x}$  is the pixel value in the image,  $N$  is the number of total data points,  $M$  is the number of the endmembers and  $p_{ik}$  is the proportion of the  $k^{th}$  endmember  $\mathbf{e}_k$  at  $i^{th}$  pixel [28].

The sum of squared distance (SSD) term is used in the ICE's objective function to constrain the size of the simplex.

$$SSD = \sum_{k=1}^{M-1} \sum_{l=k+1}^M (\mathbf{e}_k - \mathbf{e}_l)^T (\mathbf{e}_k - \mathbf{e}_l). \quad (2.2)$$

After the combination of RSS and SSD, the objective function for ICE algorithm is represented as:

$$RSS_{reg} = (1 - \mu) \frac{RSS}{N} + \mu V, \quad (2.3)$$

where  $\mu$  is a regularization parameter which trades off the above two terms in the equations and the  $V$  here is the sum of variance across the bands of the endmember vertices which can be represented as:

$$V = \frac{SSD}{M}(M - 1). \quad (2.4)$$

The range of the regularization parameter  $\mu$  is between 0 and 1; it has a significant influence on the estimation. When  $\mu = 0$ , the optimization problem is ill-defined, and when  $\mu = 1$ , all estimated endmembers collapse to a single point close to the mean of the data.

The ICE algorithm performs an iterative process for minimizing both endmembers and the proportions for all pixels in the image. Endmembers are selected randomly in the image in the first iteration. A separate minimization of all  $N$  terms is used to estimate the proportion maps. Each minimization is a quadratic programming problem. Endmembers are estimated after solving the proportion estimation by the equations given below:

$$\mathbf{e}_t = (\mathbf{P}^T \mathbf{P} + \lambda(\mathbf{I}_M - \frac{\mathbf{1} * \mathbf{1}^T}{M}))^{-1} \mathbf{P}^T \mathbf{x}_t, \quad (2.5)$$

where  $\mathbf{e}_t$  is the signature of the endmember vector values in the  $t^{th}$  band,  $\mathbf{P}$  is the proportion matrix with the size as  $N \times M$ ,  $\mathbf{I}_M$  is the  $M \times M$  identity matrix,  $\mathbf{1}$  is the one-vector with size as  $M$ , and  $\mathbf{x}_t$  is the pixel value in the image data [19]. Here  $\lambda$  is given by  $\lambda = \frac{N_\mu}{(M-1)(1-\mu)}$ .

### 2.1.3 Sparse Endmember Estimation Method

Many hyperspectral unmixing methods require a number of endmembers in the image as the input. However, it is a big challenge to estimate or choose the number of endmembers for hyperspectral images. If the number of endmembers is set to a value smaller than the true value, then some signatures we learn will be mixed together or some endmembers will be missed. If the number is set to a value bigger than the true number, then the endmember estimation algorithm may return unnecessary signatures that do not correspond to any physically relevant materials. Sparse endmember estimation methods have been developed to attempt to address these situations.

One algorithm that belongs to this area is the sparsity promoting iterative constrained endmembers (SPICE) algorithm proposed by Zare et al [19]. The SPICE algorithm is an extension of ICE algorithm we mentioned before. Based on the objective function proposed in ICE algorithm, a sparsity promoting term is added to help prune the unnecessary endmembers [19]. The sparsity promoting term (SPT) is represented as:

$$SPT = \sum_{k=1}^M \gamma_k \sum_{i=1}^N p_{ik}, \quad (2.6)$$

where  $\gamma_k$  is given by:

$$\gamma_k = \frac{\Gamma}{\sum_{i=1}^N p_{ik}^{old}}, \quad (2.7)$$

where  $N$  is the number of total data points,  $M$  is the number of the endmembers,  $p_{ik}$  is the proportion of the  $k^{th}$  endmember  $e_k$  at  $i^{th}$  pixel,  $p_{ik}^{old}$  is the proportion value in last iteration and  $\Gamma$  is a constant parameter which determines the degree of which the proportions are driven to 0. The proportion values of unnecessary endmembers are driven to 0 due to the

sparsity term. Thus, the the endmembers with the corresponding proportion values are 0 will be pruned, then the sparse set of endmember is formed.

The objective function is combined by the SPT term and the objective function from ICE algorithm which is given by:

$$RSS_{reg} = (1 - \mu) \frac{RSS}{N} + \mu V + SPT. \quad (2.8)$$

The same iterative processes are applied in the SPICE algorithm to estimate the end-members and the corresponding proportion values. After the signatures of endmembers are estimated, the proportion maps can be achieved by the least squares optimization by using the quadratic programming method.

Another representative method is Sparse Endmember Extraction and Demixing which was proposed in [29]. Each data point  $\mathbf{x}_i$  is assigned a weight, and the data points with higher weights are chosen as the potential endmembers. The optimization function is given as:

$$w_{i,.} = \arg \min_{\tilde{w}} \|\mathbf{x}_i - \sum_{j=1}^n \tilde{w}_j \mathbf{x}_j\|_{l_2}^2 + \gamma_i \|\tilde{\mathbf{w}}\|_{l_0}, \quad (2.9)$$

subject to the constraints as:

$$\tilde{w}_j \geq 0, \tilde{w}_i = 0. \quad (2.10)$$

Here  $w_{ij}$  represents the weight for the corresponding pixel,  $n$  is the number of data points. The first term in the objective function refers to the reconstruction error term, and the second one is the sparse penalty term.  $\gamma_i$  is the parameter weights these two terms. Higher value of  $\gamma_i$  will result in more sparse solution [30].

Data points with weights as zero are pruned, and the remaining points are sorted based

on the relation  $\preceq_0, \preceq_2$ .  $\mathbf{x}_i \preceq_0 \mathbf{x}_j$  is given when  $\|w_{\cdot,i}\|_{l_0} \leq \|w_{\cdot,j}\|_{l_0}$  and  $\mathbf{x}_i \preceq_2 \mathbf{x}_j$  is given when  $\frac{\|w_{\cdot,i}\|_{l_2}}{\|w_{\cdot,i}\|_{l_0}} \leq \frac{\|w_{\cdot,j}\|_{l_2}}{\|w_{\cdot,j}\|_{l_0}}$ .

Two new indices  $r_i^0, r_i^2$  will be created by these two relations. The  $r_i^0$  index ranks the pixel vector based on the number of times a particular vector is represented by the remaining element of  $\mathbf{x}_i$ . The  $r_i^2$  index indicates the importance of  $\mathbf{x}_i$  in the representation. Thus, the final list is created and the endmember set is given by the highest rankings [30]. The proportion matrix  $\alpha_i$  is estimated by:

$$\alpha_{i,\cdot} = \arg \min_{\tilde{\alpha}} \|\mathbf{x}_i - \sum_{j=1}^s \tilde{\alpha}_j \mathbf{e}_j\|_{l_2}^2 + \beta \|\tilde{\alpha}\|_{l_1}, \quad (2.11)$$

where  $\mathbf{e}_j$  is the signature of  $j$ -th endmember,  $s$  is the number of endmembers,  $\beta_i$  is the trade-off parameter and the equation subjects to the constraints as:

$$\tilde{\alpha}_j \geq 0, \sum_{j=1}^s \tilde{\alpha}_j \approx 1. \quad (2.12)$$

### 2.1.4 Semi-Supervised Partial Membership Latent Dirichlet Allocation

In this work, we use Partial Membership Latent Dirichlet Allocation (PM-LDA)[21] to unmix our HSI data. An overview of PM-LDA is provided in this section.

PM-LDA is an extension of Latent Dirichlet Allocation (LDA) modeling in which the words have partial membership in all available topics [31]. The PM-LDA model is a hierarchical Bayesian model in which data is organized at two levels: the word level and the document level, which is illustrated in Fig 2.1.

In PM-LDA model, the random variable associated with a data point is assumed to be

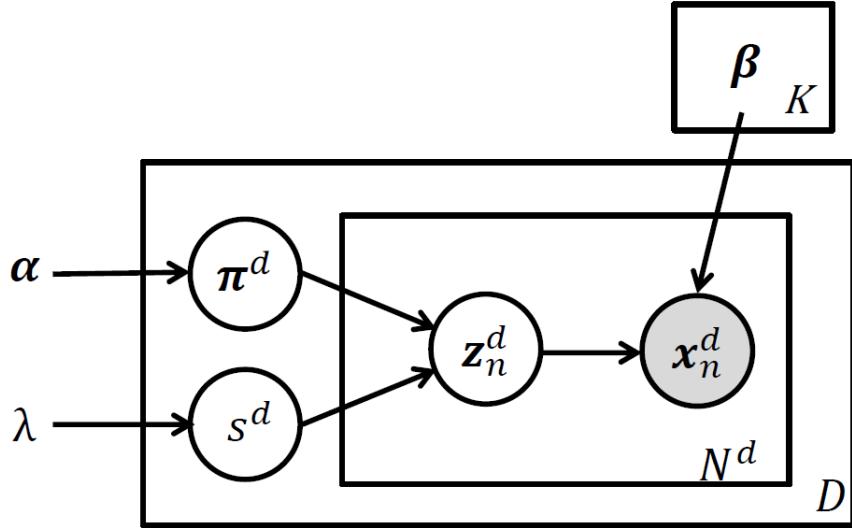


Figure 2.1: Graphical model of PM-LDA

distributed according to several topics with a continuous partial membership in each topic. Mathematically, PM-LDA model can be represented as:

$$p(\boldsymbol{\pi}^d, s^d, \mathbf{z}_n^d, \mathbf{x}_n^d | \boldsymbol{\alpha}, \lambda, \boldsymbol{\beta}) = p(\boldsymbol{\pi}^d | \boldsymbol{\alpha}) p(s^d | \lambda) p(\mathbf{z}_n^d | \boldsymbol{\pi}^d, s^d) \prod_{k=1}^K p_k(\mathbf{x}_n^d | \beta_k)^{z_{nk}^d} \quad (2.13)$$

where  $\mathbf{x}_n^d$  is the  $n$ -th word in document  $d$ ,  $\mathbf{z}_n^d$  is the partial membership vector of  $\mathbf{x}_n^d$ ,  $\boldsymbol{\pi}^d \sim \text{Dir}(\boldsymbol{\alpha})$  and  $s^d \sim \exp(\lambda)$  are the topic proportion and the level of topic mixing in document  $d$ , respectively. The parameter  $\alpha$  gives the topic mixing across a document, and the parameter  $\lambda$  controls how similar the partial membership vector of each word is expected to be the topic distribution of a document.

Given hyperparameters  $\Psi = \{\alpha, \lambda\}$  and the data set (which has been partitioned into

documents),  $\mathbf{X} = \{\mathbf{X}^1, \mathbf{X}^2, \dots, \mathbf{X}^D\}$ , the goal of parameter estimation is to estimate the topic proportion of each document,  $\boldsymbol{\pi}^d$ , the topic mixing level in each document,  $s^d$ , the partial memberships of each word in each topic,  $\mathbf{z}_n^d$ , and the parameters defining the probability distribution of each topic,  $\beta_k$ . This is done using an MCMC sampler [32].

The specific distribution forms chosen in PM-LDA model are:

$$p(\boldsymbol{\pi}^d | \boldsymbol{\alpha}) = \frac{\Gamma\left(\sum_{k=1}^K \alpha_k\right)}{\prod_{k=1}^K \Gamma(\alpha_k)} \prod_{k=1}^K (\pi_k^d)^{(\alpha_k - 1)} \quad (2.14)$$

$$p(s^d | \lambda) = \lambda e^{-\lambda s^d} \quad (2.15)$$

$$p(\mathbf{z}_n^d | \boldsymbol{\pi}^d, s^d) = \frac{\Gamma\left(\sum_{k=1}^K s^d \pi_k^d\right)}{\prod_{k=1}^K \Gamma(s^d \pi_k^d)} \prod_{k=1}^K (z_{nk}^d)^{(s^d \pi_k^d - 1)}, \quad (2.16)$$

An extension to PM-LDA, the semi-supervised PM-LDA has been proposed in [33]. For semi-supervised PM-LDA, the topic proportion for each document is drawn from a Dirichlet distribution with hyperparameter  $\boldsymbol{\alpha}$ . To impose supervision, the author proposes to draw topic proportion from the product of original distribution and a binary endmember label vector, followed by normalization for each pixel [33]. By applying the label vector, the candidate topic proportion vector is restricted to have nonzero entries on the supervised endmembers. The change of target proportion impacts the candidate topic membership (proportions) for each pixel because topic membership is drawn from the Dirichlet distribution of which parameter is the product of scaling factor and topic proportion. Therefore, the topic membership is also confined within supervised endmembers for which the superpixel belongs.

Algorithm 1 summarizes a Metropolis-within-Gibbs sampler to perform parameter estimation for semi-supervised PM-LDA [33].

---

**Algorithm 1** Semi-Supervised PM-LDA [33]

---

**Input:** A corpus  $\mathbf{D}$ , the number of topics  $K$ , hyperparameters  $\Psi = \{\alpha, \lambda\}$ , and the number of iterations  $T$

**Output:** Collection of all samples: topic proportions  $\Pi^{(t)}$ , scaling factors  $\mathbf{S}^{(t)}$ , membership vectors  $\mathbf{M}^{(t)}$ , topic component parameters  $\beta^{(t)}$

- 1: **for**  $t = 1 : T$  **do**
  - 2:   Generate topic prior  $\tau$ : The size of  $\tau$  is  $K$  by  $D$ ,  $\tau_{kd}$  is labeled as 1 if topic  $k$  is found in document  $d$ ; otherwise, it is labeled as 0.
  - 3:   **for**  $d = 1 : D$  **do**
  - 4:     Sample  $\pi^d$ : Draw candidate:  $\pi^\dagger \sim \tau^d \text{Dir}(\alpha)$   
      Accept candidate with probability:  

$$a_\pi = \min \left\{ 1, \frac{p(\pi^\dagger, s^{(t-1)}, \mathbf{Z}^{(t-1)}, \mathbf{X} | \Psi) p(\pi^{(t-1)} | \alpha)}{p(\pi^{(t-1)}, s^{(t-1)}, \mathbf{Z}^{(t-1)}, \mathbf{X} | \Psi) p(\pi^\dagger | \alpha)} \right\}$$
  - 5:     Sample  $s^d$ : Draw candidate:  $s^\dagger \sim \exp(\lambda)$   
      Accept candidate with probability:  

$$a_s = \min \left\{ 1, \frac{p(\pi^{(t)}, s^\dagger, \mathbf{Z}^{(t-1)}, \mathbf{X} | \Psi) p(s^{(t-1)} | \lambda)}{p(\pi^{(t)}, s^{(t-1)}, \mathbf{Z}^{(t-1)}, \mathbf{X} | \Psi) p(s^\dagger | \lambda)} \right\}$$
  - 6:     **for**  $n = 1 : N^d$  **do**
  - 7:       Sample  $\mathbf{z}_n^d$ : Draw candidate:  $\mathbf{z}_n^\dagger \sim \text{Dir}(\mathbf{1}_K)$   
      Accept candidate with probability:  

$$a_z = \min \left\{ 1, \frac{p(\pi^{(t)}, s^{(t)}, \mathbf{z}_n^\dagger, \mathbf{x}_n | \Psi)}{p(\pi^{(t)}, s^{(t)}, \mathbf{z}_n^{(t-1)}, \mathbf{x}_n | \Psi)} \right\}$$
  - 8:     **end for**
  - 9:   **end for**
  - 10:   **for**  $k = 1 : K$  **do**
  - 11:     Sample  $\mu_k$ : Draw proposal:  $\mu_k^\dagger \sim \mathcal{N}(\cdot | \mu_D, \Sigma_D)$   
       $\mu_D$  and  $\Sigma_D$  are mean and covariance of the data  
      Accept candidate with probability:  

$$a_k = \min \left\{ 1, \frac{p(\Pi^{(t)}, \mathbf{S}^{(t)}, \mathbf{M}^{(t)}, \mathbf{D} | \mu_k^\dagger) \mathcal{N}(\mu_k^{(t-1)} | \mu_D, \Sigma_D)}{p(\Pi^{(t)}, \mathbf{S}^{(t)}, \mathbf{M}^{(t)}, \mathbf{D} | \mu_k^{(t-1)}) \mathcal{N}(\mu_k^\dagger | \mu_D, \Sigma_D)} \right\}$$
  - 12:     **end for**
  - 13:     Sample covariance matrices  $\Sigma = \sigma^2 \mathbf{I}$ :  
      Draw candidate from:  $\sigma^2 \sim \text{Unif}(0, u)$   
      with  $u = \frac{1}{2} \{ \max_{\mathbf{x}_n} d^2(\mathbf{x}_n - \mu_D) - \min_{\mathbf{x}_n} d^2(\mathbf{x}_n - \mu_D) \}$   
      Accept candidate with probability:  

$$a_\Sigma = \min \left\{ 1, \frac{p(\Pi^{(t)}, \mathbf{S}^{(t)}, \mathbf{M}^{(t)}, \mathbf{D} | \Sigma^\dagger)}{p(\Pi^{(t)}, \mathbf{S}^{(t)}, \mathbf{M}^{(t)}, \mathbf{D} | \Sigma^{(t-1)})} \right\}.$$
  - 14:   **end for**
-

## **2.2 Hyperspectral Image Segmentation Algorithms**

The statistical analysis of hyperspectral image normally starts with taking measures on each pixel independently or large segmented areas that associated with one scene feature. For pixel level process, the results are sensitive to the instrument noise and the variability within class. For the segment level process, image segmentation can help reduce the noise and variability. However, the segmentation is difficult to perform automatically. In this section, we will summary a few existing hyperspectral image segmentation algorithms, and these techniques can be grouped into three ways: segmentation working in the spatial domain, segmentation working in the spectral domain and combining spatial-based and spectral-based segmentation [34].

### **2.2.1 Working in the spatial domain**

The algorithms working in this domain try to identify spatially connected pixel sets or regions. Some methods like watershed transformation [35] are extended into hyperspectral area.

#### **Segmentation of hyperspectral images based on watershed transformation**

The watershed segmentation was first introduced in 1979 by Beucher and Lantuejoul [36] which has been widely used for color images [37]. The image is treated as a topographic relief and divided into several catchment basins in which each is associated with a minimum in the image. The transformation normally starts with the gradient function of the image, and the gradient values on the edge between the objects and the minimum will be higher. Then we can use the edges to partition the whole image into different “homo-

geneous” regions. The results returned by the watershed algorithm are normally several oversegmentation regions and watershed pixels, which are located on the edges between each region. Some post-processing can be used to generate the fine partition, like merging the neighboring regions.

This method is hard to perform on hyperspectral images that are composed of hundreds of spectral channels. Watershed transformation needs an input of the gradient that has to be a scalar function. When we try to perform this on hyperspectral images, information from all bands should be considered. One scheme of performing watershed transformation on hyperspectral images is proposed in [38].

The general steps to process the watershed transformation on hyperspectral images are: (1) feature extractions; (2) computing the gradient; (3) going through the regular watershed to return a segmentation map.

Let us assume a hyperspectral image have  $B$  bands, and is composed of  $n$  pixel vectors  $\mathbf{x} = \mathbf{x}_j \in \mathbb{R}^B, j = 1, 2, \dots, n$ . The  $B$  bands can be denoted by  $\mathbf{x}_\lambda, \lambda = 1, 2, \dots, B$ .

Before computing the gradient, feature extraction is performed on the original image. This process can be done by one of the transformation like the principal component analysis (PCA) [39], independent component analysis (ICA) [40]. The aim of this step is to reduce the dimensionality that converts the original image into a one-band image and keeps most information to partition the image.

If the result of the dimensionality reduction can obtain a good distinguishing ability, then we can perform the regular watershed algorithm to get the segmentation. However, keeping more bands can carry more information to help the partition. We can still have a multi-band image after the feature extraction step. To process the multi-band image, there are several different methods, which can be grouped into the following ways.

1). The first way is to compute a vectorial gradient. Vectorial gradients measure the distance between pixel vectors, and can convert the hyperspectral image into a one-band gradient. There are several vectorial gradients, like metric-based gradient [41], color morphological gradient [42] and robust color morphological gradient [42] have been proposed.

For a pixel vector  $\mathbf{x}_i$ , denote  $\psi = [\mathbf{x}_i^1, \mathbf{x}_i^2, \dots, \mathbf{x}_i^j]$  as a set of  $j$  vectors which are the neighbors of  $\mathbf{x}_i$  (assume  $\psi$  does not contain  $\mathbf{x}_i$ ). The metric-based gradient is defined as:

$$\nabla_{\psi,d}^{MB}(\mathbf{x}_i) = \sup_{a \in \psi} d(\mathbf{x}_i, \mathbf{x}_i^a) - \inf_{b \in \psi} d(\mathbf{x}_i, \mathbf{x}_i^b). \quad (2.17)$$

Here the function  $d$  represent the distance between two vectors. Euclidean distance is one of the most widely used distances.  $\mathbf{x}_i^a$  and  $\mathbf{x}_i^b$  are two neighboring pixel vectors of  $\mathbf{x}_i$ .

For a pixel vector  $\mathbf{x}_i$ , denote  $\mathcal{X} = [\mathbf{x}_i^1, \mathbf{x}_i^2, \dots, \mathbf{x}_i^j]$  as a set of  $j$  vectors within a structure element  $E$ . The color morphological gradient (CMG)  $\nabla_{\psi,d}^{CM}$  is defined as:

$$\nabla_{\psi,d}^{CM}(\mathbf{x}_i) = \max_{a,b \in \mathcal{X}} d(\mathbf{x}_i^a, \mathbf{x}_i^b). \quad (2.18)$$

The above equation can be rewritten if  $d$  is the Euclidean distance as:

$$\nabla_{\psi,d}^{CM}(\mathbf{x}_i) = \max_{a,b \in \mathcal{X}} \|d(\mathbf{x}_i^a, \mathbf{x}_i^b)\|_2. \quad (2.19)$$

The drawback is CMG is very sensitive to outliers. To fix this problem, robust color morphological gradient (RCMG) is introduced. The difference of RCMG is removing two pixels that are furthest and computing the CMG of the remaining pixels, then repeat this process several time to make the CMG robust. By using the Euclidean distance, RCMG denoted as  $\nabla_{\psi,d}^{RCM}$  can be defined as:

$$\nabla_{\psi,d}^{RCM}(\mathbf{x}_i) = \max_{a,b \in [\mathcal{X}-REM_r]} \|d(\mathbf{x}_i^a, \mathbf{x}_i^b)\|_2, \quad (2.20)$$

where  $RCM_r$  is the set of  $r$  pairs of pixels removed,  $r$  is a parameter determined by the amount of noise present in the image.

After computing the vectorial gradient by either way above, watershed segmentation can be used on the gradient to get the segmentation map.

2). The second way is to compute a multidimension gradient. Consider the  $B$ -band image is composed of a set of one-band images. Each band can be computed to get a gradient by using one of the above vectorial gradient methods, such as RCMG. All  $B$  gradients are combined together into one by a linear or non-linear operator. Watershed segmentation can be performed on the combined gradient.

3) The third way is to combine watershed segmentation maps. For this method, each gradient is computed, watershed algorithm is applied to each gradient to obtain a segmentation map. These  $B$  segmentation maps will be combined into one final segmentation map.

## 2.2.2 Working in the spectral domain

Three image segmentation methods, including edge-based, region-based and feature clustering, are identified in [34]. The first two methods work in the spatial domain to search the discontinuities in the image and measure the similarities between different regions. The cluster-based method which works in the spectral domain aims at partitioning the input image into homogeneous sets of pixels. This method was applied to hyperspectral image segmentation in [43].

## Segmentation of hyperspectral images based on partitional clustering techniques

ISODATA (Iterative Self-Organizing Data Analysis Technique Algorithm) [44] and Expectation Maximization (*EM*) algorithm [45] are two widely used algorithms to perform a single partition of the image data. General steps of clustering algorithms are feature extraction, similarity measure and grouping.

Feature extraction uses some transformation methods to obtain new effective features, which can be used for clustering. For a hyperspectral image, it is composed of hundreds of spectral bands, which contain much useful information. The feature extraction step here is often used to reduce the dimension of the data. Principal component analysis (PCA) [39] and independent component analysis (ICA) [40] are two of the most common used methods for dimensionality reduction.

Similarity measurement is used to qualify the similarity between pixels. To group a set of pixels into a same cluster, we assume pixels within the cluster should have similar spectral information. Several methods to perform similarity measure are introduced in [46]. One of the most common used methods is Euclidean distance between pixel pairs' spectral values.

In the grouping step, a clustering criterion is iterative optimized based on the result of similarity measure.

Denote a hyperspectral image as  $n$  pixel vectors  $\mathbf{X} = \mathbf{x}_i \in \mathbf{R}^B, i = 1, 2, \dots, n$ , here the number of spectral bands is  $B$ . On each optimization iteration  $j$ , a obtained partition is denoted as  $\mathbf{Q}_1^j, \mathbf{Q}_2^j, \dots, \mathbf{Q}_C^j$ , where a cluster  $c$  is defined as  $\mathbf{Q}_c^j = \mathbf{x}_{i,c}^j \in \mathbf{R}^B, i = 1, 2, \dots, m_c^j$ .  $m_c^j$  is the number of pixels in this cluster.

1) Clustering can be performed by ISODATA algorithm. ISODATA was first introduced by Ball and Hall in [44]. It firstly randomly partitions pixel vectors into several clusters as

initialization. Then at each iteration, it reassigns pixel vectors to clusters, which make the clustering criterion reduced. The process is performed until it converges [39]. The most common used criterion is the squared-error criterion which is defined as:

$$e^2(\mathbf{X}, \gamma) = \sum_{c=1}^C \sum_{i=1}^{m_c} \|\mathbf{x}_{i,c} - \boldsymbol{\mu}_c\|^2. \quad (2.21)$$

Here a clustering  $\gamma$  of  $\mathbf{X}$  into  $C$  clusters is computed and  $\boldsymbol{\mu}_c$  is the centroid of the cluster  $c$  [43].

When ISODATA is applied for hyperspectral images, the feature vector is represented by the spectral vector. It has been used in classifying hyperspectral images in [47, 48].

2) Clustering also can be performed by the *EM* algorithm. *EM* algorithm was first introduced in [45] which assumes a statistical model to characterize the data. The aim is to estimate the parameters of the distribution that data is drawn from. Multivariate Gaussian probability distribution is most often used, which the pixel can be modeled as:

$$p(\mathbf{x}) = \sum_{c=1}^C \omega_c \phi_c(\mathbf{x}; \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c). \quad (2.22)$$

Here  $\omega_c \in [0, 1]$  is the mixing weight with the constraint as  $\sum_{c=1}^C \omega_c = 1$ .  $\phi(\boldsymbol{\mu}, \boldsymbol{\Sigma})$  is the multivariate Gaussian density with mean as  $\boldsymbol{\mu}$  and covariance matrix as  $\boldsymbol{\Sigma}$  which is defined as:

$$\phi(\boldsymbol{\mu}, \boldsymbol{\sigma}) = \frac{1}{(2\pi)^{\frac{B}{2}}} \frac{1}{|\boldsymbol{\Sigma}_c|^{\frac{1}{2}}} \exp -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c^{-1} (\mathbf{x} - \boldsymbol{\mu}_c). \quad (2.23)$$

*EM* algorithm is performed to estimate parameters  $\Psi = C, \omega_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c; c = 1, \dots, C$  [49]. The requirement of *EM* algorithm is feature vectors  $\mathbf{X}$  after dimension reduction and an upper bound cluster numbers  $C_{max}$ .

To perform the initialization, set  $C = C_{max}$ , determine the first partition  $\mathbf{Q}_c^0, c = 1, 2, \dots, C$ . Randomly choose  $C$  pixels as cluster centers and assign the rest to the clusters based on the smallest Euclidean distance to the cluster center.

For each optimization iteration  $j$ ,  $\boldsymbol{\mu}_c$  and  $\Sigma_c$  for  $c = 1, 2, \dots, C$  are estimated by component-wise Maximum Likelihood estimates which is defined as:

$$\boldsymbol{\mu}_c^j = \frac{1}{m_c^{j-1}} \sum_{i=1}^{m_c^{j-1}} \mathbf{x}_{i,c}^{j-1} \quad (2.24)$$

$$\Sigma_c^j = \frac{1}{m_c^{j-1}} \sum_{i=1}^{m_c^{j-1}} (\mathbf{x}_{i,c}^{j-1} - \boldsymbol{\mu}_{i,c}^j)(\mathbf{x}_{i,c}^{j-1} - \boldsymbol{\mu}_{i,c}^j)^T \quad (2.25)$$

$$\omega_c^j = \frac{m_c^{j-1}}{n}. \quad (2.26)$$

After the parameters are estimated, assign each pixel in  $\mathbf{X}$  to one cluster according to the maximum a posterior probability criteria which is defined as:

$$\mathbf{x}_i \in \mathbf{Q}_c^j : Pr(c|\mathbf{x}_i) = \max_l Pr(l|\mathbf{x}_i) \quad (2.27)$$

where

$$Pr(c|\mathbf{x}_i) = \frac{\omega_c^j \phi_c(\mathbf{x}_i; \boldsymbol{\mu}_c^j, \Sigma_c^j)}{\sum_{c=1}^C \omega_c^j \phi_c(\mathbf{x}_i; \boldsymbol{\mu}_c^j, \Sigma_c^j)}. \quad (2.28)$$

An elimination step can be performed. When  $m_c^j$  is less than the dimensionality of pixels, eliminate the cluster  $c$ . The pixels assigned to the eliminated cluster will be reassigned to another one in next iteration. Then continue the parameter estimation, cluster assignment and elimination until convergence criterion is achieved.

When the convergence criterion is achieved, each pixel has a label corresponding to the cluster it is assigned. But a drawback is obvious. Because no spatial information is used, pixels from a same cluster may not be connected. Some authors [43] have performed a connected component algorithm [50] to the partition result. Then a good segmentation map can be obtained.

Many other authors have also applied the partition clustering algorithm to hyperspectral images which have been proposed in [51, 52].

### 2.2.3 Combining spatial-based and spectral-based segmentation

For hyperspectral image segmentation, both spectral and spatial information are very important. Many authors have proposed to combine these information together on segmentation process to improve the result [17, 38, 53, 54, 54–66].

#### Segmentation of hyperspectral image based on FH algorithm

The FH algorithm was first introduced by Felzenszwalb and Huttenlocher in [67]. In this approach, RGB or grayscale images with  $n$  pixels are considered as 8-connected graph  $G(V, E)$ , where  $V = v_i^n = 1$  represents each pixel and  $E = (v_i, v_j)$  are the edges between each pixel and its neighbors. A weight is carried by each edge which is defined as a divergence function  $d(v_i, v_j)$ . The divergence function like Euclidean distance usually measures the distance between feature vectors extracted from the original image.

FH algorithm has been extended to hyperspectral images by Thompson in [57]. They use the spectral information to represent the feature vectors, then use Euclidean distance  $d_e$  or spectral angle divergence(SAD)  $d_{sa}$  as divergence function which are defined as below:

$$d_e(v_i, v_j) = \sqrt{\sum_{\lambda} (\rho_{i\lambda} - \rho_{j\lambda})^2} \quad (2.29)$$

$$d_{sa}(v_i, v_j) = \sum_{\lambda} \frac{\rho_{i\lambda} \rho_{j\lambda}}{|\rho_{i\lambda}| |\rho_{j\lambda}|}, \quad (2.30)$$

where  $\lambda$  is the total wavelengths,  $v_i$  and  $v_j$  are two pixels and  $\rho_{i\lambda}$  is the value of  $i$ -th spectrum at wavelength  $\lambda$ .

An agglomerative clustering algorithm is used to partition the hyperspectral image into segments  $S$ . Segmentation starts with each pixel represents a segment by itself, then merges neighboring segments if no boundary between them. The boundary evidence is determined in [57] which is the smallest connecting edge weight is smaller than the largest edge weight in the minimum spanning tree (MST) of either segment[58].

The maximum internal difference inside a segment  $S$  is denoted by  $\text{Int}(S)$ , which is defined as:

$$\begin{aligned} \text{Int}(S) &= \max d(v_i, v_j) \\ &\forall v_i \in S, v_j \in S, (v_i, v_j) \in \text{MST}(S), \end{aligned} \quad (2.31)$$

where  $\text{MST}(S)$  is the minimum spanning tree of segment  $S$ .

For any neighboring segments  $S_a$  and  $S_b$ ,  $E$  represents all connected edges, the smallest connecting edge weight is defined as:

$$\begin{aligned} \text{Dif}(S_a, S_b) &= \min d(v_i, v_j), \\ &\forall v_i \in S_a, v_j \in S_b, (v_i, v_j) \in E. \end{aligned} \quad (2.32)$$

Two neighboring segments will not be merged when the smallest connecting weight

$Dif$  is smaller than the minimum of either maximum internal weight. The threshold is defined below which is biased by a constant  $k$  and the inversely proportional to the segment size[58].

$$Dif(S_a, S_b) < \min(\mathbf{Int}(S_a) + \frac{k}{|S_a|}, \mathbf{Int}(S_b) + \frac{k}{|S_b|}), \quad (2.33)$$

where parameter  $k$  is used to control the smaller returned segment's size.

Due to the instrument noise and atmospheric distortions, there are some pixels forming a few small isolated segments in the segmentation map. To overcome this problem, the final operation called “clean-up” which is merging all segments if the area is smaller than a certain threshold is applied. The segmentation map is then obtained.

### Segmentation of hyperspectral image based on ultrametric contour map

Ultrametric contour map (UCM) is widely used in superpixel estimation for RGB or grayscale images [68]. Brightness, color and texture cues are extracted and used to estimate the probability of the existence of a boundary between each pixel in the image [63, 64].

The UCM algorithm have been extended to hyperspectral images and very high resolution (VHR) remote sensing images by authors in [63, 64]. Principal component analysis (PCA) is used to reduce the dimensionality and represent the hyperspectral image in a compact form, then superpixel estimation is followed.

The first step is feature extraction using oriented Gaussian derivative filters at multiple scales through all wavelength  $\lambda$ :

$$I_{local}(x, \theta) = \sum_{\lambda} \sum_{s(\lambda)} \sum_{i(\lambda)} \omega_{i(\lambda), s(\lambda)} G(x, \theta; \sigma(i(\lambda), s(\lambda))), \quad (2.34)$$

where function  $G$  is the Gaussian filter,  $\theta$  is the oriented angle and  $\omega_{i(\lambda),s(\lambda)}$  is the weights depend on the channels and scales. With this process, local features at different orientations which integrate features from all bands are obtained [63, 64].

The second step is weighted graph construction by connecting pixels within a certain distance by the weighted edge, which was defined as:

$$W(x, y) = \exp[-\alpha \max_{z(x,y)} \max_{\theta} I_{local}(z(x, y), \theta)], \quad (2.35)$$

where  $\alpha$  here is a constant and  $z(x, y)$  represents any pixels lying on the line segments connecting  $x$  and  $y$ .

The third step is eigenvector computation. The top  $K$  eigenvectors  $e_k(x)$  are computed based on the spectral clustering strategy.

The fourth step is gradient computation on the eigenvectors. The process is combining all information from each PCA channel into one, which is defined as:

$$I_{spectral}(x, \theta) = \sum_k \nabla_{\theta}(e_k(x)). \quad (2.36)$$

The fifth step is the linear combination of information in step 1 and step 3.

The last step is performing oriented watershed transform [63]. The image can be segmented into superpixels.

### **Segmentation of hyperspectral image based on the normalized cut algorithm**

Many existing spectral graph algorithms have been used to segment the RGB or grayscale imagery. One of the most commonly used methods is normalized cuts algorithm, which was introduced by Shi and Malik [69]. It uses contour and texture cues to partition the

given graph recursively while minimizing the cost of the cut at the partition boundaries.

Gillis and Bowles have proposed using a joint model to combine the spectral and spatial information based on normalized cuts to perform hyperspectral images segmentation in [65].

The image is represented by a graph  $G = (V, E)$  where vertices are denoted as  $V = (v_1, \dots, v_n)$  and edges are denoted as  $E = e_{i,j}$  which represent the connections between vertices. All graphs are assumed to be undirected and weighted. The weight on each edge is used to measure the similarity of pixels, which is defined as:

$$\omega_{i,j} = e^{-\omega(\rho_i, \rho_j)} \begin{cases} e^{\frac{-d(\rho_i, \rho_j)}{\sigma}}, & d(\rho_i, \rho_j) < r \\ 0, & \text{else} \end{cases} \quad (2.37)$$

where  $\omega(\rho_i, \rho_j)$  is the spectral distance which is measured by the spectral angle while  $d(\rho_i, \rho_j)$  is the spatial distance between pixel  $\rho_i$  and  $\rho_j$ . They are defined as:

$$\omega(\rho_i, \rho_j) = \cos^{-1}\left(\frac{\langle \rho_i, \rho_j \rangle}{\|\rho_i\| \|\rho_j\|}\right) \quad (2.38)$$

$$d(\rho_i, \rho_j) = (i_1 - j_1)^2 + (i_2 - j_2)^2. \quad (2.39)$$

where  $i_1, i_2$  and  $j_1, j_2$  are the coordinates of pixel  $\rho_i$  and  $\rho_j$  in the image.  $r$  and  $\sigma$  are user defined parameters.

The goal of the segmentation is to partition all pixels of the image into two clusters. This algorithm performs a “top-down” partition recursively until it converges. To partition a graph into two subgraphs, any edges that connect nodes in different cluster will be removed. Minimization the weights of the removed edges is used to optimize the partition. Let us

denote the two subgraphs as  $A$  and  $B$ .  $\text{cut}$  is defined as the sum of all weights:

$$\text{cut}(A, B) = \sum_{i \in A, j \in B} \omega_{i,j}. \quad (2.40)$$

Here  $A \cup B = V$  and  $A \cap B = \emptyset$  are assumed.

To avoid the outliers, a normalization cut operation has been proposed by Shi and Malik [69] which is given as:

$$N\text{cut}(A, B) = \frac{\text{cut}(A, B)}{\text{assoc}(A)} + \frac{\text{cut}(A, B)}{\text{assoc}(B)}, \quad (2.41)$$

where  $\text{assoc}(A)$  is defined as:

$$\text{assoc}(A) = \text{cut}(A, V) = \sum_{i \in A, v \in V} \omega_{i,v}. \quad (2.42)$$

The minimization problem can be changed into solving the eigen-decomposition problem which is denoted as:

$$Lv = \lambda Dv, \quad (2.43)$$

where  $v \in \mathbb{R}^n$  is an indicator vector whose entries are either  $-k$  with  $k$  as a constant or 0 and subject to  $v' D \mathbf{1} = 0$  where  $\mathbf{1}$  is the n-dimensional vector with all elements as 1.  $L$  is the Laplacian matrix and  $D$  is the degree matrix. The degree of a node is defined as:

$$\deg(v_i) = \sum_j \omega_{i,j} \quad (2.44)$$

The degree matrix  $D$  is a diagonal matrix with entries as  $d_{i,j} = \deg(v_i)$  when  $i = j$  and all other parts as zero.

The Laplacian matrix is defined as:

$$L = D - A, \quad (2.45)$$

where  $A$  here is the adjacency matrix with entries as  $a_{i,j} = \omega_{i,j}$ .

When we transfer the problem to the generalized eigenvalue problem  $Lv = \lambda Dv$ , the eigenvector corresponding to the smallest nonzero eigenvalue can be used to partition the graph by choosing an appropriate threshold. All pixels in the image are partitioned into two clusters. Recursively partition is performed on the largest one among all segments until some stopping criterion is reached. Finally, the image is segmented.

### Segmentation of hyperspectral image based on k-means algorithm

$K$ -means clustering is a very common used method with the iterative process. Firstly, select  $k$  points as centroids for initialization, calculate the distance between each point and each centroid and assign the point to the closest centroid. Then reassign the centroid by calculating the mean in each cluster until the centroids do not change any more.

Authors in [15] have applied  $K$ -means in the hyperspectral images by combining the spectral distance and spatial distance together. Let us denote a hyperspectral image as  $n$  pixel vectors  $\mathbf{X} = \mathbf{x}_i \in \mathbf{R}^B, i = 1, 2, \dots, n$ , here the number of spectral bands is  $B$ . The spectral distance of two pixels is defined as:

$$d_{spectral} = \sum_{\lambda=1}^B \|\mathbf{x}_i(\lambda) - \mathbf{x}_j(\lambda)\|_2^2. \quad (2.46)$$

If the  $(a_i, b_i)$  and  $(a_j, b_j)$  are the spatial coordinates for  $\mathbf{x}_i$  and  $\mathbf{x}_j$ . The spatial distance of these two pixels is defined as:

$$d_{spatial} = \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2}. \quad (2.47)$$

The combination function of these two distances is defined as:

$$d_{i,j} = d_{spectral} + md_{spatial}, \quad (2.48)$$

where  $m$  is the scaling factor which is used to control the importance of spatial distance.

After the distance function of two pixels is defined, segmentation result is obtained by simply going through the  $K$ -means algorithm. The detail of  $K$ -means algorithm is presented in Algorithm 2.

---

**Algorithm 2**  $K$ -Means Algorithm

---

**Input:** data points, number of clusters( $k$ )

**Output:** centroids, centroid membership

```

1: if centroids are not initialized then
2:   Randomly select  $k$  points as centroids
3: end if
4: for each cluster center do
5:   Calculate the distance from each centroid to each data point
6:   Assign each data point to the closest centroid
7:   Recalculate centroid by calculating the mean of a cluster
8:   if centroids do not change and each data point maintains centroid classification then
9:     end for
10:   else
11:     Continue loop
12:   end if
13: end for
```

---

## 2.3 Cluster Validity

After clustering or segmentation, methods used to quantitatively evaluate the result is a necessary task, and generally, it is known as cluster validity. In this section, we will discuss three well-known cluster indices to evaluate the hard clustering results from different algorithms [70].

### 2.3.1 The Dunn and Dunn-like Indices

Dunn index was introduced in [71]. In the Dunn index, the dissimilarity function of two different clusters  $C_i$  and  $C_j$  is defined as:

$$d(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y}), \quad (2.49)$$

where  $\mathbf{x}$  and  $\mathbf{y}$  are the pixel vectors.

The diameter of the cluster  $C$  is defined as:

$$\text{diam}(C) = \max_{\mathbf{x}, \mathbf{y} \in C} d(\mathbf{x}, \mathbf{y}). \quad (2.50)$$

Here the diameter is used to represent the distance between two farthest data points within the cluster  $C$ , and that is, the dispersion of this cluster.

The Dunn index with  $m$  clusters is defined as:

$$D_m = \min_{i=1, \dots, m} \left\{ \min_{j=i+1, \dots, m} \frac{d(C_i, C_j)}{\max_{k=1, \dots, m} \text{diam}(C_k)} \right\}. \quad (2.51)$$

From the definition, if the returned clusters are well-separated, the Dunn index should be large because the diameter of the cluster should be small and the dissimilarity of the

clusters should be large. Therefore, larger Dunn index value indicates better separated clustering results.

Another thing to notice is the Dunn index does not exhibit any trend with respect to the number of cluster. The reason for this is the maximum operation is taken with respect to the number of cluster and the maximum of  $D_m$  can be used to indicate the best partition results.

When the Dunn index is used to evaluate the clustering results, due to the high dimension of the data, the calculation is very time consuming, Also, the index value is also sensitive to noise, since outliers will significantly increase the diameter of a cluster.

There are also many Dunn-like indices that have been proposed and used to evaluate the clustering results [72, 73]. The difference between them is that they have different definitions of dissimilarity and diameter.

### 2.3.2 Davies-Bouldin Index

Davies-Bouldin index was proposed in [74]. Denote  $R_{ij}$  as the similarity measurement of cluster  $C_i$  and  $C_j$ ,  $s_i$  as the dispersion of the cluster  $C_i$  and  $d_{ij}$  as the dissimilarity of these two clusters.

The similarity  $R_{ij}$  is required to satisfy the following conditions [74]:

- 1)  $R_{ij} \geq 0$ .
- 2)  $R_{ij} = R_{ji}$ .
- 3) If  $s_i = 0$  and  $s_j = 0$ , then  $R_{ij} = 0$ .
- 4) If  $s_j > s_k$  and  $d_{ij} = d_{ik}$ , then  $R_{ij} > R_{ik}$ .
- 5) If  $s_j = s_k$  and  $d_{ij} < d_{ik}$ , then  $R_{ij} > R_{ik}$ .

We can indicate that the similarity is symmetric and nonnegative from these conditions.

The similarity function  $R_{ij}$  is defined as:

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}, \quad (2.52)$$

while in [74], the dissimilarity function is defined as:

$$d_{ij} = \|\mathbf{w}_i - \mathbf{w}_j\|_2^2, \quad (2.53)$$

where  $\mathbf{w}_i$  and  $\mathbf{w}_j$  is the representations of these two clusters, usually, mean vector is used.

The dispersion  $s_i$  is defined as:

$$s_i = \left( \frac{1}{n_i} \sum_{\mathbf{x} \in C_i} \|\mathbf{x} - \mathbf{w}_i\|^2 \right)^{\frac{1}{2}}, \quad (2.54)$$

where  $n_i$  is the number of data points in cluster  $C_i$ .

Given the above equations, the Davies-Bouldin index with respect to the cluster number  $m$  is defined as

$$DB_m = \frac{1}{m} \sum_{i=1}^m R_i, \quad (2.55)$$

where  $R_i$  is defined as:

$$R_i = \max_{j=1, \dots, m, j \neq i} R_{ij}, i = 1, \dots, m. \quad (2.56)$$

Based on these definitions, the Davies-Bouldin index indicates the average similarity between clusters, and the smaller value will indicate the better clustering results.

Evaluating clustering results by Davies-Bouldin index does not depend on any particular clustering algorithms. Actually, it can be used to evaluate regardless to how the

clustering algorithms work and how many clusters they generate [74].

### 2.3.3 Silhouette Index

The Silhouette index was proposed in [75]. Denote  $C_{C_i}$  as the cluster and  $\mathbf{x}_i$  as the data points. For each  $\mathbf{x}_i$  in the cluster  $C_{C_i}$ , the average distance  $a_i$ , which is the average distance between  $x_i$  and the other data points in this cluster, is defined as:

$$a_i = d_{avg}^{ps}(\mathbf{x}_i, C_{C_i} - \mathbf{x}_i), \quad (2.57)$$

where the function  $d_{avg}^{ps}$  is the average distance function between a data point and a cluster.

Denote  $b_i$  as the average distance between  $\mathbf{x}_i$  and its closest neighboring cluster,  $b_i$  is defined as:

$$b_i = \min_{k=1, \dots, m, k \neq C_i} d_{avg}^{ps}(\mathbf{x}_i, C_k), \quad (2.58)$$

Denote  $s_i$  as the silhouette width of a data point  $\mathbf{x}_i$ , and  $s_i$  is defined as:

$$s_i = \frac{b_i - a_i}{\max(b_i - a_i)}. \quad (2.59)$$

It is clear that the range of  $s_i$  is  $[-1, 1]$ . When it is close to 1, we can indicate that  $b_i$  is much bigger than  $a_i$ , it means  $\mathbf{x}_i$  is much closer to the cluster it belongs than its neighboring cluster.  $\mathbf{x}_i$  is well assigned.

With the above equations, the silhouette index of a cluster is defined as:

$$S_j = \frac{1}{n_j} \sum_{i: \mathbf{x}_i \in C_j} s_i, j = 1, \dots, m, \quad (2.60)$$

where  $n_j$  is the number of data points in  $C_j$ .

Based all definitions, the global silhouette index is defined as:

$$S_m = \frac{1}{m} \sum_{j=1}^m S_j. \quad (2.61)$$

Based on the definitions, we can indicate that the larger silhouette index value, the better the clustering result. The global silhouette shows no trends with respect to the number of clusters.

# Chapter 3

## Proposed Methods

In this chapter, we will introduce our proposed method for hyperspectral image superpixel segmentation. Each step of process will be discussed in detail.

### 3.1 Methodology

In this thesis, we proposed a map-guided hyperspectral unmixing-based superpixel segmentation algorithm. The method starts with an initialized superpixel segmentation result obtained using a modified SLIC algorithm, then using the initial superpixels, hyperspectral unmixing is performed by semi-supervised Partial Membership Latent Dirichlet Allocation (PM-LDA). Using the unmixing results, a  $K$ -means based superpixel segmentation method is proposed to generate the final superpixel segmentation. The process of the proposed method is summarized in Fig 3.1. The red parts are the initial superpixel segmentation, the green parts are unmixing process and the blue parts are the final superpixel generation process. The details will be discussed in the following sections.

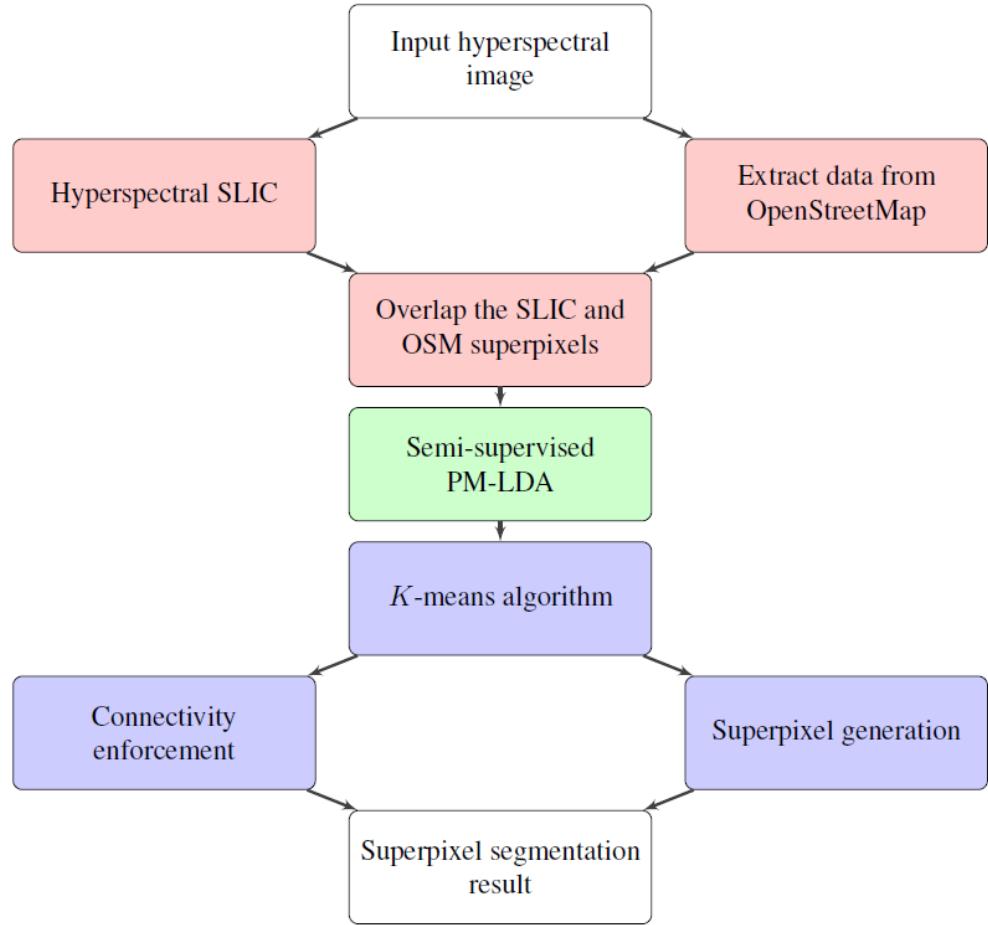


Figure 3.1: The Flowchart of Proposed Method

### 3.2 Initial Superpixel Segmentation Based On Map-Guided Hyperspectral SLIC

To perform the initial superpixel segmentation, a well known superpixel algorithm Simple Linear Iterative Clustering (SLIC) [76] has been extended to work on hyperspectral images. The SLIC algorithm has been widely used in RGB or grayscale images, and it generates superpixels based on the color similarity and proximity in the image plane. SLIC performs a local clustering of pixels in 5-D space with features as  $L,a,b$  value and the coordinates

of the pixel. A *Lab* color space is a color-opponent space with dimensions  $L$  for lightness and  $a$  and  $b$  for the color-opponent dimensions, based on nonlinearly compressed (e.g. CIE XYZ) coordinates [77].

Some authors have proposed to use SLIC on hyperspectral images by performing dimensionality reduction of the original data in [78]. In this thesis, we also extend SLIC to hyperspectral images for superpixel segmentation. However, the difference in our method is that we use all spectral information without dimensionality reduction and combine them with the spatial information as features. The *Lab* features are replaced by the all bands information. Let us denote a hyperspectral image as  $N$  pixel vectors  $\mathbf{X} = \mathbf{x}_i \in \mathbf{R}^B, i = 1, 2, \dots, N$ , here the number of spectral bands is  $B$ . One input for the algorithm is the desired number of superpixels  $K$ . SLIC enforces that the expected superpixel's size and their spatial extent are approximately equal, so the approximately size of each superpixel is  $N/K$  and at every grid interval  $S$  there can be a cluster center.

The  $K$  cluster center is defined as  $\mathbf{c}_k = [\mathbf{x}_k(1), \mathbf{x}_k(2), \dots, \mathbf{x}_k(B), a_k, b_k]$ , where  $(a_k, b_k)$  is the spatial location of pixel  $k$ . Because the approximate area of a superpixel is  $S^2$ , the search area for the pixels nearest to each cluster center can be safely assumed as  $2S \times 2S$ .

Our hyperspectral SLIC approach starts with sampling  $K$  cluster centers with regular space and moving them to find the lowest gradient position. Image gradients are defined as:

$$G(a, b) = \|\mathbf{x}(a + 1, b) - \mathbf{x}(a - 1, b + 1)\|^2 + \|\mathbf{x}(a, b + 1) - \mathbf{x}(a, b - 1)\|_2^2, \quad (3.1)$$

where  $\mathbf{x}(a, b)$  is the spectral vector corresponding to the pixel at position  $(a, b)$ .

Each pixel is assigned to a nearest cluster center. After all pixels are assigned, we use the average feature vector of all pixels within the cluster to represent the new cluster

centers. Then this process will be iteratively repeated until the convergence.

A post-process based on OpenStreetMap (OSM) [79] is performed on the segmentation map returned by hyperspectral SLIC to help us get the initial segmentation result.

OpenStreetMap is a collaborative project to create a free editable map of the world. Here we use Pavia University area as an example to show how OSM looks like in Fig 3.2.

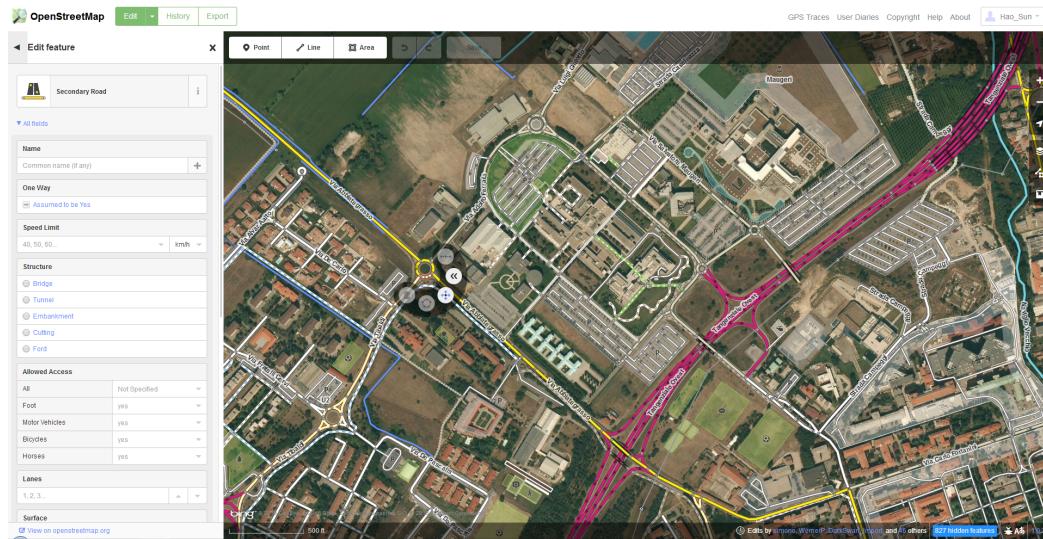


Figure 3.2: OpenStreetMap on Pavia University

From the database of OSM, we can extract a “.osm” data. After parsing the “.osm” data, we can get the information for that area like node, way, relation and tag. Here the node represents a specific point on the earth’s surface defined by its latitude and longitude and each node comprised of at least a pair of coordinates. The way here refers to an ordered list of nodes that define polylines. The ways are used to represent linear features such as buildings and roads. The relation here represents a multi-purpose data structure that documents a relationship between two or more data elements (nodes and ways). The tag here describes the meaning of the particular region. Fig 3.3 shows the nodes and ways from OSM.

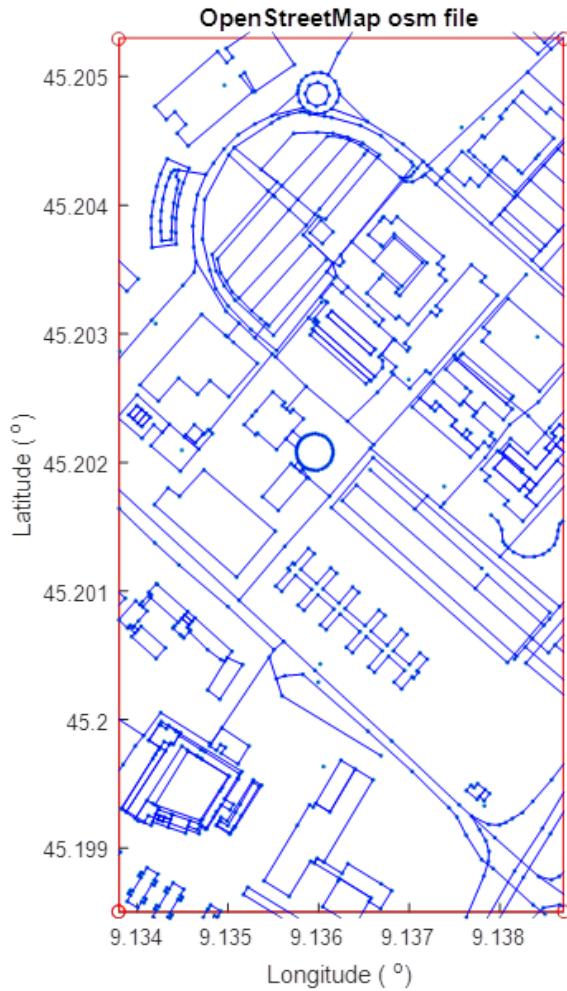


Figure 3.3: Nodes and Ways from OpenStreetMap on Pavia University

With information from all these nodes and ways, objects in an area can be represented as a polygon or a line by connecting each node corresponding to that object. Since all latitude and longitude coordinates are given, affine transformation can be performed to match the latitude and longitude of the nodes with the coordinate location in the image. A few points in Fig 3.3 are manually chosen, also points with the same location in the RGB figure of Pavia are manually chosen. With these two groups of points which contain the geographical

locations and the coordinates' information in the image, a transformation matrix can be constructed. Then all points with geographical information can be transformed into pixels with coordinates in the image. Buildings represented by polygons and roads represented by lines are constructed by the transformed coordinates. Then the image is overlapped by these polygons and lines. Fig 3.4 shows the result of this process.

Each polygon from OSM is treated as a superpixel which represents a specific region with a label. To guide the initial superpixel segmentation process, superpixels from hyperspectral SLIC which overlap with polygons from OSM are merged as one superpixel. In this thesis, we use the polygons which are labeled as buildings.

The map-guided hyperspectral SLIC is summarized in Algorithm 3

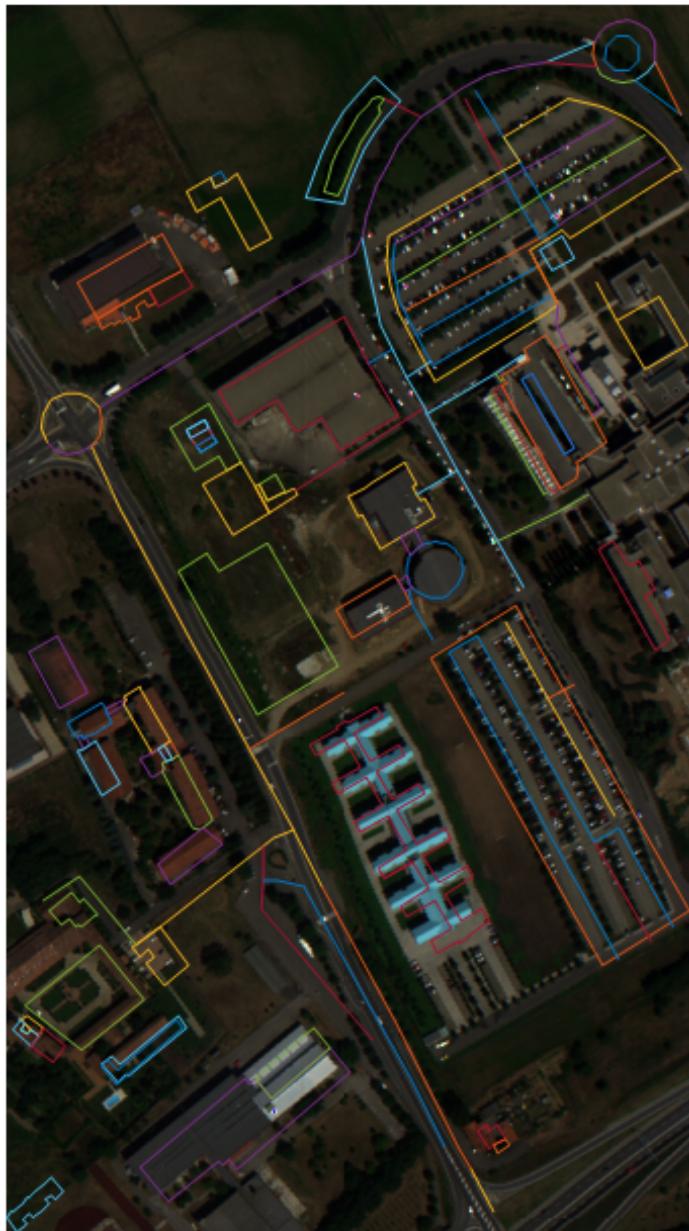


Figure 3.4: Overlap OpenStreetMap information with the image of Pavia University

---

**Algorithm 3** Hyperspectral SLIC superpixel segmentation

---

**Input:** original hyperspectral data, desired number of clusters(superpixels)  $K$  , scaling factor  $m$

**Output:** superpixels

- 1: Initialize cluster centers  $\mathbf{c}_k$  by sampling pixels at regular grid steps  $S$ .
- 2: Perturb cluster centers in an  $n \times n$  neighborhood, to the lowest gradient position.
- 3: **repeat**
- 4:   **for** each cluster center  $\mathbf{c}_k$  **do**
- 5:     Calculate the spectral distance of pixel  $\mathbf{x}_i$  and  $\mathbf{c}_k$  over all bands  $\lambda$ :  
$$d_{spectral} = \sum_{\lambda=1}^B \|\mathbf{x}_i(\lambda) - \mathbf{c}_k(\lambda)\|_2^2$$
- 6:     Calculate the spatial distance of pixel  $\mathbf{x}_i$  and  $\mathbf{c}_k$ :  
$$d_{spatial} = \sqrt{(a_{\mathbf{x}_i} - a_{\mathbf{c}_k})^2 + (b_{\mathbf{x}_i} - b_{\mathbf{c}_k})^2}$$
 ( $a, b$  are the coordinates)
- 7:     Assign the best matching pixels from a  $2S \times 2S$  square neighborhood around the cluster center according to the spectral and spatial combination distance function:  
$$d_{\mathbf{x}_i, \mathbf{c}_k} = d_{spectral} + \frac{m}{S} d_{spatial}$$
- 8:   **end for**
- 9:   Compute new cluster centers by taking mean operation
- 10:   **Until** stopping criterion reaches.
- 11:   Extract and parse data from OpenStreetMap
- 12:   Use affine transformation to generate coordinates from latitude and longitude and Construct polygons and lines based these coordinates
- 13:   **for** each polygon **do**
- 14:     **if** superpixels returned from hyperspectral SLIC have overlap with this polygon **then**
- 15:       Merge these superpixels into one
- 16:     **end if**
- 17:   **end for**

---

### 3.3 Hyperspectral Unmixing Based On Semi-Supervised PM-LDA

In this thesis, a hyperspectral image is first over-segmented into spatially-contiguous superpixels. Then hyperspectral unmixing process is followed and performed by semi-supervised PM-LDA [33].

Each superpixel in the hyperspectral image is assumed to be a document. Each end-member is treated as a topic. We assume that the topic distributions are Gaussian, then

the  $K$  topic distributions with parameters  $\beta_k = \{\mu_k, \Sigma_k\}$  correspond to the  $K$  Gaussian endmember distributions. The partial membership vector for data point  $n$  in document  $d$ ,  $\mathbf{z}_n^d$ , is the proportion vector associated with the  $n^{th}$  data point in the  $d^{th}$  superpixel. The topic proportion vectors for a document,  $\boldsymbol{\pi}^d$ , correspond to the average proportion vector for a superpixel with the mixing level  $s^d$  corresponding to how much each proportion vector in the document is likely to vary from the average proportion vector. Thus, an entire hyperspectral scene is modeled as a corpus in PM-LDA.

Given the initial superpixel segmentation, supervision is imposed by assigning an end-member label  $\tau_j$  to each superpixel  $j$ . To be specific, we assume there is a probability (including zero) that superpixel  $j$  contains endmember  $i$  if  $\tau_{ij} = 1$ , while superpixel  $j$  does not contain endmember  $i$  if  $\tau_{ij} = 0$  (with probability 1). For example, in University of Pavia dataset, we set 6 as the number of endmembers and name them as  $e_1, e_2, e_3, e_4, e_5, e_6$ . Endmembers are selected to be supervised if some superpixels contain these endmembers. For example, the endmember  $e_1$  is a painted metal sheets material, it is easy to label the superpixels containing  $e_1$  as  $\tau_{1j} = 1$  and all other superpixels as  $\tau_{1j} = 0$ . Meanwhile, end-members are treated as unsupervised if the pixels containing them are widely distributed over a large number of superpixels. Consider the extreme case, if the endmember prior matrix  $\tau_{ij} = \mathbf{1}$  for all superpixels, the semi-supervised PM-LDA degraded to standard PM-LDA.

The details of semi-supervised PM-LDA are summarized in Algorithm 1 which is mentioned in Chapter 2.

### 3.4 Superpixel Segmentation Based On Proportion Maps

After the proportion maps are obtained using semi-supervised PM-LDA, the proportion vectors are used as the features to perform the segmentation. The  $K$ -means algorithm is a well-known clustering algorithm which was first proposed in [80]. It aims to partition several observations into  $K$  clusters in which each observation is assigned to a cluster with the nearest distance. Thus,  $K$ -means can successfully obtain the useful information from the observation and perform the segmentation on hyperspectral images. The details of  $K$ -means algorithm are explained in Algorithm 2. We apply  $K$ -means clustering to the proportion vectors obtained using semi-supervised PM-LDA.

To generate the superpixels, we propose to use connected components algorithm [81] to relabel each component as a superpixel. Here a connected component represents as a set of pixels in which each pixels is connected to all other pixels. The connected components algorithm aims to find all connected components in the image and assigns a unique label to all points in the same component. In our case, for each cluster returned by  $K$ -means algorithm, connected components algorithm is applied to relabel each connected component and produce superpixels.

However, a problem is that the results returned are not continuous. A few pixels in the vicinity of a large superpixel have the same label but are not connected to it. To enforce the connectivity, we propose to set a threshold to find the small disjoint segments and assign them with the labels of the biggest neighboring superpixel. The details of this process are summarized in Algorithm 4.

---

**Algorithm 4** Superpixel Generation and Connectivity Enforcement Algorithm

---

**Input:** clustering results  $C_1, C_2, \dots, C_n$ , threshold  $\epsilon$

**Output:** superpixels

```
1: for each cluster  $C_i$  do
2:   Assign 1 to the pixels which are in cluster  $C_i$ , assign 0 to others
3:   Use connected component algorithms to generate components
4:   Assign unique labels to each components to make them superpixels
5:   for each superpixel do
6:     if the size of superpixel is smaller than  $\epsilon$  then
7:       assign it with the label of the biggest neighboring superpixel
8:     end if
9:   end for
10: end for
```

---

### 3.5 Evaluation

To evaluate the superpixel segmentation result, cluster validity is used to quantitatively show the performance of the segmentation. Three widely used cluster validity indices are reviewed in Chapter 2. We use this to quantitatively evaluate the results. However, these indices are not a perfect fit for our situation. In order to consider the superpixel case, a few changes are made based on the original Dunn index and Davies-Bouldin (DB) index. With these changes, only the dissimilarity of the neighboring superpixels is emphasized and the calculations are less time consuming.

Each superpixel obtained from our method has a major endmember in that area, while the neighboring superpixel is represented by another endmember. However, one endmember can exist in several superpixels which are not spatially connected, so the dissimilarities of these superpixel should not be enforced.

In the original Dunn index, when we calculate the dissimilarity from different clusters, we consider each pair of clusters. Assuming each superpixel as a cluster, in the modified Dunn index, the superpixel and its neighboring superpixels are only considered and

calculated. The modified Dunn index with  $m$  clusters is defined in equation 3.2.

$$Dunn_m^{modified} = \min_{i=1,\dots,m} \left\{ \min_{j=neighboring(i)} \frac{d(C_i, C_j)}{\max_{k=1,\dots,m} diam(C_k)} \right\}. \quad (3.2)$$

Here the dissimilarity  $d(C_i, C_j)$  of two different clusters  $C_i$  and  $C_j$  and the diameter  $diam(C)$  of the cluster  $C$  are defined in equation 2.49 and equation 2.50.

The details of how to compute modified Dunn index are presented in Algorithm 5.

---

**Algorithm 5** Modified Dunn index

---

**Input:** superpixels (clusters)  $C_1, C_2, \dots, C_m$

**Output:** modified Dunn index

- 1: **for** each superpixel  $C_i$  **do**
  - 2:     Calculate diameter  $diam(C_i)$  of the superpixel  $C_i$ :  

$$diam(C_i) = \max_{\mathbf{x}, \mathbf{y} \in C_i} d(\mathbf{x}, \mathbf{y})$$
  - 3:     Find all neighboring superpixels  $C_j$  of  $C_i$ ,  $j = 1, \dots, n$
  - 4:     **for** each superpixel  $C_j$  **do**
  - 5:         Calculate dissimilarity  $d(C_i, C_j)$  of two different superpixels  $C_i$  and  $C_j$ :  

$$d(C_i, C_j) = \min_{\mathbf{x} \in C_i, \mathbf{y} \in C_j} d(\mathbf{x}, \mathbf{y})$$
  - 6:     **end for**
  - 7:     Calculate modified Dunn index:  

$$Dunn_m^{modified} = \min_{i=1,\dots,m} \left\{ \min_{j=1,\dots,n} \frac{d(C_i, C_j)}{\max_{k=1,\dots,m} diam(C_k)} \right\}$$
  - 8: **end for**
- 

Since the change of our modified Dunn index is considering the neighboring superpixels for each superpixel when the dissimilarity is calculated, the result can be affected by the superpixel numbers. If the image is oversegmented, the neighboring superpixels will be similar to the current one, so the dissimilarity will be small and the modified Dunn index will be small too. It means the partition result is not good.

As long as the image is not oversegmented, our modified Dunn index can evaluate the superpixel segmentation result. The larger modified Dunn index indicates that the dissimilarity between superpixels is big and the diameter within each superpixel is small, so the

result with a larger modified Dunn index has a well-separated partition.

In the original Davies-Bouldin (DB) index, when the similarity of two clusters is calculated, each pair is considered. In the modified DB index, we calculate the similarity between the cluster and its neighboring clusters.

The details of how to compute modified DB index are presented in Algorithm 6.

---

**Algorithm 6** Modified Davies-Bouldin index

---

**Input:** superpixels (clusters)  $C_1, C_2, \dots, C_m$

**Output:** modified DB index

```

1: for each superpixel  $C_i$  do
2:   Calculate dispersion  $s_i$  of the superpixel  $C_i$ :

$$s_i = (\frac{1}{n_i} \sum_{x \in C_i} \|x - w_i\|^2)^{\frac{1}{2}}$$
,  $w_i$  is the mean vector of superpixel  $C_i$ ,  $n_i$  is the
   number of data points in cluster  $C_i$ 
3:   Find all neighboring superpixels  $C_j$  of  $C_i$ ,  $j = 1, \dots, n$ 
4:   for each superpixel  $C_j$  do
5:     Calculate dispersion  $s_j$  of the superpixel  $C_j$ 
6:     Calculate dissimilarity  $d_{ij}$  of two superpixels  $C_i$  and  $C_j$ :

$$d_{ij} = \|w_i - w_j\|_2^2$$
,  $w_j$  is the mean vector of superpixel  $C_j$ 
7:     Calculate similarity  $R_{ij}$  of two superpixels  $C_i$  and  $C_j$ :

$$R_{ij} = \frac{s_i + s_j}{d_{ij}}$$

8:   end for
9:   Calculate similarity  $R_i$  of superpixel  $C_i$ :

$$R_i = \max_{j=1, \dots, m, j \neq i} R_{ij}, i = 1, \dots, m$$

10:  end for
11:  Calculate modified DB index:

$$DB_m^{modified} = \frac{1}{m} \sum_{i=1}^m R_i$$


```

---

Just as the modified Dunn index, the modified Davies-Bouldin index is sensitive to the oversegmented partition. It will be very large if the image is oversegmented. Otherwise, it can be used to evaluate the segmentation results. The smaller modified Davies-Bouldin index indicates a better segmentation result, since the similarity of each superpixel is small and dissimilarity between superpixels is big.

# **Chapter 4**

## **Experiment**

In this chapter, we will perform our proposed superpixel segmentation method on two hyperspectral data sets. The superpixel results and the evaluation results will be included and discussed.

### **4.1 HySens Pavia University Data**

The experiment is first performed on the Pavia University data[3]. This image contains several materials such as asphalt, painted metal sheets, vegetation, red roof, bare soil and shadow. The superpixels are generated based on these materials.

#### **4.1.1 Data Set Description**

The Pavia University data set was collected by the Reflective Optics System Imaging Spectrometer (ROSIS) at University of Pavia, Italy in July 2002. Deutschen Zentrum operated

the flight for Luftund Raumfahrt (DLR, the German Aerospace Agency) in the framework of the HySens project, managed and sponsored by the European Union [3]. The ROSIS generates 115 bands cover 430 to 860 nm with 4nm bandwidth. The image contains  $340 \times 610$  pixels and consists of 103 bands with 12 noisiest bands removed. The spatial resolution is 1.3 m [82]. Fig 4.1 shows the RGB image of Pavia University data.

### 4.1.2 Superpixel Segmentation Results

First, an initial map-guided hyperspectral SLIC is performed on the Pavia data. The parameter setting for hyperspectral SLIC is the desired superpixel numbers  $K = 500$  and the scaling factor  $m = 20$ . Fig 4.2 shows the result returned from hyperspectral SLIC, the extracted data from OSM and the final segmentation result.

After the initial superpixels are obtained, the unmixing process is performed by semi-supervised PM-LDA. For Pavia data, the number of endmembers is set as 6. As a pre-processing step, all the pixel signatures in the data set are normalized to have unit length as the input for all methods. Superpixels with endmembers of blue roof building and red roof building are selected to be supervised in this study. The parameters setting which we choose by manual trial and error for semi-supervised PM-LDA:  $K = 6$ ,  $\lambda = 1$ ,  $\alpha = 0.3$ ,  $\epsilon = 5\%$  and  $T = 200$ .

The unmixing results are presented in Fig 4.3.

After the proportion maps are obtained,  $K$ -means algorithm is used to perform the clustering. The “clean-up” process based on connected component is followed to construct the superpixels and enforce the connectivity. The parameter  $k$  as the number of clusters sets to be 6 as same as the number of endmembers in the unmixing process.

We compare our result with 5 other algorithms, which are SLIC, SLIC combines with



Figure 4.1: RGB Image of Pavia University

OpenStreetMap (SLIC+OSM), hyperspectral normalized cuts (HNC) [65], hyperspectral ultra contour map (HUCM)[64] and PM-LDA based superpixel segmentation.

SLIC and SLIC combines with OpenStreetMap (SLIC+OSM) are the initial superpixel segmentation methods in the first step of our proposed method. Comparing these two will help us to ensure our process is effective. The parameter settings for all methods are selected as follows to yield the best performance from trial and error: for SLIC and SLIC+OSM, the desired number of superpixels as  $K = 500$  and the weighting parameter between spectral and spatial difference as  $m = 20$ .

Normalized Cuts (NC) and Ultra Contour Map (UCM) are two widely used superpixel segmentation algorithms for regular color images. They are modified to extend to working on hyperspectral images in [64, 65]. The parameters setting for these are selected from the proposed paper [64, 65] as follows to yield the best performance: for HNC, the ratio of largest segment over all pixels as  $r = 0.01$  and the scale for spatial and spectral distance as  $\sigma = 50$ ; for HUCM, the scale for the boundaries  $k = 0.3$ .

PM-LDA based superpixel segmentation shares almost same process as our proposed method. The only difference between these two is the map guidance from social media. By comparing these two algorithms, we can ensure whether the map guidance matters. The parameters setting are the same as our proposed method:  $K = 6$ ,  $\lambda = 1$ ,  $\alpha = 0.3$ ,  $\epsilon = 5\%$  and  $T = 200$ .

The superpixel segmentation results for the comparison algorithms are presented in Fig 4.4. From Fig 4.4, we can see that the result returned by SLIC is oversegmented, and the result returned by SLIC+OSM is also oversegmented but it can draw most building area as a superpixel. While for result obtained from HNC and HUCM, the superpixels are bigger than SLIC but it somehow clusters different materials into a superpixel incorrectly.

For result obtained from PM-LDA based superpixel segmentation, many superpixels are included a lot of noises. However, the result obtained from our proposed algorithm can segment the image into several superpixels based on the materials in that area.

We also zoom in results to compare the details of the superpixels. Using painted metal sheets as an example, the images are presented in Fig 4.5. From Fig 4.5, in the SLIC result, it partitions the painted metal sheets part into several superpixels and it also excludes some areas; in the SLIC+OSM result, it includes the whole area into a superpixel but it also contains some vegetation and asphalt; in the HNC and HUCM results, both results cannot get the pure painted metal sheet area; in PM-LDA result, it can get some vegetation and shadow superpixels but it includes some asphalt into the painted metal sheet superpixel. While in the result from our proposed method, it can partition the vegetation, shadow and painted metal sheets correctly.

Another example is presented in Fig 4.6. From Fig 4.6, in the SLIC and SLIC+OSM result, they partition the grey part into several superpixels and they also exclude some areas; in the HNC result, it segments the whole grey building into a single superpixel but also includes some shadows; in PM-LDA result, it can include most part of grey building but also includes noise. While in the result from our proposed method, it can partition the shadow and grey building correctly.

To quantitatively evaluate the superpixel segmentation results, we use the Dunn, DB and Silhouette indices and the modified Dunn and DB indices to measure the performance.

The results of the original Dunn, DB and Silhouette indices are presented in Table 4.1. From Table 4.1, we can notice that validity indices returned from our method have better performance than the others. For Dunn index, our proposed method achieves the maximum value. This means our segmentation result has a smaller diameter within a superpixel and

a bigger dissimilarity between superpixels based on Equation 2.51. For Davies-Bouldin index, the minimum DB index indicates the smaller similarity and larger dissimilarity based on the definition. For Silhouette index, the maximum value indicates that the average distance from a pixel to the superpixel it belongs are much smaller than the distance to its neighboring superpixel. All these validity indices represent the same thing as the superpixel segmentation results in Fig 4.4. Superpixels obtained from our proposed method are more uniform and distinguished than superpixels from other algorithms.

	Dunn	Davies-Bouldin	Silhouette
SLIC	$0.0334 \pm 0.0058$	$16.5688 \pm 0.6134$	$-0.8355 \pm 0.0288$
SLIC+OSM	$0.0398 \pm 0.0047$	$16.1190 \pm 0.7945$	$-0.7950 \pm 0.0304$
HNC	$0.0486 \pm 0.0152$	$11.4451 \pm 0.6579$	$-0.7244 \pm 0.0478$
HUCM	$0.0377 \pm 0.0134$	$19.2957 \pm 0.9425$	$-0.7502 \pm 0.0312$
PM-LDA	$0.0607 \pm 0.0143$	$10.9946 \pm 0.8253$	$-0.7059 \pm 0.0484$
Proposed method	<b><math>0.0950 \pm 0.0134</math></b>	<b><math>8.7118 \pm 0.7336</math></b>	<b><math>-0.5606 \pm 0.0502</math></b>

Table 4.1: Dunn, Davies-Bouldin, Silhouette indices and standard deviation on Pavia University

Another thing we need to mention is when we incorporate information from crowd-sourced maps, the validity indices improve. It can be verified by comparing the results of SLIC to SLIC+OSM and PM-LDA to our proposed algorithm. So map-guided superpixel segmentation algorithm often improves the performance due to the additional supervision. Since there are often some shift when we transform the geographical information to coordinates in the image, map-guided algorithms do not perform well in every situation, such as when it was incorporated into the normalized cuts algorithm.

We also modify the original Dunn and DB indices to be more suitable to the superpixel results by considering the neighboring cluster during the calculation. Table 4.2 shows the results of modified validity indices. From Table 4.2, we can see that our proposed algorithm also has the best performance when just considering the neighboring superpixels. In the

results of modified Dunn index, our proposed method can achieve the maximum value which indicates that the smaller diameter within the superpixel and the larger dissimilarity with the neighboring superpixels. In modified Davies-Bouldin index, the value returned by our proposed method is the smallest.

	Modified Dunn	Modified Davies-Bouldin
SLIC	$0.0334 \pm 0.0042$	$5621.6830 \pm 38.3245$
SLIC+OSM	$0.0388 \pm 0.0035$	$4390.3406 \pm 38.4685$
HNC	$0.0486 \pm 0.0034$	$879.7921 \pm 58.3872$
HUCM	$0.0377 \pm 0.0031$	$863.0718 \pm 43.4423$
PM-LDA	$0.1180 \pm 0.0245$	$732.0209 \pm 52.3542$
Proposed method	<b><math>0.3489 \pm 0.0672</math></b>	<b><math>250.0514 \pm 42.8685</math></b>

Table 4.2: Modified Dunn, Davies-Bouldin indices on Pavia University

The most influential parameters for our proposed method are the desired number of end-members for semi-supervised PM-LDA, and the desired number of clusters for  $K$ -means algorithm. If an inappropriate number is chosen, the results of superpixel segmentation degrade. To investigate this, we vary the parameter  $k$  from 3 to 20, and report some representative results in Fig 4.7.

It is clear that when the parameter  $k$  changes, the superpixel segmentation result change significantly from Fig 4.7. When  $k = 3$ , the results can identify most asphalt and building superpixels, but it misclassifies many other materials like vegetation, bare soil, shadow into same superpixels. The reason for this is the clusters returned from  $K$ -means algorithm are smaller than the existing materials in the image. Also, when we set  $k$  as 5, the result is fairly good, but it cannot identify some superpixels with material as shadow and cannot distinguish some small regions surrounded by some larger regions like a vegetation region surrounding by bare soil. Meanwhile, when we tune  $k$  to be big, like 10 or 20, it obviously generates oversegmentation by separating smaller one superpixel into several ones. In our

experience, the trick to choosing the suitable number for parameter  $k$  is set to it around the number of endmembers. For Pavia University data,  $k = 7$  is recommended by manual trial and error.

In order to evaluate our suggestion for choosing parameter  $k$ , we use cluster validity indices to measure the performance with different parameter settings. The results of Dunn, DB and Silhouette indices are presented in Table 4.3. From Table 4.3, Dunn and Silhouette indices achieve the maximum value and Davies-Bouldin index achieves minimum value when  $k = 7$ . This indicates that when  $k = 7$  the proposed method has the best performance in this experiment since the results have smaller similarity within each superpixel and larger dissimilarity between superpixels. The number of endmembers in Pavia data is 6, so the result with  $k = 5$  also have good validity values.

	Dunn	Davies-Bouldin	Silhouette
$k = 3$	$0.0523 \pm 0.0125$	$12.6930 \pm 0.8315$	$-0.8109 \pm 0.0418$
$k = 5$	$0.0812 \pm 0.0152$	$9.4327 \pm 0.8122$	$-0.6212 \pm 0.0485$
$k = 7$	<b><math>0.0950 \pm 0.0134</math></b>	<b><math>8.7118 \pm 0.7336</math></b>	<b><math>-0.5606 \pm 0.0502</math></b>
$k = 10$	$0.0597 \pm 0.0141$	$10.2736 \pm 0.9125$	$-0.7352 \pm 0.0424$
$k = 20$	$0.0375 \pm 0.0112$	$15.3341 \pm 0.9678$	$-0.9095 \pm 0.0611$

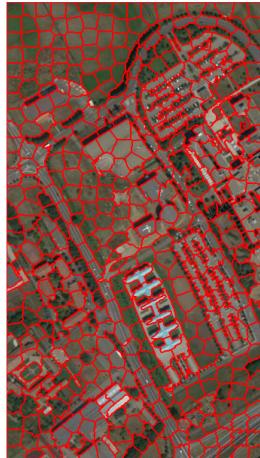
Table 4.3: Dunn, Davies-Bouldin, Silhouette indices and standard deviation on Pavia University when change parameter  $k$

We also evaluate the results with different parameter  $k$  by our proposed modified Dunn and Davies-Bouldin indices. The validity values and standard deviations are presented in Table 4.4. From Table 4.4, when  $k = 7$ , modified Dunn index has largest value and modified Davies-Bouldin index has smallest value. This indicates that the difference between the neighboring superpixels is big and the similarity within each superpixel is small based on the definition of the modified indices. The validity values of result with  $k = 5$  are close with the values with  $k = 7$ . This indicates  $k = 7$  is a proper underlying number of clusters

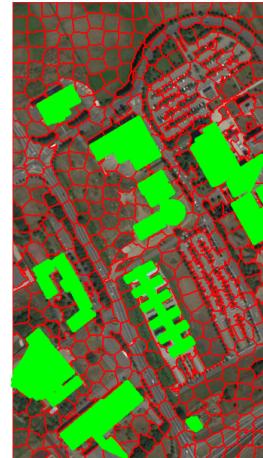
in the Pavia data which proves that the best choice of parameter  $k$  is around the number of endmembers in the data.

	Modified Dunn	Modified Davies-Bouldin
$k = 3$	$0.0398 \pm 0.0037$	$1409.0450 \pm 37.4435$
$k = 5$	$0.3421 \pm 0.0055$	$273.7531 \pm 40.8521$
$k = 7$	<b><math>0.3489 \pm 0.0672</math></b>	<b><math>250.0514 \pm 42.8685</math></b>
$k = 10$	$0.0760 \pm 0.0014$	$756.8493 \pm 48.5622$
$k = 20$	$0.00398 \pm 0.0011$	$2126.8493 \pm 43.4159$

Table 4.4: Modified Dunn, Davies-Bouldin indices on Pavia University and standard deviation on Pavia University when change parameter  $k$



(a) Superpixels from SLIC

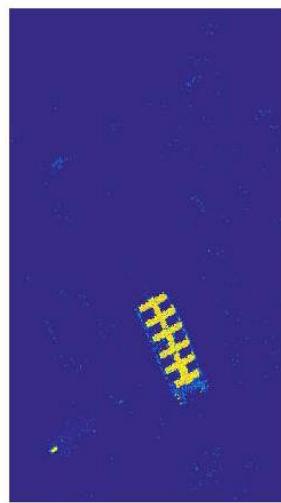


(b) Buildings extracted from OpenStreetMap

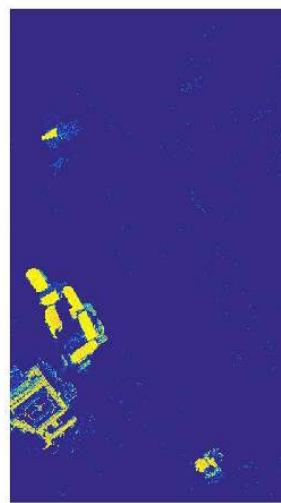


(c) Superpixels by map-guided hy-perspectral SLIC

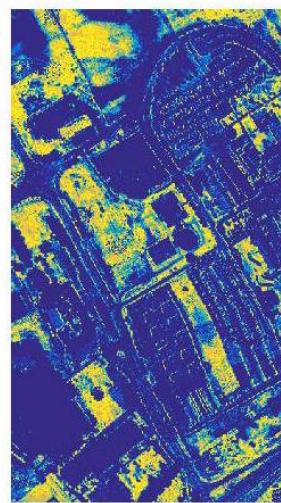
Figure 4.2: Superpixels on Pavia University



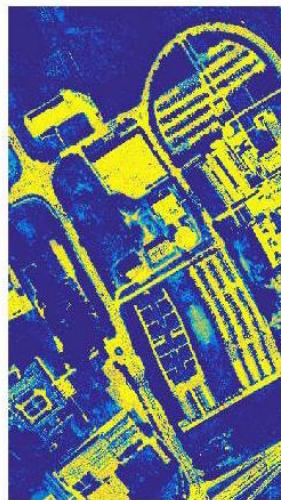
(a) Painted metal sheets



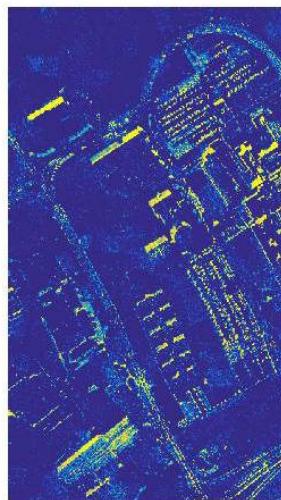
(b) Red roof



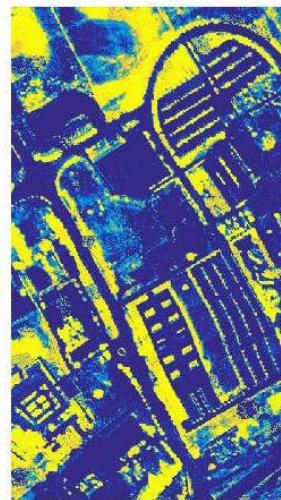
(c) Bare soil



(d) Asphalt



(e) Shadow



(f) Vegetation

Figure 4.3: Estimated proportion maps using semi-supervised PM-LDA on Pavia University

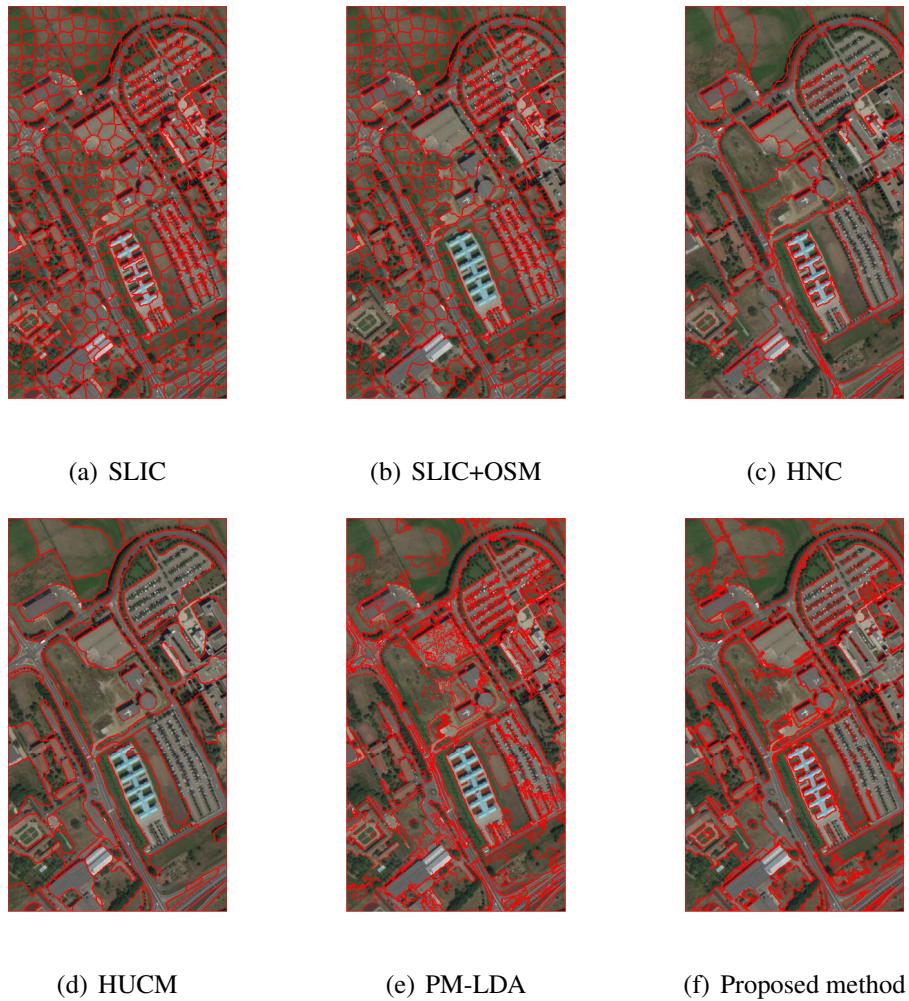


Figure 4.4: Superpixel Segmentation results on Pavia University

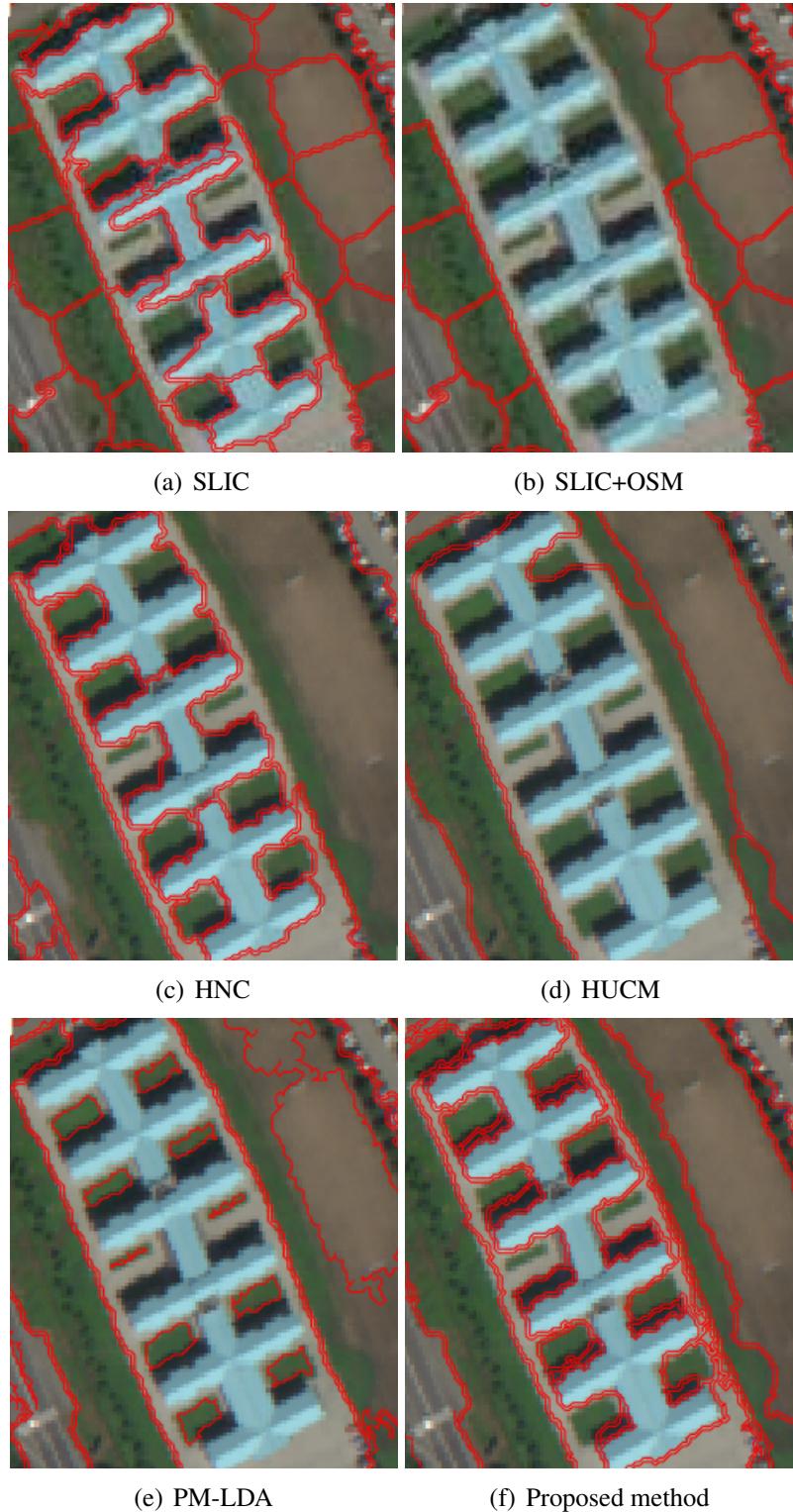


Figure 4.5: Zoom in to painted metal sheets area on superpixel segmentation results



(a) SLIC



(b) SLIC+OSM



(c) HNC



(d) HUCM



(e) PM-LDA



(f) Proposed method

Figure 4.6: Zoom in to grey building area on superpixel segmentation results

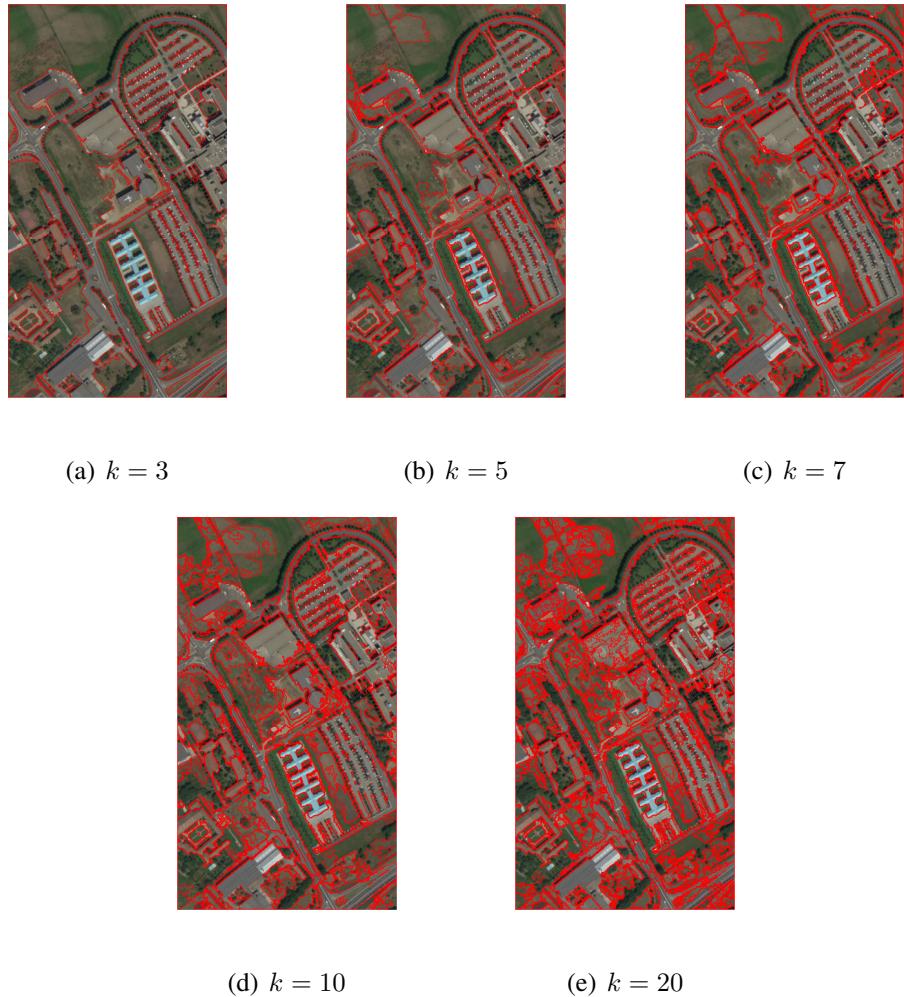


Figure 4.7: Superpixel segmentation results on Pavia University when change parameter  $k$

## 4.2 MUUFL Gulfport Hyperspectral Data

The second experiment is performed on MUUFL Gulfport hyperspectral data[83].

### 4.2.1 Data Set Description

MUUFL Gulfport hyperspectral data was collected by the CASI-1500 hyperspectral imager over the campus of the University of Southern Mississippi-Gulfpark in Long Beach, Mississippi in November 2010. The image is consisting of 72 bands with the wavelength range of 375 to 1050 nm and contains  $325 \times 337$  pixels in total [82]. The RGB image of MUUFL Gulfport data set is presented in Fig 4.8.

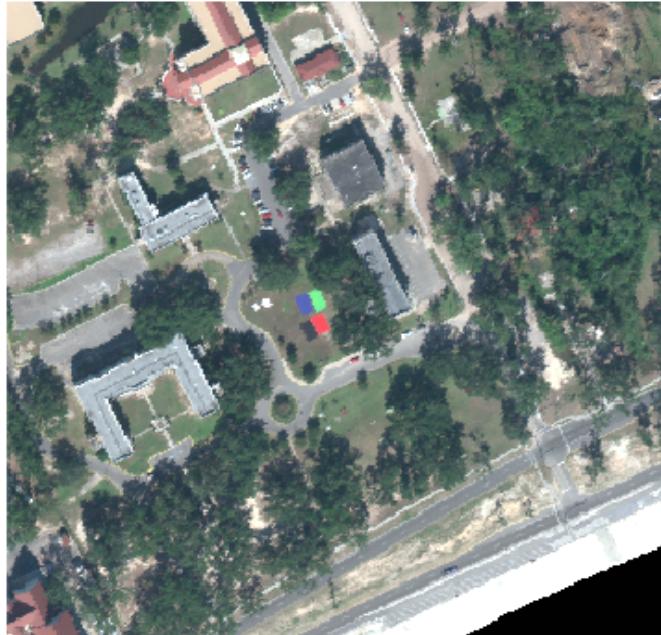
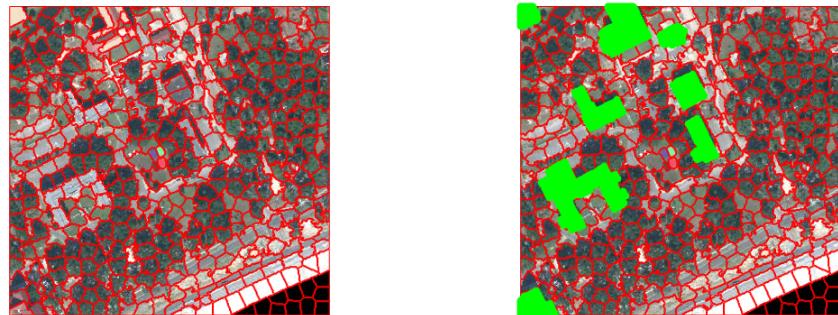


Figure 4.8: RGB Image of Gulfport Data

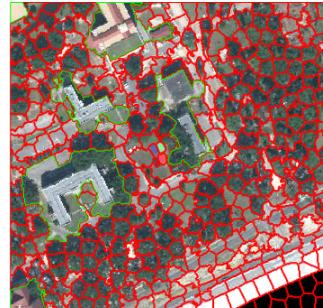
## 4.2.2 Superpixel Segmentation Results

First, an initial map-guided hyperspectral SLIC is perform on Gulfport data. The parameters for hyperspectral SLIC are the desired number of superpixels  $k = 500$  and the weighting parameter between spectral and spatial difference  $m = 20$ . The results of this process are presented in Fig 4.9.



(a) Superpixels by SLIC

(b) Buildings extracted from OpenStreetMap



(c) Superpixels by map-guided hyperspectral SLIC

Figure 4.9: Superpixels on Gulfport

For the hyperspectral unmixing process, the number of endmembers is set as 7. The desired endmembers are red roof, soil, light grey roof, shadow, asphalt, beach sand and vegetation. As a pre-processing step, all the pixel signatures in the data set are normalized

to have unit length as the input for all methods. The parameter setting for semi-supervised PM-LDA is  $K = 7$ ,  $\lambda = 1$ ,  $\alpha = 0.3$ ,  $\epsilon = 10\%$  and  $T = 200$ .

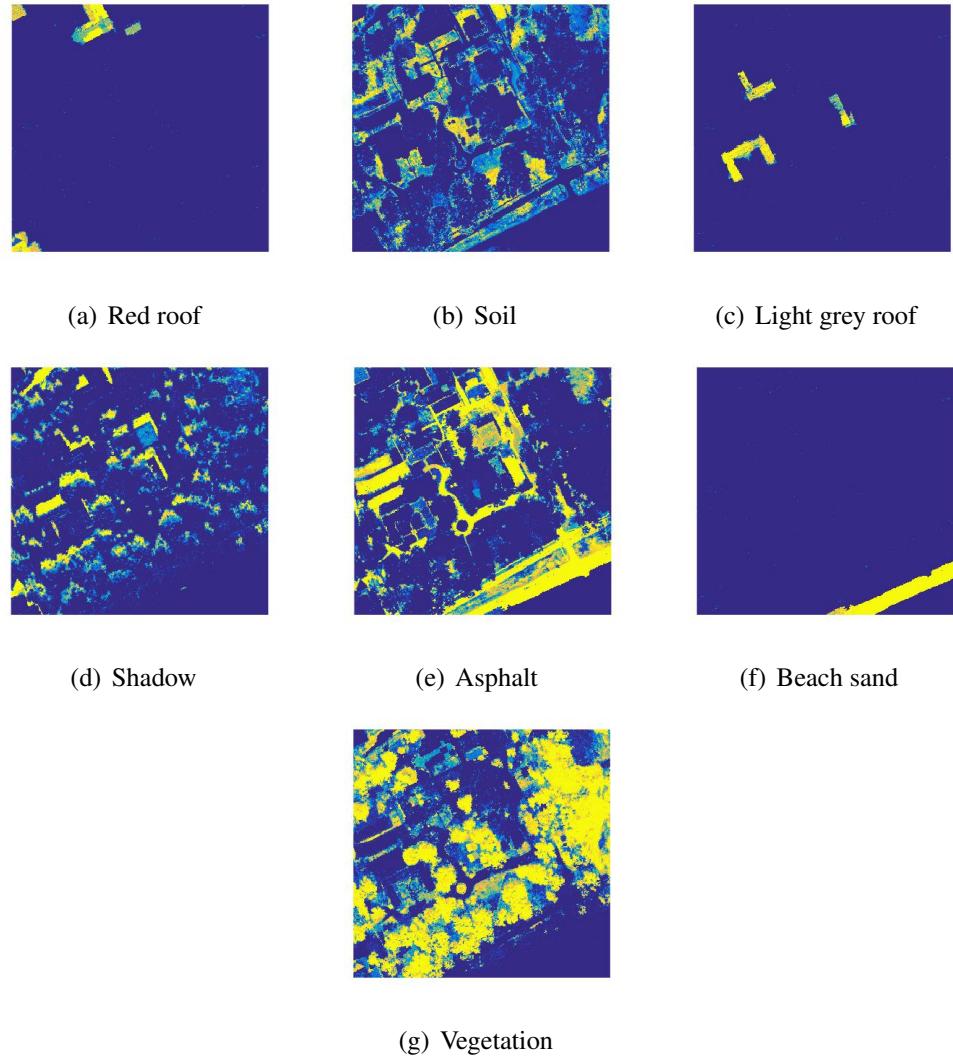


Figure 4.10: Estimated proportion maps for semi-supervised PM-LDA on Gulfport

After the proportion maps are obtained,  $K$ -means algorithm is used to perform the clustering and the “clean-up” process based on connected component algorithm is followed to construct the superpixels and enforce the connectivity. The parameter  $k$  as the number

of clusters for  $K$ -means algorithm is set to be 7 as same as the number of endmembers in the unmixing process.

5 other algorithms, which are SLIC, SLIC combines with OpenStreetMap (SLIC+OSM), hyperspectral normalize cut (HNC) [65], hyperspectral ultra contour map (HUCM)[64] and PM-LDA based superpixel segmentation are used to compare with our proposed method.

The parameter settings for all methods are selected as follows to yield the best performance by trial and error or from the propose paper: for SLIC and SLIC+OSM, the desired number of superpixels as  $K = 500$  and the weighting parameter between spectral and spatial difference as  $m = 20$ ; for HNC, the ratio of largest segment over all pixels as  $r = 0.01$  and the scale for spatial and spectral distance as  $sigma = 50$ ; for HUCM, the scale for the boundaries  $k = 0.1$ ; for PM-LDA, the parameter setting is same as supervised PM-LDA  $K = 7$ ,  $\lambda = 1$ ,  $\alpha = 0.3$  and  $T = 200$ .

The superpixel segmentation results for the comparison algorithms are presented in Fig 4.11. From Fig 4.11, we can see that the result returned by SLIC is oversegmented, and the result returned by SLIC+OSM is also oversegmented but can draw most building area as a superpixel. While for result obtained from HNC and HUCM, the superpixels are bigger than SLIC but it somehow clusters different materials into a superpixel incorrectly. For result obtained from PM-LDA based superpixel segmentation, many superpixels are noisy. However, the result obtained from our proposed algorithm almost can find every target material and segment the image into several superpixels based on the materials in that area.

We also can zoom in on the results to compare the details of the obtained superpixels. Using the red roof building in the bottom left corner as an example, the images are presented in Fig 4.12. From Fig 4.12, in the SLIC result, the result is fairly good, but it includes some shadow parts into the red roof building suerpixels; in the SLIC+OSM result,

the superpixel is kind of large, it though includes all red roof building parts but also includes some shadow and vegetation parts; in the HNC, HUCM and PM-LDA results, all of them partition the red roof building part into several superpixels and the segmentation results are very noisy. While for the result from our proposed method, it can segment the red roof building part into a superpixel correctly and also segment the shadow and vegetation parts into different superpixels.

We also zoom in another area which is the red building in the top middle of the image. The results obtained from different algorithms are shown in Fig 4.13. In the experiment, we treat the yellow and red building as a same material due to the similarity of these two endmembers. From Fig 4.13, in the SLIC result, it partition this building into several superpixels, so it is oversegmented; in the SLIC+OSM result, with the guidance from OSM, it can cover all building part, but it also includes some shadow into the building's superpixel; in the HNC and HUCM results, they both can identify some building parts but not all of them; and in the PM-LDA result, it can cover most building parts, but it is noisy. It separates the building and includes some vegetation parts into the superpixels. While in result from our proposed method, it can perfectly identify the building and include them in a superpixel correctly.

Quantitative evaluation is also performed to compare the performances of different algorithms on Gulfport data. Firstly, we use the original cluster validity which includes Dunn, DB and Silhouette indices to evaluate the segmentation results. Table 4.5 shows the validity values.

From Table 4.5, our proposed method performs best among these 6 algorithms with the maximum Dunn index, minimum Davies-Bouldin index and maximum Silhouette index. Based on the definition of these cluster validity indices, a small similarity within a super-

	Dunn	Davies-Bouldin	Silhouette
SLIC	$8.2197 \times 10^{-4} \pm 0.000035$	$12.7719 \pm 0.0843$	$-0.9059 \pm 0.0288$
SLIC+OSM	$9.0419 \times 10^{-4} \pm 0.000041$	$12.6382 \pm 0.0756$	$-0.8824 \pm 0.0217$
HNC	$0.0136 \pm 0.0082$	$12.5697 \pm 0.0942$	$-0.7950 \pm 0.0318$
HUCM	$0.0399 \pm 0.0076$	$9.3895 \pm 0.0432$	$-0.6591 \pm 0.0282$
PM-LDA	$0.0666 \pm 0.0125$	$8.0485 \pm 0.0541$	$-0.5461 \pm 0.0342$
Proposed method	<b><math>0.0776 \pm 0.0158</math></b>	<b><math>8.0345 \pm 0.0480</math></b>	<b><math>-0.5149 \pm 0.0279</math></b>

Table 4.5: Dunn, Davies-Bouldin, Silhouette indices and standard deviation on Gulfport

pixel and a big dissimilarity between superpixels can yield a better segmentation result. The SLIC and SLIC+OSM have the worst performance when evaluating by Dunn index. The reason for this is the oversegmentation from SLIC will give a small dissimilarity between superpixels. However, with the help from the OpenStreetMap, SLIC+OSM is a little better. The results from PM-LDA based segmentation is fairly good, however, without the supervision supplied from OpenStreetMap, it misclassifies some different materials into one superpixel. This results in a large similarity within this superpixel. Thus, the validity values are not as good as our proposed method.

	Modified Dunn	Modified Davies-Bouldin
SLIC	$2.2661 \times 10^{-15} \pm 0.0$	$2293.18959 \pm 51.3345$
SLIC+OSM	$2.6411 \times 10^{-15} \pm 0.0$	$2004.9428 \pm 48.4523$
HNC	$0.0114 \pm 0.0026$	$1785.3369 \pm 32.6512$
HUCM	$0.0430 \pm 0.0022$	$1289.3697 \pm 43.8725$
PM-LDA	$0.1033 \pm 0.0437$	$429.2411 \pm 39.3742$
Proposed method	<b><math>0.2601 \pm 0.0352</math></b>	<b><math>192.1993 \pm 38.4756</math></b>

Table 4.6: Modified Dunn, Davies-Bouldin indices and standard deviation on Gulfport

From Table 4.6, it is clear that our proposed method still outperforms others. The biggest modified Dunn index and the smallest modified Davies-Bouldin index can be used to indicate that our superpixel segmentation result has a small similarity within each superpixel and a large dissimilarity between neighboring superpixels. From the results from

SLIC and SLIC+OSM, we still can find that map guidance improves the performance overall. This point is also proved when comparing the results from PM-LDA based segmentation with our proposed method.

We also test our proposed method on Gulfport data with different parameter setting for  $k$  which is the desired number of endmembers for semi-supervised PM-LDA, also the desired number of clusters for  $K$ -means.

It is clear from Fig 4.14 that when the parameter  $k$  is chosen around 7 which the number of endmembers, the results are obviously better. When  $k = 3$ , we can see it misclassifies many materials into a superpixel. While  $k$  is large like 10 or 20, the image is oversegmented.

In order to evaluate our suggestion to choose parameter  $k$ , we use cluster validity indices to measure the performance with different parameter setting. The results of Dunn, Davies-Bouldin and Silhouette indices are presented in Table 4.3. From Table 4.7, Dunn and silhouette indices achieve the maximum value and Davies-Bouldin index achieves minimum value when  $k = 7$ . This indicates that when  $k = 7$  the superpixel segmentation from our proposed method has a small similarity for each superpixel and a large dissimilarity between superpixels. These are also proved in Fig 4.14. When  $k = 7$ , the superpixels are uniform and very distinguished from others. It almost can group a major material into one superpixel.

We also evaluate the results with different parameter  $k$  by our proposed modified Dunn and Davies-Bouldin indices. The validity values and standard deviations are presented in Table 4.8.

From Table 4.4, when  $k = 7$ , modified Dunn index has largest value and modified Davies-Bouldin index has smallest value. This indicates  $k = 7$  is a proper underlying

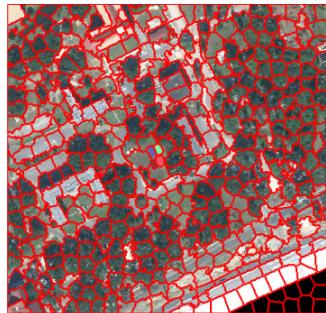
	Dunn	Davies-Bouldin	Silhouette
$k = 3$	$0.0433 \pm 0.012$	$11.4155 \pm 0.0612$	$-7133 \pm 0.0218$
$k = 5$	$0.0768 \pm 0.0136$	$9.0775 \pm 0.0745$	$-0.5421 \pm 0.0301$
$k = 7$	<b><math>0.0776 \pm 0.0158</math></b>	<b><math>8.0345 \pm 0.0572</math></b>	<b><math>-0.5149 \pm 0.0279</math></b>
$k = 10$	$0.0352 \pm 0.013$	$10.7060 \pm 0.0425$	$-0.5791 \pm 0.0318$
$k = 20$	$0.0116 \pm 0.014$	$17.4260 \pm 0.0453$	$-0.6939 \pm 0.0224$

Table 4.7: Dunn, Davies-Bouldin, Silhouette indices and standard deviation on Gulfport when change parameter  $k$

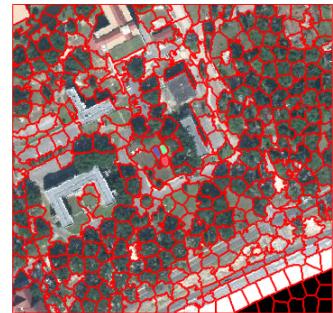
	Modified Dunn	Modified Davies-Bouldin
$k = 3$	$0.0815 \pm 0.0103$	$575.0662 \pm 36.4352$
$k = 5$	$0.1758 \pm 0.0278$	$238.5556 \pm 34.2915$
$k = 7$	<b><math>0.2601 \pm 0.0352</math></b>	<b><math>192.1993 \pm 38.4756</math></b>
$k = 10$	$0.1135 \pm 0.0314$	$315.1223 \pm 42.6458$
$k = 20$	$0.0623 \pm 0.0211$	$2286.2325 \pm 63.4752$

Table 4.8: Modified Dunn, Davies-Bouldin indices and standard deviation on Gulfport when change parameter  $k$

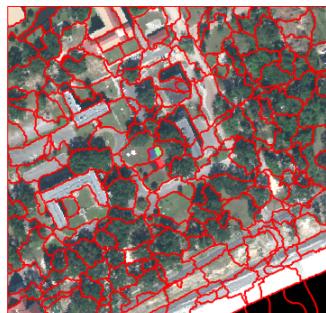
number of clusters in the Gulfport data. Based on the definitions of the modified indices, a smaller similarity within each superpixel and a larger dissimilarity between the neighboring superpixel will give a larger modified Dunn index and a smaller modified Davies-Bouldin index. This means the returned superpixels can partition the image well by grouping one material into one superpixel.



(a) SLIC



(b) SLIC+OSM



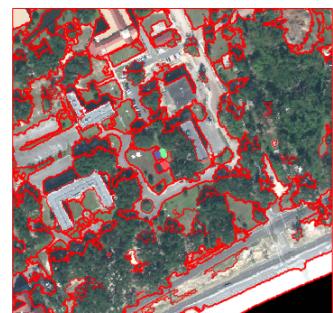
(c) HNC



(d) HUCM



(e) PM-LDA



(f) Proposed method

Figure 4.11: Superpixel Segmentation on Gulfport with 6 algorithms



(a) SLIC



(b) SLIC+OSM



(c) HNC



(d) HUCM



(e) PM-LDA



(f) Proposed method

Figure 4.12: Zoom in to the red roof building area on superpixel segmentation results



(a) SLIC



(b) SLIC+OSM



(c) HNC



(d) HUCM



(e) PM-LDA

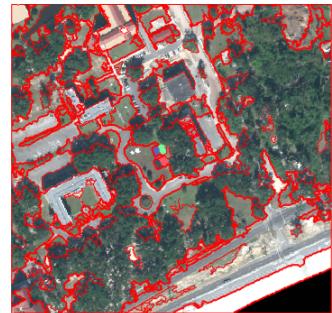


(f) Proposed method

Figure 4.13: Zoom in to another red roof building area on superpixel segmentation results



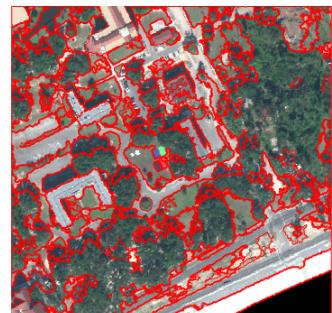
(a)  $k = 3$



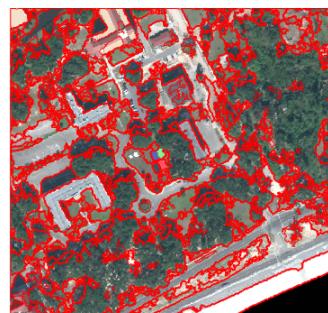
(b)  $k = 6$



(c)  $k = 7$



(d)  $k = 10$



(e)  $k = 20$

Figure 4.14: Superpixel segmentation results on Gulfport when change parameter  $k$

# Chapter 5

## Conclusion and Future Work

The focus of this thesis is to solve the problem of hyperspectral image superpixel segmentation. In chapter 2, some existing hyperspectral image segmentation algorithms are reviewed. Most algorithms start with a dimension reduction process, usually performed by PCA. The essential process of these methods is partitioning the data by comparing the spectral and spatial distances between each segment. The proposed method is based on the hyperspectral unmixing with the map guidance, then segmenting the proportion maps based on the  $K$ -means algorithm. Simply speaking, we are doing dimension reduction by unmixing which has more physical meaning than PCA.

The proposed method starts with an initial superpixel segmentation by map-guided hyperspectral SLIC. The map information extracted from social media like OpenStreetMap can supply useful labels for some areas like some buildings in the image. Given a few labeled superpixels, semi-supervised PM-LDA is used to perform the hyperspectral unmixing. Then we use  $K$ -means algorithm to segment the proportion maps and incorporate a “clean-up” process by assigning the small unconnected segments to the largest neighbor-

ing superpixels. The combination of these steps result in an approach for HSI.

Our proposed method maintains a very good performance in two real hyperspectral data of University of Pavia and MUUFL Gulfport. In the visualization superpixel segmentation results, it is clear that our result has more accurate segmentation and better performance in the detail area compared with another 5 different algorithms. We also evaluate the results by the cluster validity such as Dunn, Davies-Bouldin (DB) and Silhouette indices. Based on these values, our proposed method also has a better performance than others. To make the cluster validity more suitable with superpixel case, we modify the Dunn and DB indices by just considering the neighboring superpixels when the dissimilarities between different segments are calculated. The returned results still prove our advancement compared with the other 5 algorithms.

There are much future works to be done. Most importantly, speeding up the algorithm is an important area of study. The proposed algorithm is time consuming during the hyperspectral unmixing. One approach may be to leverage GPUs.

# Bibliography

- [1] Robert W Basedow, Dwayne C Carmer, and Mark E Anderson. Hydice system: Implementation and performance. In *SPIE's 1995 Symposium on OE/Aerospace Sensing and Dual Use Photonics*, pages 258–267. International Society for Optics and Photonics, 1995.
- [2] Gregg Vane, Robert O Green, Thomas G Chrien, Harry T Enmark, Earl G Hansen, and Wallace M Porter. The airborne visible/infrared imaging spectrometer (aviris). *Remote Sensing of Environment*, 44(2-3):127–143, 1993.
- [3] Martin Habermeyer. Proceedings of the 3rd earsel workshop on imaging spectroscopy, herrsching, germany 13-16 may 2003. 2003.
- [4] David R Thompson, Rebecca Castano, and Martha S Gilmore. Sparse superpixel unmixing for exploratory analysis of crism hyperspectral images. In *2009 First Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4. IEEE, 2009.
- [5] Mario Parente, JL Bishop, and James.F Bell. Spectral unmixing for mineral identification in pancam images of soils in gusev crater, mars. *Icarus*, 203(2):421–436, 2009.

- [6] Jason M Brenchley, David A Price, Timothy W Schacker, Tedi E Asher, Guido Silvestri, Srinivas Rao, Zachary Kazzaz, Ethan Bornstein, Olivier Lambotte, Daniel Altman, et al. Microbial translocation is a cause of systemic immune activation in chronic hiv infection. *Nature medicine*, 12(12):1365–1371, 2006.
- [7] Reza Barati, Sajjad Rahimi, and Gholam Hossein Akbari. Analysis of dynamic wave model for flood routing in natural rivers. *Water Science and Engineering*, 5(3):243–258, 2012.
- [8] Michael J Abrams and Douglas C Comer. Multispectral and hyperspectral technology and archaeological applications. In *Mapping Archaeological Landscapes from Space*, pages 57–71. Springer, 2013.
- [9] Stephen H Savage, Thomas E Levy, and Ian W Jones. Archaeological remote sensing in jordans faynan copper mining district with hyperspectral imagery. In *Mapping Archaeological Landscapes from Space*, pages 97–110. Springer, 2013.
- [10] Wayne Hu. Power spectrum tomography with weak lensing. *The Astrophysical Journal Letters*, 522(1):L21, 1999.
- [11] John B Adams, Milton O Smith, and Paul E Johnson. Spectral mixture modeling: A new analysis of rock and soil types at the viking lander 1 site. *Journal of Geophysical Research: Solid Earth*, 91(B8):8098–8112, 1986.
- [12] Nirmal Keshava and John F Mustard. Spectral unmixing. *IEEE signal processing magazine*, 19(1):44–57, 2002.
- [13] José M Bioucas-Dias, Antonio Plaza, Nicolas Dobigeon, Mario Parente, Qian Du, Paul Gader, and Jocelyn Chanussot. Hyperspectral unmixing overview: Geometrical,

statistical, and sparse regression-based approaches. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 5(2):354–379, 2012.

- [14] Robert B Singer and Thomas B McCord. Mars-large scale mixing of bright and dark surface materials and implications for analysis of spectral reflectance. In *Lunar and Planetary Science Conference Proceedings*, volume 10, pages 1835–1848, 1979.
- [15] Xiaoxiao Du. Accounting for spectral variability in hyperspectral unmixing using beta endmember distribution. Master’s thesis, Univ. of Missouri, Columbia, MO, Dec. 2013.
- [16] Mathieu Fauvel, Jocelyn Chanussot, and Jon Atli Benediktsson. Kernel principal component analysis for feature reduction in hyperspectrale images analysis. In *Proceedings of the 7th Nordic Signal Processing Symposium-NORSIG 2006*, pages 238–241. IEEE, 2006.
- [17] Bin Luo and Jocelyn Chanussot. Unsupervised classification of hyperspectral images by using linear unmixing algorithm. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 2877–2880. IEEE, 2009.
- [18] José MP Nascimento and José MB Dias. Vertex component analysis: A fast algorithm to unmix hyperspectral data. *IEEE transactions on Geoscience and Remote Sensing*, 43(4):898–910, 2005.
- [19] Alina Zare and Paul Gader. Sparsity promoting iterated constrained endmember detection in hyperspectral imagery. *IEEE Geoscience and Remote Sensing Letters*, 4(3):446, 2007.

- [20] Alina Zare, Paul Gader, Ouiem Bchir, and Hichem Frigui. Piecewise convex multiple-model endmember detection and spectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 51(5):2853–2862, 2013.
- [21] Chao Chen, Alina Zare, and J Tory Cobb. Partial membership latent dirichlet allocation for image segmentation.
- [22] Joseph W Boardman, Fred A Kruse, and Robert O Green. Mapping target signatures via partial unmixing of aviris data. 1995.
- [23] Johnston Neville. Automatic endmember extraction from hyperspectral data for mineral exploration. In *International Airborne Remote Sensing Conference and Exhibition, 4 th/21 st Canadian Symposium on Remote Sensing, Ottawa, Canada*, 1999.
- [24] Michael E Winter. N-findr: an algorithm for fast autonomous spectral end-member determination in hyperspectral data. In *SPIE’s International Symposium on Optical Science, Engineering, and Instrumentation*, pages 266–275. International Society for Optics and Photonics, 1999.
- [25] João Rosário, José MP Nascimento, and Mário Véstias. Fpga-based architecture for hyperspectral endmember extraction. In *SPIE Remote Sensing*, pages 924703–924703. International Society for Optics and Photonics, 2014.
- [26] Stephen Wright and Jorge Nocedal. Numerical optimization, vol. 2springer. *New York*, 1999.
- [27] Mark Berman, Harri Kiiveri, Ryan Lagerstrom, Andreas Ernst, Rob Dunne, and Jonathan F Huntington. Ice: A statistical approach to identifying endmembers

in hyperspectral images. *IEEE transactions on Geoscience and Remote Sensing*, 42(10):2085–2095, 2004.

- [28] Taylor Glenn, Alina Zare, and Paul Gader. Bayesian fuzzy clustering. *IEEE Transaction on Fuzzy Systems*, 8(1545-1561):2–4, 2015.
- [29] Martin Ehler and Matthew Hirn. Sparse endmember extraction and demixing. In *IGARSS*, pages 1385–1388, 2012.
- [30] PIYUSH PRADEEP KHOPKAR. Hyperspectral unmixing and band weighting for multiple endmember sets. Master’s thesis, University of Missouri-Columbia, Columbia, MO, 2014.
- [31] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.
- [32] Gareth O Roberts and Adrian FM Smith. Simple conditions for the convergence of the gibbs sampler and metropolis-hastings algorithms. *Stochastic processes and their applications*, 49(2):207–216, 1994.
- [33] Sheng Zou. Semi-supervised interactive unmixing for hyperspectral image analysis. Master’s thesis, University of Missouri-Columbia, Columbia, MO, 2016.
- [34] King-Sun Fu and JK Mui. A survey on image segmentation. *Pattern recognition*, 13(1):3–16, 1981.
- [35] Jos BTM Roerdink and Arnold Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta informaticae*, 41(1, 2):187–228, 2000.

- [36] Serge Beucher and Christian Lantu joul. Use of watersheds in contour detection. 1979.
- [37] Leila Shafarenko, Maria Petrou, and Josef Kittler. Automatic watershed segmentation of randomly textured color images. *IEEE transactions on Image Processing*, 6(11):1530–1544, 1997.
- [38] Yuliya Tarabalka, Jocelyn Chanussot, and Jon Atli Benediktsson. Segmentation and classification of hyperspectral images using watershed transformation. *Pattern Recognition*, 43(7):2367–2379, 2010.
- [39] John A Richards and JA Richards. *Remote sensing digital image analysis*, volume 3. Springer, 1999.
- [40] Jing Wang and Chein-I Chang. Independent component analysis-based dimensionality reduction with applications in hyperspectral image analysis. *IEEE transactions on geoscience and remote sensing*, 44(6):1586–1600, 2006.
- [41] Guillaume Noyel, Jesus Angulo, and Dominique Jeulin. Morphological segmentation of hyperspectral images. *Image Anal. Stereol*, 26(3):101–109, 2007.
- [42] Adrian N Evans and Xin U Liu. A morphological gradient approach to color edge detection. *IEEE Transactions on Image Processing*, 15(6):1454–1463, 2006.
- [43] Yuliya Tarabalka, J n Atli Benediktsson, and Jocelyn Chanussot. Spectral–spatial classification of hyperspectral imagery based on partitional clustering techniques. *IEEE Transactions on Geoscience and Remote Sensing*, 47(8):2973–2987, 2009.
- [44] GH Ball and DJ Hall. Isodata, a novel method of data analysis and classification, technique report, 1965.

- [45] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.
- [46] Edwin Diday and JC Simon. Clustering analysis. In *Digital pattern recognition*, pages 47–94. Springer, 1980.
- [47] Noritoshi Kamagata, Keitarou Hara, Masaru Mori, Yukio Akamatsu, Yunqing Li, and Yoshinobu Hoshino. A new method of vegetation mapping by object-based classification using high resolution satellite data. In *Proc. 1st Int. Conf. Object-Based Image Anal*, volume 36, page 4, 2006.
- [48] Soo Chin Liew, Chew Wai Chang, and Kim Hwa Lim. Hyperspectral land cover classification of eo-1 hyperion data by principal component analysis and pixel unmixing. In *Geoscience and Remote Sensing Symposium, 2002. IGARSS'02. 2002 IEEE International*, volume 6, pages 3111–3113. IEEE, 2002.
- [49] Mathieu Fauvel, Yuliya Tarabalka, Jon Atli Benediktsson, Jocelyn Chanussot, and James C Tilton. Advances in spectral-spatial classification of hyperspectral images. *Proceedings of the IEEE*, 101(3):652–675, 2013.
- [50] Linda Shapiro and George C Stockman. Computer vision. 2001. ed: Prentice Hall, 2001.
- [51] Scott G Beaven, David Stein, and LE Hoff. Comparison of gaussian mixture and linear mixture models for classification of hyperspectral data. In *Geoscience and Remote Sensing Symposium, 2000. Proceedings. IGARSS 2000. IEEE 2000 International*, volume 4, pages 1597–1599. IEEE, 2000.

- [52] Nicola Acito, Giancarlo Corsini, and Mario Diani. An unsupervised algorithm for hyperspectral image segmentation based on the gaussian mixture model. In *Geoscience and Remote Sensing Symposium, 2003. IGARSS'03. Proceedings. 2003 IEEE International*, volume 6, pages 3745–3747. IEEE, 2003.
- [53] Yuliya Tarabalka, James C Tilton, Jón Atli Benediktsson, and Jocelyn Chanussot. Marker-based hierarchical segmentation and classification approach for hyperspectral imagery. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1089–1092. IEEE, 2011.
- [54] Antonio J Plaza and James C Tilton. Automated selection of results in hierarchical segmentations of remotely sensed hyperspectral images. Technical report, DTIC Document, 2005.
- [55] Yuliya Tarabalka and James C Tilton. Spectral-spatial classification of hyperspectral images using hierarchical optimization. In *2011 3rd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4. IEEE, 2011.
- [56] Kevin Bernard, Yuliya Tarabalka, Jesus Angulo, Jocelyn Chanussot, and Jón Atli Benediktsson. Spectral–spatial classification of hyperspectral data based on a stochastic minimum spanning forest approach. *IEEE Transactions on Image Processing*, 21(4):2008–2021, 2012.
- [57] David R Thompson, Lukas Mandrake, Martha S Gilmore, and Rebecca Castaño. Superpixel endmember detection. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11):4023–4033, 2010.

- [58] Arun M Saranathan and Mario Parente. Uniformity-based superpixel segmentation of hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 54(3):1419–1430, 2016.
- [59] Guillaume Tochon, Jean-Baptiste Feret, Roberta E Martin, Raul Tupayachi, Jocelyn Chanussot, and Gregory P Asner. Binary partition tree as a hyperspectral segmentation tool for tropical rainforests. In *2012 IEEE International Geoscience and Remote Sensing Symposium*, pages 6368–6371. IEEE, 2012.
- [60] Silvia Valero, Philippe Salembier, and Jocelyn Chanussot. Hyperspectral image representation and processing with binary partition trees. *IEEE Transactions on Image Processing*, 22(4):1430–1443, 2013.
- [61] Miguel A Veganzones, Guillaume Tochon, Mauro Dalla-Mura, Antonio J Plaza, and Jocelyn Chanussot. Hyperspectral image segmentation using a new spectral unmixing-based binary partition tree representation. *IEEE Transactions on Image Processing*, 23(8):3574–3589, 2014.
- [62] Miguel Angel Veganzones, Guillaume Tochon, Mauro Dalla Mura, Antonio J Plaza, and Jocelyn Chanussot. Hyperspectral image segmentation using a new spectral mixture-based binary partition tree representation. In *2013 IEEE International Conference on Image Processing*, pages 245–249. IEEE, 2013.
- [63] Pegah Massoudifar, Anand Rangarajan, and Paul Gader. Superpixel estimation for hyperspectral imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 287–292, 2014.
- [64] Manu Sethi, Yupeng Yan, Anand Rangarajan, Ranga Raju Vatsavai, and Sanjay

- Ranka. Scalable machine learning approaches for neighborhood classification using very high resolution remote sensing imagery. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2069–2078. ACM, 2015.
- [65] David B Gillis and Jeffrey H Bowles. Hyperspectral image segmentation using spatial-spectral graphs. In *SPIE Defense, Security, and Sensing*, pages 83901Q–83901Q. International Society for Optics and Photonics, 2012.
- [66] Yuliya Tarabalka, Jón Atli Benediktsson, Jocelyn Chanussot, and James C Tilton. Multiple spectral–spatial classification approach for hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 48(11):4122–4132, 2010.
- [67] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2):167–181, 2004.
- [68] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2011.
- [69] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.
- [70] Sergios Theodoridis and Konstantinos Koutroumbas. Chapter 16 - cluster validity. In Sergios Theodoridis and Konstantinos Koutroumbas, editors, *Pattern Recognition (Fourth Edition)*, pages 863 – 913. Academic Press, Boston, fourth edition edition, 2009.

- [71] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics*, 4(1):95–104, 1974.
- [72] Ferenc Kovács, Csaba Legány, and Attila Babos. Cluster validity measurement techniques. In *6th International symposium of hungarian researchers on computational intelligence*. Citeseer, 2005.
- [73] Nikhil R Pal and J Biswas. Cluster validation using graph theoretic concepts. *Pattern Recognition*, 30(6):847–857, 1997.
- [74] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, (2):224–227, 1979.
- [75] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [76] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282, 2012.
- [77] Christine Connolly and T Fleiss. A study of efficiency and accuracy in the transformation from rgb to cielab color space. *IEEE Transactions on Image Processing*, 6(7):1046–1048, 1997.
- [78] Xuewen Zhang, Selene E Chew, Zhenlin Xu, and Nathan D Cahill. Slic superpixels for efficient graph-based dimensionality reduction of hyperspectral imagery. In *SPIE Defense+ Security*, pages 947209–947209. International Society for Optics and Photonics, 2015.

- [79] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.
- [80] James MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.
- [81] John Hopcroft and Robert Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.
- [82] Sheng Zou and Alina Zare. Instance influence estimation for hyperspectral target signature characterization using extended functions of multiple instances. In *SPIE Defense+ Security*, pages 98400S–98400S. International Society for Optics and Photonics, 2016.
- [83] Paul Gader and Alina Zare. Muufl gulfport hyperspectral and lidar airborne data set. *University of Florida, Gainesville, FL, USA, Tech. Rep. REP-2013-570*, 2013.