12-2016

Graph-based Data Modeling and Analysis for Data Fusion in Remote Sensing

Lei Fan
lxf3548@rit.edu

Graph-based Data Modeling and Analysis for Data Fusion in Remote Sensing

by

Lei Fan

B.S. Optical Engineering, Nanjing University of Science and Technology, 2011

A dissertation submitted in partial fulfillment of the
requirements for the degree of Doctor of Philosophy
in the Chester F. Carlson Center for Imaging Science
College of Science
Rochester Institute of Technology

December, 2016

Signature of the Author ______________________________

Accepted by ______________________________
Coordinator, Ph.D. Degree Program Date

CHESTER F. CARLSON CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

Ph.D. DEGREE DISSERTATION

The Ph.D. Degree Dissertation of Lei Fan
has been examined and approved by the
dissertation committee as satisfactory for the
dissertation required for the
Ph.D. degree in Imaging Science

____
Dr. David Messinger, Dissertation Advisor, Date

____
Dr. David Ross, External Chair, Date

____
Dr. Anthony Vodacek, Date

____
Dr. Sildomar Monteiro, Date

Graph-based Data Modeling and Analysis for Data Fusion in Remote Sensing

by

Lei Fan

Submitted to the
Chester F. Carlson Center for Imaging Science
in partial fulfillment of the requirements
for the Doctor of Philosophy Degree
at the Rochester Institute of Technology

Abstract

Hyperspectral imaging provides the capability of increased sensitivity and discrimination over traditional imaging methods by combining standard digital imaging with spectroscopic methods. For each individual pixel in a hyperspectral image (HSI), a continuous spectrum is sampled as the spectral reflectance/radiance signature to facilitate identification of ground cover and surface material. The abundant spectrum knowledge allows all available information from the data to be mined. The superior qualities within hyperspectral imaging allow wide applications such as mineral exploration, agriculture monitoring, and ecological surveillance, etc. The processing of massive high-dimensional HSI datasets is a challenge since many data processing techniques have a computational complexity that grows exponentially with the dimension. Besides, a HSI dataset may contain a limited number of degrees of freedom due to the high correlations between data points and among the spectra. On the other hand, merely taking advantage of the sampled spectrum of individual HSI data point may produce inaccurate results due to the mixed nature of raw HSI data, such as mixed pixels, optical interferences and etc.

Fusion strategies are widely adopted in data processing to achieve better performance, especially in the field of classification and clustering. There are mainly three types of fusion strategies, namely low-level data fusion, intermediate-level feature fusion, and high-level decision fusion. Low-level data fusion combines multi-source data that is expected to be complementary or cooperative. Intermediate-level feature fusion

aims at selection and combination of features to remove redundant information. Decision level fusion exploits a set of classifiers to provide more accurate results. The fusion strategies have wide applications including HSI data processing. With the fast development of multiple remote sensing modalities, e.g. Very High Resolution (VHR) optical sensors, LiDAR, etc., fusion of multi-source data can in principal produce more detailed information than each single source. On the other hand, besides the abundant spectral information contained in HSI data, features such as texture and shape may be employed to represent data points from a spatial perspective. Furthermore, feature fusion also includes the strategy of removing redundant and noisy features in the dataset.

One of the major problems in machine learning and pattern recognition is to develop appropriate representations for complex nonlinear data. In HSI processing, a particular data point is usually described as a vector with coordinates corresponding to the intensities measured in the spectral bands. This vector representation permits the application of linear and nonlinear transformations with linear algebra to find an alternative representation of the data. More generally, HSI is multi-dimensional in nature and the vector representation may lose the contextual correlations. Tensor representation provides a more sophisticated modeling technique and a higher-order generalization to linear subspace analysis.

In graph theory, data points can be generalized as nodes with connectivities measured from the proximity of a local neighborhood. The graph-based framework efficiently characterizes the relationships among the data and allows for convenient mathematical manipulation in many applications, such as data clustering, feature extraction, feature selection and data alignment. In this thesis, graph-based approaches applied in the field of multi-source feature and data fusion in remote sensing area are explored. We will mainly investigate the fusion of spatial, spectral and LiDAR information with linear and multilinear algebra under graph-based framework for data clustering and classification problems.

Acknowledgements

Pursuing P.hD. degree in Imaging Science here at R.I.T has become one of the greatest experiences of mine. I enjoyed my talking with every professor and staff here in Chester F. Carlson Center, and I am truly proud of being one of the Imaging Science family.

My deep gratitude goes first to my advisor Professor David Messinger, who expertly and patiently guided me through my five years study and life in Rochester. His broad knowledge in remote sensing field and enthusiasm for imaging science has always encouraged me and constantly engaged me with my research. His skillful guidance, innovative ideas and tolerant altitude are greatly appreciated. Apart from being a wonderful academic advisor and program director, Professor David Messinger is also a great life and career mentor to me.

I would like to express my sincere appreciation to my committee members: Dr. Anthony Vodacek, Dr. Sildomar Monteiro, Dr. David Ross and Dr. Tony Harkin. Though Dr. Tony Harkin can no longer serve on my committee due to a health problem, I want to express my gratitude to him for providing insightful advice to my research and I want to send my best wishes to him. I am extremely grateful to all my committee members for being supportive to my proposed work and providing to me brilliant suggestions and comments. Special thanks to Dr. David Ross for agreeing to serve on my thesis committee even when he has not even met me. I am honored to have Dr. David Ross as my external chair during Dr. Tony Harkin's absence.

I also would like to sincerely thank all the faculty and staff in the Chester F. Carlson Center for their enthusiasm in imaging science and their willingness to offer help in every way. I am grateful to Cindy Schultz, Sue Chan, Marilyn Lockwood and Joyce French for always being willing to help me with all kinds of paperwork and school affairs. A special mention to Cindy Schultz, greatly missed, who always had her door open to me and had always been cheerful to everyone.

Lots of thanks to all my fellow officemates, classmates and friends at R.I.T. I learnt a lot from everyone and enjoyed the five years life with them in Rochester. I also wish to thank the staff, Susan Joseph and Jeffrey Cox, from the International Student Service Center for always providing me great suggestions and offering me prompt help with my visa to work and study in the US. Also, I want to express my great gratitude to my mentors, Alex Loui, Basak Oztan and Elizabeth Edmunds, during my three internships experiences, respectively. Without their support and guidance, I can never gain such valuable industry experiences and great memories with co-workers.

Above ground, I am indebted to my family in China, whose value to me only grows with time. My deepest gratitude to my parents: Aihua Han and Sixin Fan, for their unconditional love and support at all times. Also, I want to thank my parents in-law: Xiulan Shao and Yuliang Wang, for their encouragement and understanding. And finally, I acknowledge my husband, Xiyu, who is my greatest champion, life companion, my anchor and best friend for the past and for the future. Without them, my thesis would not have been possible.

In the end, I thank all my friends for their supports and encouragements to me all the time. I also would like to extend my thanks to those who has helped me in every aspects in my life.

To my beloved family and friends

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation and Background

1.1.1 Multi-modalities of Remotely Sensed Data

Remote sensing is the acquisition and analysis of information from objects without using an instrument to collect the data in direct physical contact with the object. For the past decade, hyperspectral remote sensing has been an area of fast development and active research due to its ability to make a dense sampling of the spectrum. HSI data contains a set of images where each image is captured at a narrow range of wavelength in the electromagnetic spectrum. All these gray-scale images are formed as a three dimensional HSI data cube, with two spatial dimensions of the scene and one spectral dimension. An HSI pixel is sampled across the third dimension at a particular spatial location within the data, resulting in a one-dimensional spectrum vector. The spectrum of each pixel can be used to determine and characterize different materials present in a given scene, based on the unique spectral signatures. Compared to traditional RGB images or multi-spectral images, HSI data provides enhanced power in object identification and characterization [1]. The VNIR/SWIR portion of the spectrum in HSI deals with the reflective properties of solids and liquid materials, and MWIR is useful for identifying specific gases. The MWIR and LWIR range in the spectrum examines the unique emissive properties of materials regardless of day and night. Furthermore, HSI holds strong detectability of sub-pixel target by exploiting finer detail in the spectral signatures of targets and natural backgrounds [2].

With recent hyperspectral imaging technologies, HSI not only provides detailed spectra with great distinguishability in the spectral ”fingerprints”, it also provides higher spatial resolutions. Conventional HSI data processing approaches exploit the spectral signatures of individual pixels without considering the contextual information in the image domain. To enhance the accuracy, the dependency between a pixel and its surrounding neighbors could be fused into the HSI data analysis. The dependencies among neighbor data points could be understood as texture, shape and other local structure features, and be further generalized as an additional information source complementary to spectral information.

Hyperspectral imaging is categorized as passive sensing which gathers radiation that is emitted or reflected by the objects. Active sensing, on the other hand, emits energy to scan target objects then detects and measures the reflected or backscattered radiation from them. Synthetic aperture radar (SAR), Radar, and LiDAR are examples of active remote sensing. SAR imaging operates at the microwave range in the electromagnetic spectrum and can acquire data through clouds. A pixel in a SAR image may contain information such as radar backscattering intensity and phase, range measurement between the sensor and a reflecting object, etc. LiDAR instruments illuminate a target with a laser and analyze the reflected light. It measures the heights of objects and structures on the ground more precisely than using radar technologies. Also, it can make the three-dimensional positioning of objects in the scene well defined. LiDAR sensors are able to produce a 3D point cloud from which the Digital Terrain Models (DTM), Digital Surface Models (DSM) and 3D models of an object could be further calculated [3]. With the fast development of different remote sensors, the multi-modalities of remotely sensed data may promote the development of data fusion strategies for a better understanding of the area under investigation.

1.1.2 Remote Sensing Data Fusion

The multi-modalities of remotely sensed data provide more possibilities of information exploitation compared to a single data source. Different sensing devices capture different physical characteristics of the same scene and combines to enhance the understanding of the target objects. They also provide the basis for discrimination, identification and characterization. Hyperspectral images have fine spectral resolution to determine and characterize different materials based on their spectral signatures. It may also provide the possibilities of extracting spatial information in the spatial domain. LiDAR sensors collect 3D point clouds that can be used to extract elevation and curvature information to further provide separabilities between different structures. Other than the emergence of multi-sensor data and multi-feature information, the multi-temporal data, revealed by the ambition of the remote sensing community to develop new generation of sensors, also has gained increased attention. The development of novel data fusion techniques to address new challenging applications is demanding.

Remote sensing data fusion aims to integrate multi-source information generated from different perspectives, acquired with different sensors or captured at different times in order to produce fused data that contain more enriched information than one individual data source. In this procedure, multiple data sources would result in an increase in data volume, dimensionality of the feature space, and interclass separability. For the clustering and classification tasks of remotely sensed data, especially the high dimensional HSI data, fusion strategies may also include the selection of a subset of features and the eliminating of redundant features. In machine learning and pattern recognition, the available data fusion techniques can be mainly classified into three categories, namely, low-level information/data fusion, intermediate-level feature fusion and high-level decision fusion. The low-level fusion combines multiple raw data sources to generate as new input data that takes the advantages of complementary or collaborative data sources. The intermediate-level feature fusion may include feature primitive identification, feature extraction/selection, and feature combination [4]. Compared to low-level pixels, feature primitives can include pixel intensity, edge, texture, shape, length or image segments, and etc. The identification and extraction of the feature primitives can generate higher level descriptions of the data, and can be fused as another information source to improve the data processing. Feature fusion also includes the selection and combination of features to remove redundant and irrelevant features for analysis. The integration of multiple data sources may easily result in high dimensional feature space and significant amounts of redundant information. Feature extraction/selection thereby refers to the step of finding only those features that contain a significant amount of information. The high-level decision fusion takes advantage of multiple classifiers to provide better accuracy and efficiency in the final decision. There are varieties of classifiers such as Support Vector Machine (SVM), kernel-based SVM [5], k-nearest neighbor (KNN), Gaussian Maximum Likelihood (GMM), etc. Different classifiers may hold certain assumptions of the data, or involve parametric modeling that is data dependent. To avoid the disagreements between different classifiers, the utilization of multiple classifiers can provide better generalization. In certain cases, the quantity of data to be processed is too large to be handled simultaneously. A multi-classifier framework will be efficient in which the data will be processed as multiple subsets with different classifiers [6].

In practical applications, the above fusion categories does not encompass all fusion methods. Usually, the applied fusion procedure is a combination of different level of fusion techniques as described above. Based on the data sources and combinations, remote sensing data fusion may be broadly categorized as optical panchromatic (PAN) and HSI data fusion, spatial and spectral feature fusion, LiDAR point cloud and HSI fusion, Multi-temporal data fusion, etc. The purpose of fusing PAN and HSI image is to improve spatial resolution and retain the spectral fidelity of the original HSI data. This is also referred to as pan-sharpening in the literature. The spatial and spectral feature fusion aims to exploit the spatial information contained in the remote sensing spectral data, which is helpful for the discrimination of materials with different structure or shape in the scene. The extracted spatial features can also be identified as an additional data source, it then allows us to use the spatial and spectral information by means of multi-source fusion techniques. The fusion of LiDAR data and imagery such as HSI has been explored in recent years. A LiDAR point cloud is a good source for generating a highly accurate DSM model. The elevation and gradient information combined with detailed spectral knowledge can facilitate the data analysis in urban area, such as building footprint extraction, land cover mapping, forest boundary detection, etc. Multi-temporal images have gained attention in the remote sensing community recently. The fusion of multi-temporal images provides a new spatio-spectra-temporal perspective of the remotely sensed data [7]. The multiple images captured at different time or using different sensors need to be properly aligned (i.e. the spatial co-registration and compensation for the changes). Other sources of remotely sensed data, for example, SAR, GIS data, range sensors and digital photogrammetry [8], and etc., have been fused with HSI in many applications, such as environmental monitoring, object detection and 3D reconstructions. Overall, remote sensing data fusion is a demanding research field as the fast emergence of multi-modality data.

1.1.3 Data Modeling and Processing with Graphs

A graph is a powerful tool to model pairwise relationships between data points. It has wide applications in computer vision, machine learning, and image processing. For remotely sensed data, a graph can be used to model the pairwise relationship among neighboring pixels, 3D point clouds, regions or clusters, multiple features, and etc.

HSI data points can be individually represented by vectors in a high dimensional feature space. Intensive work has been performed for HSI classification and clustering during the last few decades. Parametric models, such as Gaussian maximum likelihood and linear discriminant analysis, have been investigated for the classification of spectral images [9]. These statistical parametric models based on the estimation of covariances and means, can successfully deal with multispectral images but are less reliable for hyperspectral images due to their high dimensionality. Non-parametric and nonlinear models, such as Bayesian models, kernel methods and Support Vector Machines (SVM), Neural Networks and spectral clustering (as in spectral graph theory) have been investigated for hyperspectral data. In particular, spectral clustering, as a graph-based approach, has shown remarkable performance in terms of accuracy and simplicity. Graph models used in spectral clustering, with the useful mathematical tool such as the graph Laplacian, can transform HSI data into low-dimensional feature space where different clusters are more separable.

Spectral clustering is closely related to dimensionality reduction with graph representation [10]. The fusion of multiple data sources and features can easily result in high dimensionality which may gain much more computational complexity and suffer from the ”curse of dimensionality”. Graph representations of the data or features first build a neighborhood proximity graph in the high dimensional data space and embed this graph into a low-dimensional space and preserve most of the relationships among the original data. The purpose of a graph model used for dimensionality reduction is to provide a non-parametric model with no statistical assumptions being made for the data.

The graph model can also be used to model the relationships between multiple data sources, such as multi-temporal images. Though correspondences are difficult to model due to changes in the environment, the intrinsic structures of classes contained in these sequential images are similar. Manifold alignment [11] constructs connections between disparate data sets by aligning their underlying individual manifolds and transferring information across the multiple data sets.

Additionally, data representation with a graph allows the possibility of performing combinatorial optimization to produce highly efficient solutions. For a binary case, the graph cut algorithm is the most well-known approach in computer vision that makes the energy-minimization problem directly translated to the minimum cut problem. Given a data set, data points can be efficiently labeled into two different groups by max-flow/min-cut optimization. Beyond the binary case, the energy minimization problem can be considered a maximum a posteriori (MAP) estimation problem in a Markov random field (MRF) framework which is based on undirected graphical model. Overall, graph model is a powerful tool for describing and understanding the structure of the data in many applications as described above.

1.2 Objective and Contributions

Data fusion aims to integrate data from multiple sources to enrich the knowledge about the target, and to gain better interpretation and understanding of it. With the rapid development of multiple types of remote sensors and deep exploration of the multimodal remotely sensed data, data fusion becomes an effective and demanding technique for optimum utilization of the remotely sensed data with large volume.

Recently, graph-based data modeling and processing techniques have emerged as useful tools for image data analysis. Many real-world problems have been successfully modeled with graph-based approaches in the literature. The number of concepts and processing skills that can be solved with graph representation is very large. Therefore, graph theory has found many developments and applications in the image processing area due to the feasibility of modeling pairwise relationships for discrete data. Different graph models have been proposed for image analysis, depending on the data structures and processing skills. Graphs not only provide an effective representation of the data, but also enable efficient graph-theoretical algorithms to process the data. Spectral methods, involving a graph Laplacian, directly take advantage of Eigen analysis using linear algebra. It has wide applications in clustering, dimensionality reduction and data alignment. Minimum cut/maximum network flow algorithms in graph theory can serve as a powerful tool for exact or approximate energy minimization for binary or multi-class labeling problems. The graph may also be naturally used as an efficient data encoding approach for hierarchical data representation.

Overall, the main objective of this research is to use and explore the graph-based modeling and processing techniques for data fusion, particularly in remotely sensed multi-source data analysis. More specifically, the work proposed in this thesis is an attempt to exploit graph models under multi-source data and feature fusion framework:

- As an effective modeling method for the extraction of features related to the materials and objects in a given scene. Serves as a preliminary step for other applications, such as classification and detection.
- As a tool to represent and incorporate multi-source data into the framework for the image interpretation.
- Provide mathematical fundamentals for data embedding, feature selection/extraction to reorganize input data and remove redundant information.

1.3 Outline of Thesis

This dissertation thesis summarizes the application of graph-based modeling and data fusion techniques on remotely sensed data clustering and classification. It is organized into six chapters that cover different topics related to the subject of interest in this research.

1. **Chapter 2**: An overview of data fusion strategies for remotely sensed data is presented in Chapter 2. With an increasing quantity of data captured by airborne/satellite sensors, fusion in the four fields become popular: the multi-spectral

image pan-sharpening, spatial-spectral data fusion, LiDAR and image fusion, multi-temporal data analysis. The combination of multiple source of information can greatly enhance the performance of applications related to image interpretation and understanding. In this chapter, remote sensing data classification, particularly HSI data classification, are introduced and explained. The remote sensing data classification aims at grouping observations to represent land cover features. The techniques used for classification could be categorized as unsupervised or supervised, pixel-level or object-level, etc.

2. **Chapter 3**: Graph theory has been widely used in data modeling and analysis for its strong mathematical foundation. In Chapter 3, the basic concepts and notations for spectral graph theory are explained. A topological algorithm that is purely based on a graph data structure is introduced for solving a HSI classification problem. Besides, the manifold learning techniques, which have been widely investigated for HSI data processing by represent ing the topology of the high-dimensional nonlinear data into a lower dimensional space, are discussed. An overview of manifold learning algorithms and their application to high-dimensional remotely sensed data are presented in this chapter.

3. **Chapter 4**: Spatial and spectral feature fusion in a manifold learning framework is described in Chapter 4. Remotely sensed data consists of a substantial amount of spatial and spectral information which brings a challenge to traditional image and signal processing. Methods for feature mining of HSI data are introduced to improve the use efficiency of the huge quantities of data. Algorithms based on improved affinity matrices that embedded with spatial and spectral information are proposed for efficient joint spatial-spectral HSI data clustering.

4. **Chapter 5**: High-order tensor is introduced for efficient multi-feature data representation. In HSI analysis, classification and segmentation usually represents each pixel as a vector in high-dimensional space and solves the mathematical problems use linear algebra, i.e., the algebra of matrices. Tensor-based image analysis has been explored in recent years. By adopting multilinear algebra, tensors can be efficiently used to combine multiple features, to represent super-pixels, and to reduce

the dimensionality of high dimensional data. In Chapter 5, algorithms for multi-feature fusion with high-order tensors are proposed and explained. Comparisons to traditional linear algebra based methods are provided.

5. **Chapter 6**: A summary of work in this dissertation is presented in the last Chapter. The insights, contributions and future work are also presented.

Chapter 2

Data Fusion and Classification in Remote Sensing

The multi-modalities of remotely sensed data provide more possibilities of information exploitation compared to a single data source. The fusion of multi-source data can in principal produce more detailed information to characterize the objects. Furthermore, fusion techniques can also help with removing redundant and noisy information in the data. By applying data fusion methodologies, the performance of remotely sensed data classification may be significantly improved.

2.1 Multi-source Data Fusion in Remote Sensing

In the remote sensing research field, an increasing quantity of data captured by airborne/satellite sensors has become available, including Very High Resolution (VHR) images, multi-temporal images, multi-band images, multi-polarization images, SAR images, LiDAR point clouds, and others. The large amount of multi-source data makes remote sensing data fusion more demanding and challenging. Data fusion has gained its popularity and usefulness in computer vision, machine intelligence, medical imaging and many other data processing areas. The objective of data fusion in remote sensing is to take advantage of multiple data sources to produce collaborative and complementary information for the interested scene than a single data source can provide. However, multi-source data fusion remains as a challenging task due to the existence of variations within different input data sets and the requirement for accurate data co-registration. Other than the data captured with different sensors, high-level features may be extracted from the original remote sensed data and combined into one or more feature maps that can be used as complementary data sources to the original data. For example, with recent remote sensors, the acquired multi-band images have very fine spatial resolution. The useful spatial feature information may be extracted as an independent data source that adds extra contextual information to the original spectral information. Therefore, the multi-sensor data and generated high-level feature data allow the possibility to exploit the remote sensing data by means of multi-source fusion strategies. According to the data sources, the fusion strategies for remotely sensed data are in the four fields: the multi-spectral image pan-sharpening, and spatial-spectral data fusion, LiDAR and image fusion, multi-temporal data analysis.

2.1.1 Remote sensing systems

Remote sensing aims at using aerial sensor technologies to detect and classify objects on Earth without being directly in contact with the target under investigation. Tremendous developments in the field of remote sensing have taken place in the past decades. Remote sensing is a remarkably broad subject including photographic imaging, nuclear magnetic resonance imaging, seismic tomography, multi-beam sonars, synthetic aperture radars, and etc. In this thesis proposal, we are particularly interested in the use of remote sensing data collected by airborne or spaceborne sensors for characterizing and classification of the Earth surface. Depending on the source of the energy involved in remote sensing data acquisition, two kinds of airborne imaging systems will be discussed, namely, passive and active systems. The outstanding representatives of the two systems are Hyperspectral imaging and LiDAR. We will briefly review the sensor advances in the field of hyperspectral imaging and LiDAR , respectively.

2.1.1.1 Hyperspectral imaging systems

Hyperspectral imaging, also termed imaging spectroscopy, has been increasingly used in various applications, such as food safety, pharmaceutical process monitoring and quality control, biometric, forensic, and etc. The hypserspectral sensors acquire a spectral vector with hundreds of elements from every pixel in a given scene and result in a HSI cube. The spatial and spectral characteristics of HSI are similar to photographic images and videos, so that many tools developed for those data can be directly extended for HSI analysis.

The acquisition of hyperspectral images can be obtained by airborne or spaceborne platforms. Table 2.1 displays spatial and spectral parameters of eight existing or proposed hyperspectral sensors: two airborne (HYDICE and AVIRIS) and six spaceborne (HYPERION, EnMAP, PRISMA, CHRIS6, HyspIRI and IASI) [12]. From the table, it can be observed that the spatial resolutions are higher for sensors carried by lower altitude platforms. The number of bands for the first six sensors is approximately 200 with a spectral resolution of the order of 10 nm, which offer a huge potential to discriminate materials. The last two sensors: CHRIS covers the visible bands and IASI covers the VNIR and the LWIR bands. Thereby, they offer the ability to estimate physical parameters such as temperature, moisture and etc.

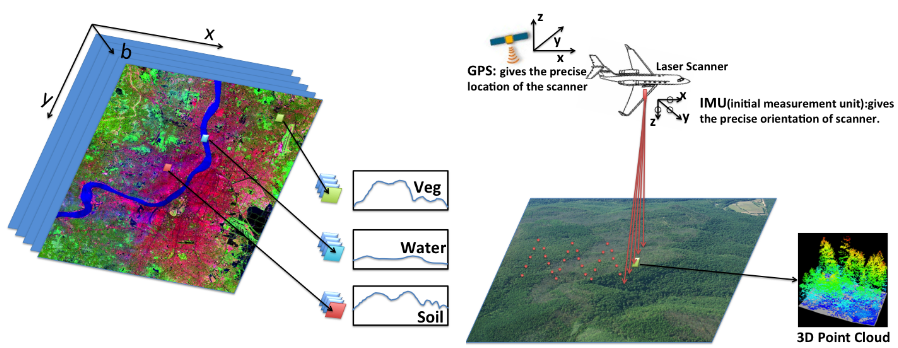


Figure 2.1: Example of Hyperspectral imaging and LiDAR imaging.

2.1.1.2 LiDAR systems

LiDAR sensing uses light in the form of a pulsed laser to measure ranges (variable distances) to the Earth. These light pulses, combined with other data collected by the airborne system, can generate precise, three-dimensional surface geometrical characteristics of the Earth. LiDAR systems for remote sensing are usually on airplanes and helicopters to be

Parameter	HY-DICE	AVIRIS	HYPER-ION	En-MAP	PRISMA	HyspIRI	CHRIS	IASI
Altitude (km)	1.6	20	705	653	614	626	556	817
Spatial resolution (m)	0.75	20	30	30	5-30	60	36	V:1-2 km; H:25km
Spectral resolution (nm)	7-14	10	10	6.5-10	10	4-12	1.3-12	0.5 cm^{-1}
Coverage (μm)	0.4-2.5	0.4-2.5	0.4-2.5	0.4-2.5	0.4-2.5	0.38-2.5 and 7.5-12	0.4-1.0	3.62-15.5(645-2760 cm^{-1})
Number of bands	210	224	220	228	238	217	63	8461
Data cube size	$200 \times 320 \times 210$	$512 \times 614 \times 224$	$660 \times 256 \times 220$	$1000 \times 1000 \times 228$	$400 \times 880 \times 238$	$620 \times 512 \times 210$	$748 \times 748 \times 63$	$765 \times 120 \times 8461$

Table 2.1: Parameters of eight Hyperspectral instruments.

used for acquiring data over broad areas. The common major components in a LiDAR system are Laser, scanner and optics, photodetector and receiver electronics, and a specialized GPS receiver. Two types of LiDAR are topographic and bathymetric. Airborne topographic LiDAR generally uses a 1064 nm near-infrared laser to map the land, while bathymetric LiDAR generally use 532 nm green light that penetrates water with much less attenuation, and measures seafloor and riverbed elevations.

In urban areas and forest study, the popular LiDAR sensors are those from the Optech ALTM-series, Leica ALS-series, RIEGL LMSseries, and the TopoSys Falcon series [13]. The basic measurement made by a LiDAR instrument is the distance between the sensor and a target surface by determining the elapsed time between the emission and arrival of the reflectedlaser pulse. Key differences among LiDAR sensors are related to the laser's wavelength, power, pulse duration and repetition rate, beam size and divergence angle, the specifics of the scanning mechanism, and the information recorded for each reflected pulse [14]. LiDAR data has been used in measurement of the three-dimensional structure of vegetation canopies, prediction of forest stand structure attributes, building footprint extraction, forest delineation, road detection, and etc.

2.1.2 Spatial-spectral data fusion

Besides detailed spectral information, HSI also contains much contextual information. For a given pixel, we can extract the shape, size and texture information of the structure to which it belongs. This information will complement the spectral signatures to help discriminate and identify different objects in a scene. Consequently, a joint spatial-spectral data fusion strategy is needed to generate more accurate results in HSI image segmentation and classification.

Several methods have been explored to add spatial information to improve the remote sensed data classification and clustering. One of the earliest proposed classifier that used both spectral and contextual information is known as the ECHO classifier [15]. It is a multi-stage classifier based on texture segmentation and statistical classification. It is hypothesized to work well where classes of interest are widely mixed with high variance. Later, Markov Random Field (MRF) was investigated in textural discrimination for remote sensing image segmentation [16, 17, 18]. An extensive literature is available on MRF modeling successfully applied in remote sensing image classification. In the MRF framework, the maximum a posteriori (MAP) decision rule is typically formulated as an iterative optimization step of the energy function, which is extremely time consuming with high resolution data. Furthermore, classical models used in MRF framework (e.g., Ising, Potts model) suffer from the high spatial resolution: neighboring pixels are highly correlated, while the standard neighbor system definition does not contain enough samples to be effective [19].

In addition, Rellier et al. [20] performed texture analysis using parameters employed by MRF to allow the characterization of different hyperspectral textures. Other than the MRF based texture analysis for hyperspectral image clustering and classification, various methods have been reported to extract the texture feature from the image region. Pixel shape index (PSI) [21] and the grey level co-occurence matrix (GLCM) [22] methods both start from a pixel in a given position then exploit the structural and shape information in outward directions in its neighborhood to complement the spectral feature space. Another model for extracting texture information, based on multiple 3D Gabor filters, captures the specific orientation, scale, and wavelength properties of hyperspectral data [23]. Similarly, wavelet analysis was also successfully utilized for hyperspectral image texture feature extraction [20, 24]. One of the most important aspects of texture description has been identified as the scale factor, and wavelet decomposition is able to generate a number of homogeneous features that represent the response of a bank of filters at different scales.

The other approach comprises multi-scale techniques as well as an adaptive neighborhood of a pixel according to the structures to which it belongs and was proposed by Benediktsson, et al [25]. They have exploited the morphological filters as an alternative way of performing joint spatial-spectral classification. Rather than defining a crisp neighbor set for each pixel, morphological filters enable the definition of an adaptive neighborhood of a pixel according to the size and shape of the structures to which it belongs. This adaptive neighborhood approach has shown its good performance for multispectral and hyperspectral data [26]. Recently, M. Dalla Mura et al. [27, 28] investigated the connected morphological operators for the analysis of very high resolution images, as well for HSI as an extension of the morphological profile based on a series of attribute filters. The attribute filters permit new possibilities for extracting morphological information in a way as it filters the spatial structures according to geometry (area, length, shape factors), texture (range, entropy), etc. Therefore, the input image can be processed according to different attributes, which can be defined with great flexibility.

2.1.3 LiDAR and image data fusion

The availability of LiDAR data has provided a new possibility for data classification and segmentation. Unlike optical and microwave sensors, LiDAR sensors directly measure both the vertical elevation and horizontal distribution of objects in a scene. The time interval between a laser pulse being emitted and retrieved can be measured by two approaches, namely, pulsed ranging (scanning LiDAR) where the travel time of a laser pulse from a sensor to a target is recorded, and continuous wave ranging (profiling LiDAR) where the phase change in a transmitted sinusoidal signal is converted into travel time [29].

The combination of optical imagery and LiDAR data has been an active research field in the remote sensing community. Images can provide detailed spectra for the discrimination of different materials in a study area, while LiDAR data can be exploited for characterizing topographical information of the scene. A single data source is difficult to provide reliable understanding of the scene. An image alone may not be able to differentiate objects of the same material, and LiDAR data alone provides little information for objects of similar geometrical structure with different material attributes. The fusion of LiDAR and imagery has been explored for a variety of applications such as DEM model generation, urban modeling, land cover classification, and etc.

The traditional approaches to generate DEM model are mostly based on stereo image matching. LiDAR sensors provide the capability of high-density 3D point data acquisition. The use of LiDAR data for DEM model generation becomes an effective alternative approach to traditional stereo matching based methods [29]. Another application of fusing LiDAR data and aerial images is 3D object detection and extraction, particularly for building footprints and roads in urban modeling. Rottensteiner et al. [30] suggested LiDAR and aerial imagery fused to improve the degree of automation and robustness for building extraction. Much research in building extraction and reconstruction using LiDAR and image fusion skills has shown to improve height accuracy and building outlines [31, 32, 33]. LiDAR data combined with imagery also works well for road extraction which is easily obscured by higher objects and difficult to accurately extract [34, 35].

The fusion of LiDAR and spectral image for land cover mapping has gained attention recently. The joint use of HSI and LiDAR data for the classification of forest areas, urban areas, and identification of trees from urban areas have been studied [3, 36, 37]. The previous works showed the incorporation of LiDAR source additionally improve the classification accuracy with extracted higher level features including curvatures and height information. Debes et al. [38] proposed a two-stream classification framework which takes advantage of both unsupervised segmentation and supervised classification using Random Forest classifier and combined with an object-level refinement to further enhance the accuracy for urban area data fusion. Liao et al. [38] adopted a graph-based framework to combine and embed spatial, spectral and LiDAR data in a lower dimensional manifold, followed by the classification with RBF-kernel based SVM.

2.1.4 Multi-temporal data analysis

In recent decade, there has been a significant increase in the interest in multi-temporal data analysis in the remote sensing community. The successful launching of the Sentinel-1 in 2014 and the launching of the coming satellites of the Copernicus program, result in great demand for development of multi-temporal data analysis.

Multi-temporal data has been used for change detection for decades. Change detection can be broadly characterized into two groups [39]: bi-temporal change detection and temporal trajectory analysis. The former category focuses on comparisons between two dates, while the latter one emphasizes more on discovering the trend of change by creating "profiles" of multi-temporal data.

Given the variations within time period, difficulties are encountered in accurate modeling and mapping of urban and forest areas. Hemissi et al. [40] proposed an advanced form of the temporal spectral signature defining the reflectance for each data point as a congregation of the spatio-spectra-temporal dimensions. Fusion of multi-temporal images facilitate the modeling of variations in spectral response due to time changes, and has proven to be effective in many applications [41, 42].

The multi-temporal Classification of remotely sensed image data takes advantage of efficient combination of different sources of information, namely, temporal, contextual, and multi-sensor to improve the results. Given an image labeled data, the problem of classifying another image of the same area obtained at a different time could be solved by passing the knowledge between the multi-temporal images. Many multi-temporal supervised methods have been developed, such as evidence reasoning [43], neural networks [44], and Bayes rule [45]. Also, since the intrinsic structures in these sequential images are similar, the spectral variations across a series of images can be efficiently reduced by grouping spectral neighbors within each image and mapping these clusters into a common space under a manifold alignment framework [46].

2.1.5 Multi-spectral image pan-sharpening

Multispectral (MS) imaging affords detailed spectral resolutions from visible to LWIR wavelength region with enhanced discriminating power for material identification and classification. One shortcoming of MS is the spatial resolution is usually coarser than

VHR panchromatic image. To overcome the limitations due to lower spatial resolution, fusion of MS data and high resolution panchromatic image can be adopted for enhanced performance. Due to the fact that the fusion of a MS with a high resolution panchromatic image is to improve the spatial resolution of it, we usually refer the process as pan-sharpening.

A panchromatic imaging detector has a single sensor which is sensitive to radiation within a wide spectral range, typically the visible part of the spectrum. Pan-sharpening is a pixel-level data fusion technique which fuses a PAN image and an HSI image to obtain one image with both high spatial and spectral resolutions. The pan-sharpening algorithms can be broadly categorized into three classes [47]: the component substitution (CS) fusion techniques, modulation-based fusion techniques and multi-resolution analysis (MRA)-based fusion techniques.

The CS method first up-samples the low-resolution MS image to have the same spatial size of the co-registered PAN image. Afterwards, with desired forward transform applied on the MS image, one component in the transformed image will be substituted by the PAN image which has higher spatial resolution. One classical CS pan-sharpening algorithm is the intensity-hue-saturation (IHS) transform [48] algorithm, in which the forward transformation of MS image is to obtain its components in the IHS space. The final pan-sharpened image is then generated by replacing the intensity component with the PAN image followed by an inverse IHS transform. Spectral distortion may occur in the fusion process, and various attempts have been made to minimize the spectral distortion in IHS pansharpening [49]. Other famous pan-sharpening algorithms, such as principal component analysis (PCA) transform fusion methods [50], Gram-Schmidt (GS) spectral sharpening algorithm [51], etc., still suffer from spectral preserving issues.

The modulation-based fusion methods are based on the idea that the spatial information is modulated into the MS images by the multiplication of the MS image with the ratio of the PAN image to the synthetic image. One typical modulation-based pan-sharpening algorithm includes Brovey transform (BT) algorithm [52], in which the synthetic image is the average of the blue, green and red bands.

The MRA-based fusion methods utilize multi-scale decomposition techniques such as Laplacian pyramids [53], multi-scale wavelets [54] and etc., to decompose MS and PAN images in different levels and apply a fusion rule onto the transform coefficients.

2.2 Classification of remotely sensed image data

Remote sensing data classification, particularly HSI data classification, has been a very active area of research in recent years. Given a set of observations (i.e., pixel vectors, LiDAR point clouds), the objective of classification is to find the hidden structure in unlabeled data with or without *a priori* knowledge. In remote sensing data classification, the objective is to group observations to represent land cover features. Examples of land cover could be forested, urban, agricultural and other types of features. The techniques used for classification could be categorized as unsupervised or supervised, pixel-level or object-level, etc. Unsupervised classification, which may also be called segmentation or clustering, is the partitioning of data into related regions. It is an important first step for other applications in data analysis and compression. Also, due to the scarcity of labeled data and large dimensionality of the input data, unsupervised segmentation of remotely sensed data is more demanding. Supervised classification takes full advantage of available training data to learn and discover the underlying structure for the unlabeled data, which usually produces better accuracy compared to unsupervised segmentation algorithms. For HSI data, traditional supervised and unsupervised classification are based on the observation of spectral feature vectors for each pixel. In other words, those traditional techniques are all pixel-based, which do not utilize the local spatial information that can only be extracted at object-level, i.e., texture, shape, size, and etc.

2.2.1 Unsupervised classification

Unsupervised classification, or segmentation, aims to extract regions by dividing pixels into disjoint sets of segments. The segments should have the property that regions are "homogeneous" within the segments and "heterogenous" between the clusters. As an initial processing step, segmentation usually facilitates easier analysis of higher level applciations, such as anomalous object detection, image retrieval, compression, and etc.

Two broad approaches to remote sensing data segmentation are threshold-based and pixel clustering. In grayscale and binary image segmentation, threshold-based approach is useful in discriminating foreground from the background. By selecting an appropriate threshold value, the gray level image can be converted to binary image. Similarly, remote sensing data, such as HSI, may be segmented into non-overlapping regions via threshold-based method. In [55], a manual multithresholding (MT) approach was proposed to segment HSI into different regions. They first extract spectral index (SI) image from HSI as a way to capture one or more spectral characteristics. Afterwards, the SI image is segmented into clusters by simply slicing the SI value into intervals of equal width. If a spectrum's SI value lies in the first interval, it is classified to region 1; if it lies in the second interval, it is classified to region 2; and so on. The segmentation problem becomes the selection of proper value for the thresholds. The threshold selection techniques can further be categorized into local/global, simple/adaptive methods. Global threshold selection finds one simple threshold value for the entire image data set, whereas local threshold selection usually adaptively calculates the threshold for different pixels [56]. However, threshold-based techniques, although computationally less expensive, did not get much attention. Threshold-based techniques inherently work well with grayscale image while the remote sensed imagery of interest here usually contains multiple spectral bands. Other than that, the multi-dimensional spectral vector associated with each pixel naturally makes each of the pixel an individual feature instance and suitable for pixel-based clustering.

Pixel-based clustering groups pixels into homogeneous regions by clustering their feature vectors. The multi-band nature makes pixel-based clustering the natural choice for segmentation. The simplest and most commonly used methods for remote sensing data are k-means and ISODATA algorithms. Both of them are based on statistical modeling of the data to iteratively find the optimum partition of the data. In general, these two methods start by randomly picking several data points as the cluster center candidates and assigning each pixel to the closest candidate. Then statistical parameters are re-calculated and new cluster centers are updated based the new clusters. This process iterates until variations are below a certain threshold. Other statistical based algorithms applied to HSI data include the Gaussian mixture models [57].

Other than statistical based clustering algorithms, spectral clustering (SC) has been recently investigated for HSI pixel clustering. Compared to traditional clustering algorithms, SC has some obvious advantages. It is highly related to manifold learning which studies the underlying structure by modeling the pairwise relationship between pixels. Therefore, it facilitates the recognition of clusters of unusual shapes. On one hand, SC serves as a graph-based embedding which implies a clustering condition. On the other hand, SC seeks to find a clustering condition that can capture flat, elongated and even curved nonlinear data clusters.

2.2.2 Supervised classification

Supervised classification approaches first learn from an available training set to understand the statistical or geometrical characteristics of the clusters that exist in the data set. Once trained, the classifier is used to assign labels to unlabeled data points according to the learned knowledge of the data set. Traditional supervised classification methods for remote sensing data are pixel based on the spectrum of each pixel itself. Spatial information can be fused to the classification process by separately extracting spatial contents from neighboring pixels to be combined with spectral knowledge. The object-based classification methods usually work in the same way as a pixel-based classification, with the difference that we utilize image segmentation results as the input to the classifier. Therefore, object-based classification methods inherently contain higher-level information, such as statistical parameters, texture, shape, and etc., that can only be characterized at the object level instead of from individual pixels.

2.2.2.1 Pixel-level classification

For remote sensed data, especially HSI, the multi-dimensional spectra naturally makes each of the pixels an individual feature instance. In other words, pixel-level classification of an image is to categorize single pixels into land cover classes with only the spectral information for each pixel utilized as its feature vector. With a lack of neighboring information to each pixel, there usually exists a ″pepper and salt″ phenomenon in HSI classification map.

One of the earliest and widely used supervised classification algorithm for remote sensing data is the Gaussian Maximum Likelihood (GML) classifier, which is based on Bayesian probability theory and has become a widely accepted statistical model for remotely sensed data. It first learns and establishes a multi-dimensional histogram for each class from the training data, and assigns unlabeled data to classes based on a posteriori probability. The main problem involved in the probabilistic model is that the limited availability of training data may result in overfitting. The other problem is the underlying data distributions of the classes do not necessarily satisfiy a normal distribution.

Artificial neural network (ANN) classifier is one of the most widely used nonparametric classification algorithm in pattern recognition field. It usually consists of one input layer, at least one hidden layer and one output layer. The ANN may be able to handle non-normal, complex feature spaces and multivariate data types. In the literature, it also has been found that the difference between the performances of other classifiers and ANN increase in favor of ANN with the increasing number of channels. One important property of ANN classifier is, with a sophisticated paradigm, it can handle data with a small training set. However, with high-dimensional input data, the complexity dramatically increases in ANN classifier. Therefore, dimensionality reduction is frequently applied to high dimensional data to achieve tolerable training time.

Support vector machines (SVM) have been investigated intensively as an alternative approach to the usual statistical and neural classifiers for high-dimensional images. The SVM classifier is based on the concept of a decision plane which identifies the boundary lines that separate a set of objects with different class labels. In order to handle nonlinear boundaries, "kernel tricks" are widely adopted for SVM classifier to project the original data into a higher dimensional implicit feature space where classes are linearly separable. SVM appears to be advantageous compared to other supervised classifiers in the presence of heterogeneous classes with only small training set, and it is less sensitive to the curse of dimensionality. The other advantage of the SVM classifier is that it requires little effort for architecture design since it involves few control parameters. However, SVM was originally proposed to solve two-class problem. To effectively solve multi-class classification problem with SVM classifiers is still an ongoing research issue.

The Random forest classifier, as one of the ensemble methods, combines the predictions of several decision tree classifiers in order to improve generalizability over a single estimator. A decision tree classifier is advantageous to pixel-level remote sensed data classification because of its flexibility, intuitive simplicity and computational efficiency. It recursively partitions a data set into smaller subsets according to different criteria at each node in a tree structure. The classification procedure is strictly non-parametric and does not make assumptions of the data distribution. In random forest classifier, each decision tree in the ensemble is built from a sample of data points drawn with replacement from the training set. In the final step, every pixel is classified based upon a majority vote from all the tree predictors used in the forest to obtain its final label.

2.2.2.2 Object-level classification

In object-leveled classification, the input processing units are no longer single pixels but image objects. In order to apply object-level classification, first the remote sensed data points needs to be grouped into meaningful clusters based on their individual features. Secondly, the clusters need to be effectively represented and used as the input instances to the classifiers with different region-level classification rules, such as spectral, spatial, contextual and textual information. Thereby, the obvious advantages of object-level classification to the conventional pixel-level classification is the incorporation of additional information, namely, texture, shape, size, and etc., which can only be extracted at object-level. The other advantage of object-level classification is the spectral properties of individual pixel are averaged for the object or structure it belongs to. Because spectral mixing increases in a remotely sensed image that has coarse spatial resolution, this may cause confusion in the classification. By extracting and averaging the spectral information at object-level, the within-object variability can be reduced.

In general, object-level classification includes an unsupervised multi-resolution segmentation and a knowledge-based classification. The classification can usually adopt the general classifiers as mentioned above. The multi-resolution segmentation starts from treating each pixel as a separate object and gradually combining them into regions. One commonly used technique for multi-resolution segmentation is bottom-up merging which iteratively groups adjacent and similar pixels into meaningful objects. Subsequently, adjacent and similar image objects are merged to form bigger ones based on object-level similarity measure until stopping criterion is met.

Recently, geographic object-based image analysis (GEOBIA) has become a new discipline for the analysis of remote sensing data. This is a sub-discipline of Geographic Information Science (GIScience) devoted to developing automated methods to partition remote sensing imagery into meaningful image-objects, and assessing their characteristics through spatial, spectral and temporal scales, so as to generate new geographic information in GIS-ready format [58]. In this new discipline, many theories, methods and tools have been developed to replicate human interpretation of remote sensed images in automated/semi-automated ways. Overall, object-level analysis of remote sensed images offers new possibilities for situations where spectral properties are not unique, but where shape or neighborhood relations are distinct. Also, object-level processing is closely related to human cognitive perception of the scene compared to pixel-wise understanding of the image.

2.3 Summary

This Chapter introduced the data fusion strategies for remotely sensed data and addressed the classification problem in remote sensing area. The development in remote sensing tools and technologies provide more possibilities in data fusion. The high spatial resolution in hyperspectral imaging enables spatial feature mining, so that features from spatial domain be exploited simultaneously with the detailed spectral information. LiDAR sensors collect 3D point clouds that can be used to extract elevation and curvature information to further provide separabilities between different structures. Overall, the remote sensing data fusion aims to integrate multi-source information generated from different perspectives in order to produce fused information that contain more enriched information than a single source can provide.

Classification has been a popular research topic in remote sensing area for a long time. In remote sensing data classification, the objective is to group observations to represent different land cover types, such as vegetation, building, water, soil and etc. The fusion of multiple source of data and multi-domain features can help with more accurate image interpretation and improve the classification performance. The techniques used for classification of remotely sensed data could be categorized as unsupervised or supervised, pixel-level or object-level, etc. Unsupervised classification is to group data points into meaningful and disjoint sets of segments without any a priori knowledge. On the other hand, supervised classification first learns from an available set of known data points to understand the statistical or geometrical characteristics of land types that exist in the data set. Based on the type of processing unit, classification can also be categorized as pixel-based or object-based classification. Usually, a unsupervised classification scheme is applied on the input dataset first to obtain small and homogeneous clusters as the smallest processing unit that will be sent to classifiers for supervised learning.

Chapter 3

Graph Modeling of Remotely Sensed Data

3.1 Graph theory

Graph theory has been widely used in data modeling and analysis for its strong mathematical foundation. Particularly, graph theory has found many applications in image processing and analysis due to the suitability of using graphs to represent discrete data and model pairwise relationship. The history of graph theory can be traced to 1736, when the Swiss mathematician Leonhard Euler published the first paper regarding the Knigsberg bridge problem. The term graph refers to a set of vertices and edges that connect the vertices. As a branch of mathematics, many primers on the mathematics of graph theory may be found in the literature [59, 60]. In this chapter, we will briefly introduce the basic concepts and notations for spectral graph theory and connect the concepts to image processing and analysis.

3.1.1 Basic mathematical foundations

3.1.1.1 Basic terminologies

Intuitively, a graph is composed of a set of elements and a set of pairwise relationships between them. The elements are called nodes/vertices, and the relationships are represented by edges. Formally, we denote the vertex set and edge set as V and E, respectively. A graph G is therefore defined by the sets $G = (V, E)$ in which $E \subseteq V \times V$ since each edge is a subset of two vertices. The edge set E contains unordered pairings of vertices such that if two vertices v_i and v_j are endpoints of an edge, then $\{v_i, v_j\} \in E$, and these two vertices are called adjacent to each other. A graph G may be viewed as weighted graph by assigning a value to the weight of an edge incident to two vertices as $w(v_i, v_j)$ or w_{ij}. Intuitively, an edge weight of zero means no edge connecting two vertices. Additionally, one edge may be considered to be oriented/directed if $w_{ij} \neq w_{ji}$. Here, we only consider undirected graphs. Given these preliminaries, we may proceed to the representations of graphs in data modeling and analysis.

3.1.1.2 Graph representation

Different data structures used for the representation of graphs have different impacts on the size of data, computational speed and ease of use. Generally, there are two type of commonly used graph representations: matrix representation and list representation.

A matrix representation, particularly sparse matrix, may provide efficient storage for image data and is also very convenient to work with. There are two matrix representations for a graph. An incidence matrix is a two-dimensional Boolean matrix, in which the rows represent the vertices set, the columns represent the edges set, and the entries indicate if the vertex at a row is incident to the edge at a column. However, it may preserve the edge orientation information but not the edge weight. The other matrix, named adjacency matrix (or affinity matrix/weight matrix), is a two-dimensional square matrix, in which both the rows and columns represent vertices, and the entries not only indicate if the vertex at a row is incident to the vertex at a column, but also the weight associated with the edge in between, written as:

$$[W]_{ij} = \begin{cases} \omega_{ij}, & if\ \{v_i, v_j\} \in E \\ 0, & otherwise \end{cases}$$

Each vertex has a certain degree obtained from the neighborhood, denoted as $\Gamma(v_i)$, which is defined as the total edges from the set of all adjacent vertices to it: $deg(v_i) =$

$\sum_{v_j \in \Gamma(v_i)} w_{ij}$. Thereby, a degree matrix D can be obtained from the affinity matrix W as a diagonal matrix whose entries are the degree for every node $deg(v_i)$ at $i = j$, and zero elsewhere. The Laplacian matrix of an undirected graph is defined as the difference of the diagonal degree matrix and adjacency matrix $L = D - W$. The analysis of its eigenvalues and eigenvectors can provide many of the graph's structural properties. As one of the most important metrics in graph theory, we refer readers to [61] for an overview of its properties.

Adjacency list is another graph representation which could be less memory consuming. For each vertex, it only stores a list of adjacent vertices and edge weights. However, searching for the entries may be less efficient compared to adjacency matrix representation.

Here, we will represent the data using an adjacency matrix approach in our study.

3.1.2 Graph construction

For image processing, the vertex set in a graph G is the pixel set. In order to create a weighted graph representation for a given image, we need to first build edges that connect vertices based on certain criteria, as well as assign weights to edges that characterize the neighborhood and similarities of the endpoints. Given an edge, the weight is computed by comparing their features to reflect the similarity of the endpoints. A Gaussian kernel function, $w_{ij} = exp(-\frac{|x_i - x_j|^2}{\sigma^2})$, where x_i, x_j are the feature vectors, and σ is a kernel parameter; is most commonly adopted for the similarity measure of its two endpoints due to its simplicity and effectiveness. However, the scaling parameter σ needs to be specified manually and is data-dependent.

In the spatial domain, pixels in an imagery are considered as organized data lying on a two-dimensional grid. Given one pixel (x_i^n, x_j^n), several distances can be considered to find its neighbors on the image grid structure. City-block distance $\mu(v_i, v_j) = |x_i^1 - x_j^1| + |x_i^2 - x_j^2|$ and the Chebyshev distance: $\mu(v_i, v_j) = max(|x_i^1 - x_j^1|, |x_i^2 - x_j^2|)$ are usually adopted for 2D image pixels. Given those distances, a τ-neighborhood graph can be defined as $\mathcal{N}_\tau(v_i) = \{v_j \in \mathcal{V} \setminus v_i : \mu(v_i, v_j) \leq \tau\}$. For example, a 4-adjacency and 8-adjacency grid graph can be obtained with the city-block and the Chebyshev distance, respectively, with $\tau \leq 1$.

In the spectral domain, where pixels of HSI lie in a high dimensional space as a set of unorganized data, the typical choice of the distance $\mu(v_i, v_j)$ is the Euclidean distance. Edge construction techniques include τ-neighborhood graph, k-nearest neighborhood graph (*knn*) and modified k-nearest-neighborhood graph. In pattern recognition, k-nearest neighbor is a commonly used supervised classification technique which assigns an unlabeled object to the labeled data in its k-nearest neighbors via majority vote. In graph construction, two vertices v_i and v_j are connected if one of them is among the k nearest neighbors of the other one or both. For the latter case, *knn* may also be called mutual-*knn*. The constraint of mutuality results in a tendency to connect data points of similar density but leaving regions with different density unconnected. However, the number k is data-driven, and is usually arbitrarily and manually chosen. Therefore, there are other variations to k-nearest-neighborhood graph, such as adaptive k-nearest-neighborhood graph (*ANN*). Edges in those graphs are created by automatically estimating the number of k for each data point based on its local density. One objective of applying adaptive rules to connect vertices in a graph for HSI or other remote sensed data is to provide more separability among data points in different classes while to maintain a high connectivity within each class. Much work has been done in the literature and we will discuss it in Chapter 4.

3.2 Topology-based classification for HSI

3.2.1 Topological Anomaly Detection (TAD)

TAD [62] is an algorithm proposed to detect anomalies in a spectral image by identifying background components. By comparing every pixel to the background model, a map of TAD scores, where a pixel with higher score indicates an increased anomalousness, could be produced as a reference map for target or anomaly detection. Inspired by the TAD algorithm, a topology-based classification scheme was proposed by extracting the largest background components from TAD algorithm as the regions of interest (ROI) to be used together with supervised classifiers for spectral image classification. This scheme has practical value for its nonparametric nature and easy implementation.

In order to analyze the topological structure of a given spectral image, a simple graph has to be created to model the relationships between the data points. Considering computational complexity, only a random subsample S of all pixels is chosen for modeling the background. Once the subsample S is selected, pairwise relationship is constructed by the measurement of Euclidean distance. An edge is created if the Euclidean distance is below a threshold, and only 10% closest pair of vertices are kept indicating their strongest similarities to each other. When the graph is created, all connected components are identified and the largest components, those containing more than 2% of the pixels in the subsample, are designated as the background model. To rank each pixel against the background model, the data mining notion of codensity is applied. The kth codensity at a vertex in a graph is the distance to its kth nearest neighbor, or the radius of the smallest sphere enclosing k neighbors. The TAD rank that characterizes the dissimilarity to the background model is defined as the sum of the distances to its 3^{rd},4^{th}, and 5^{th} nearest neighbors in the modeled background. This TAD score ranking scheme allows those vertices near background components with low density to have higher scores than vertices near background components with high density.

3.2.2 TAD-based semi-supervised classification

To generate accurate land cover maps usually requires the use of supervised classification method, which needs human generated training data. All the supervised classifications usually follow a sequence of operations: 1. Defining the Training Sites. 2. Extraction of Signatures. 3. Classification of the Image. Usually, the more training sites and the larger the training sets are, the better classification results can be achieved. In other words, the quality of a supervised classification may depend on the quality of the training data. However, the first two steps may be infeasible due to the cost associated with the acquisition of labeled data. Instead, an alternative approach, purely unsupervised training information extraction based on the TAD algorithm is proposed. The automatic generation of training set allows the utilization of classification algorithms in an unsupervised fashion.

Automatic class feature extraction

In order to capture sufficient information to model every class in the scene, a number of tiles of size $l \times l$ are selected. Compared to randomly selected pixels, a tile as a unit contains more local variance than a single pixel. Based on the pixels from selected tiles, a graph is constructed using mutual k nearest neighbor (mutual k-NN) method with the advanced triangle inequality algorithm (ATRIA).

ATRIA requires the user to input the desired number of nearest neighbors k. If k is set too high, more pixels will be linked together which may result in not enough connected components produced to accurately model the training data. However, this problem may be solved by keeping only the closest edges. Originally, TAD identifies the 10% closest edges w calculated on a fully connected graph and uses the strongly connected pixels as the background model. However, mutual k-NN graph produces fewer edges than a fully connected graph, we only keep p percent closeted edges in the mutual k-NN graph. If the value of k is set too high, p can be lowered to discard more edges and keep only the closest edges. In this way, we may break down a relatively connected graph into multiple connected components. In the opposite, if k is set too low, it may cause an overproduction of connected components with relatively smaller size. In order to represent each identified component as a class, the size of a single component is screened out if it is below the threshold t, defined as q percent of the number of pixels in the background components. After conducting multiple experiments on various hyperspectral data sets, we find set k to 4 usually produces stable results (in most cases it gives 3-12 connected components to be modeled as the training data set later). As p and q are more in a ”case by case” fashion, the user could easily modify them to produce better results.

Accuracy evaluation

Given labeled data, we can compare the automatically generated ground truth components to evaluate the classification accuracy. From the statistical point of view, the shape of the distribution for a ground truth class in the spectral domain could be multimodal. The shape of a distribution may be more vividly characterized by the behavior of the ”tails”. One land cover distribution may have one or more long ”tails”, and each ”tail” has high-density points thus easily identified as multiple connected components by TAD. This spectral variation within one land cover class requires us to identify and combine subclasses into a single land cover type. This could be done either in the training stage or after the class map is obtained. Actually, subclasses produced by TAD may aid in the training process for supervised classification. The use of subclasses instead of a combined set as the training data will allow the classifier to establish better local decision boundaries and achieve higher accuracy. Therefore, we first input the ground truth components identified by TAD into classifier, such as Gaussian Maximum Likelihood (GML) and Minimum Distance to the Mean (MDM), to generate an over-classified map and merge the subclasses afterwards.

For each class, we represent it as a multi-dimensional Gaussian distribution and compare it to the labeled data to find the best match. Those classes that are mapped to the same labeled data will be merged. The way we compare two distributions is using Bhattacharyya distance:

$$B = 1/8[\mu_1 - \mu_2]^t \Sigma^{-1} [\mu_1 - \mu_2] + 1/2 \, ln(\frac{det\Sigma}{\sqrt{det\Sigma_1 + det\Sigma_2}})$$

where μ_i, Σ_i are the mean and covariance of the distribution for class i, $\Sigma = (\Sigma_1 + \Sigma_2)/2$.

Two hyperspectral images were used to test the semi-supervised classification scheme. One was collected at Cooke City, Montana on July 4, 2006 by the HyMap sensor, operated by HyVista Corporation. Each pixel has approximately 3 meters of ground resolution. The other image is the famous University of Pavia, which was collected by the Reflective Optics System Imaging Spectrometer (ROSIS). It has 115 spectral bands ranging from 0.43 to 0.86 μm with 1.3 m spatial resolution. Before running the TAD, first 15 PCA bands were selected as the input data instead of original image to remove redundancy and reduce computational complexity. After running TAD algorithm, eight connected components and eleven connected components were identified as the training set for Cooke City and Pavia University, respectively. To evaluate the accuracy, we measure the similarity of each obtained subclass to the reference labeled data to determine the labels for each subclass. The Bhattacharyya distance between every pair of subclass and labeled data are listed in Table 3.1.

In this table, each row corresponds to labeled data and each column corresponds to a subclass produced by TAD+ GML/MDM method. Ideally, every labeled data should find its matched subclasses by identifying the smallest Bhattacharyya distances in each row. However, due to the high similarity among different land covers, it usually ends up with some subclasses or labeled data that do not have matched pair. To avoid the appearance

(a) B-Distance table for Cooke City Image

ROIs	1	2	3	4	5	6	7	8
Buildings	11.97	7.60	11.01	2.41	7.19	10.04	13.47	17.6
Trees	5.06	5.99	5.56	11.42	5.62	1.22	2.12	6.03
Grass	3.85	3.79	9.10	13.75	4.88	4.36	21.18	41.55
Road	7.80	6.17	10.11	8.80	2.43	7.92	18.13	39.92
Grey	3.55	5.50	2.74	11.15	8.68	5.88	16.75	27.88

(b) B-Distance table for Pavia University Image

ROIs	1	2	3	4	5	6	7	8	9	10	11
Roads	10.68	0.96	8.91	4.38	0.68	1.12	3.25	3.90	9.74	1.86	3.26
Meadows	4.43	9.05	3.52	1.32	5.26	4.54	4.48	2.50	8.92	6.06	5.59
Gravel	14.83	1.79	20.71	5.89	2.90	3.41	5.38	5.29	10.02	2.09	5.01
Trees	10.57	10.15	7.22	2.16	7.54	8.96	2.41	0.69	14.55	9.40	7.49
metalSheets	37.54	9.15	21.02	11.63	9.80	15.20	8.40	10.16	54.59	11.12	4.28
bareSoil	4.03	4.80	2.63	1.63	3.62	3.03	4.09	3.16	3.27	2.78	3.53
bitumen	21.35	1.91	25.00	7.91	2.53	2.67	5.64	6.57	34.71	3.94	5.78
bricks	9.85	1.52	12.00	5.18	2.66	2.59	5.04	4.75	7.31	1.76	4.37
shadow	81.81	3.80	30.12	9.62	5.71	12.95	4.64	4.81	80.53	7.65	6.50

Table 3.1: Bhattacharyya distance between every pair of classes generated using TAD with labeled data.

of unmatched class, we applied two schemes:

Scheme a Step 1. Instead of labeling each column one by one, we first match the two classes in accordance to the smallest entry in this table. Step 2. Delete the matched column and row in step 1, then repeat step 1 until all ground truth classes are matched. Step 3. find the unlabeled subclasses and match them to the ground truth class with respect to the smallest Bhattacharyya distance.

Scheme b Step 1. For each row in the table, we reorder the subclasses in accordance with the Bhattacharyya distance in ascending order. However, the first column stores the minimum Bhattacharyya distance for every ground truth class. Hence we match the subclasses in the first column to its associated ground truth class. Note that if one subclass is listed more than once in this column, it will only be labeled with the ground truth class corresponds to the smallest Bhattacharyya distance. Step 2. Delete the rows of the labeled ground truth classes, and move to the second column and repeat from first step until all ground truth classes are matched. Step 3. find the unlabeled subclasses and match them to the ground truth class according to the smallest Bhattacharyya distance. Those two match and label processes could both produce a fully mapped classification image with the same number of classes as in the land cover ROIs. We will apply the two algorithms and then obtain two final classification maps, but only keep the one which gives better classification accuracy.

Figure 3.1 shows the classification map for Cooke City and Pavia University using TAD based approach. A comparison of overall accuracy (OA) associated with the semi-supervised classification algorithm proposed in this paper and traditional unsupervised classification algorithm K-mean is presented in Figure 3.4. It is obvious that the TAD + GML/MDM scheme proposed in this paper has much higher overall accuracy than traditional unsupervised classifiers K-mean. The performance of this semi-supervised classification scheme may be further improved by adding up more pixels into the background components identified by the TAD algorithm. TAD tends to line up pixels that are spectrally related, therefore if the distributions of two classes are separable then TAD more easily separates them into two background components. Otherwise, TAD might divide them apart at wrong interface or combine them instead.

3.3 Manifold learning for HSI data analysis

An HSI image contains detailed spectral information which results in high dimensionality and highly correlated spectral content. As a graph based approach, manifold learning is widely applied to HSI data to represent the topology of the high-dimensional nonlinear data in a lower dimensional space while seeking to preserve the original data intrinsic structure.

Manifold learning can be categorized according to different criteria, such as, linear and nonlinear, unsupervised and supervised, local, global or iterative, and etc. Linear approaches [63] can be easily used in an unsupervised fashion, or applied on out-of-sample data with an explicit transformation matrix. Nonlinear approaches can represent

Figure 3.1: Classification Maps for (a) Cooke City and (b) Pavia University.

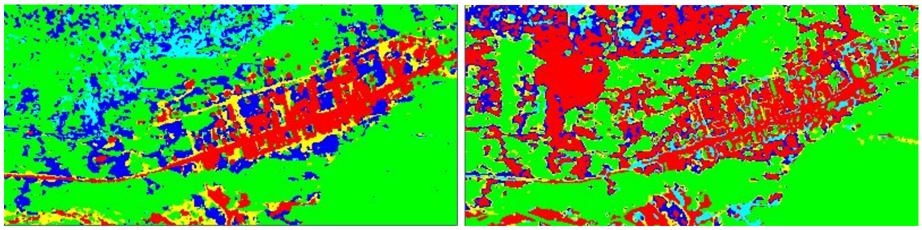


Figure 3.2: Cooke City Classification Map. Left: using TAD+GML; OA = 84.5%. Right: using TAD+MDM; OA = 52.5%.

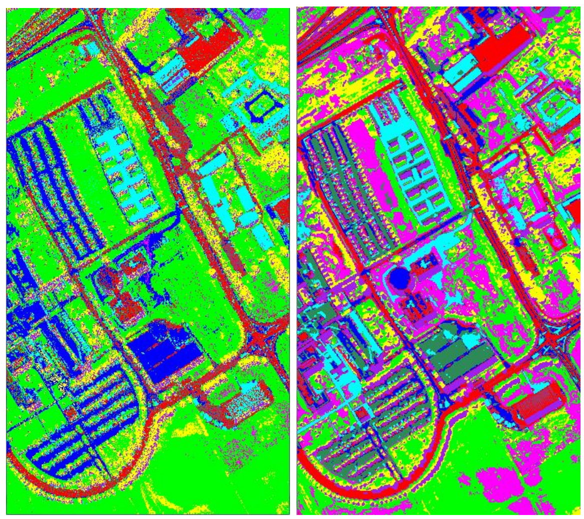


Figure 3.3: Pavia University Classification Map. Left: using TAD+GML; OA = 62.9%. Right: using TAD+MDM; OA = 56.7%.

the nonlinearities of real world data but do not have an explicit mapping matrix. It thereby can not be directly used on out-of-sample data. Kernel-based approaches [64] provide the possibility of solving the non-linear problem in a linear fashion by mapping data into a higher dimensional space without necessarily providing explicit projection transformation. A tensor based manifold learning approach [65], as an extension of conventional supervised manifold learning, adopts multi-linear algebra instead of linear

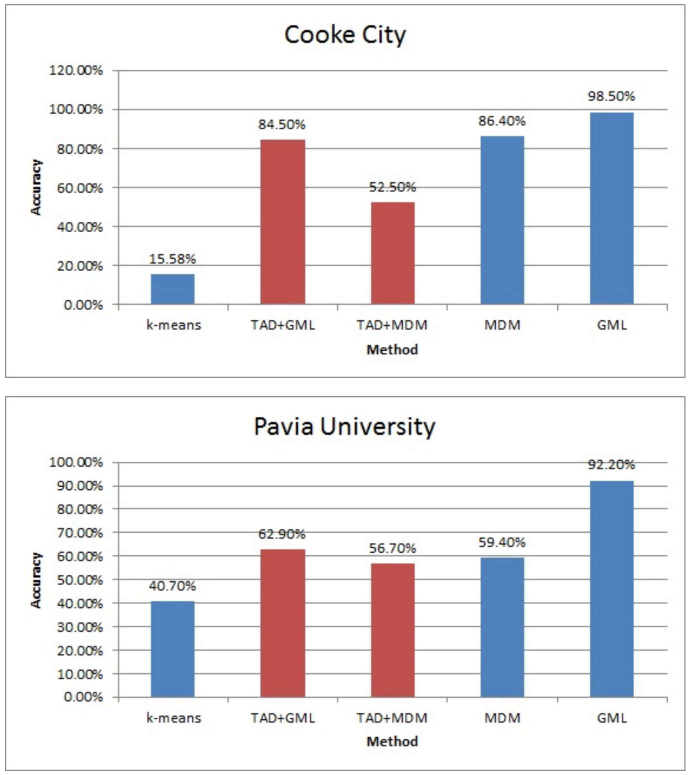


Figure 3.4: Comparison of the OA of traditional unsupervised, supervised classifiers and the semi-supervised classifier presented in this paper.

algebra to provide more intuitive modeling of HSI data, and simultaneously incorporate spatial information. Global manifold learning [66] aims at preserving the structure of the whole dataset which causes greater computational overhead. Local manifold learning [67] exploits techniques for preserving the local geometry of the data which may reduce the computational complexity. Iterative manifold learning [68] obtains manifold embedding by iterative optimization which helps reducing the crowding or overlapping of class boundaries.

3.3.1 Graph embedding framework

Consider a set of n data samples $\mathbf{X} = [\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}], \mathbf{x_i} \in \mathbf{R^m}$, where m is the input data dimension. Graph-based feature extraction is one of the popular dimensionality reduction approach that seeks to find a set of manifold coordinates $\mathbf{Y} = [\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_n}], \mathbf{y_i} \in \mathbf{R^p}$ where $p \ll m$, through a feature mapping $\mathbf{\Phi} : \mathbf{x} \rightarrow \mathbf{y}$ which can be linear, nonlinear, supervised, unsupervised, kernel-based, tensor-based, etc. Yan, et al. [69] proposed a general framework, named graph embedding, to unify multiple kinds of dimensionality reduction algorithms (feature extraction) within a common framework with constraints from scale normalization or a penalty graph that characterizes a statistical or geometric property that should be avoided.

The graph embedding framework aims at identifying relationships among the vertices of G that best characterize the similarities. Under graph embedding framework, the one-dimensional case, where the resultant manifold coordinates are $Y = [y_1, y_2, ..., y_n]$, could be obtained by solving for the objective function

$$y^* = arg \min_{\mathbf{yBy^T=r}} \sum_{i \neq j} ||y_i - y_j||^2 \mathbf{W_{ij}} = arg \min_{\mathbf{yBy^T=r}} \mathbf{yLy^T}$$

where r is a constant and B is a constraint matrix, which is defined to avoid a trivial solution of the objective function. The definition of r and B both depends on the specific approach. The objective is to have resultant coordinates y_i, y_j closer to each other if they have larger weights w_{ij}. The solution is an optimization problem which can be easily obtained via solving the eigen decomposition problem $\mathbf{L}y = \lambda \mathbf{B}y$, where in this one-dimensional case, the resultant coordinate y_i is the i-th element in the eigenvector with the smallest non-zero eigenvalue.

Isometric Feature Mapping (ISOMAP) is a widely used global nonlinear manifold learning algorithm which can be represented in the graph embedding framework. It assumes the local feature space formed by the nearest neighbors is linear, and the global nonlinear transformation can be derived by linking each individual linear space. Given the shortest path distance matrix S_{stp} [70], the Laplacian matrix is defined as $\mathbf{L} = \mathbf{HTH}/2$, where $\mathbf{H} = \mathbf{I} - (1/n)\mathbf{ee^T}$, $T_{ij} = [(S_{stp})_{ij}]^2$. The matrix $\mathbf{W}$ is defined as $-L_{ij}, i \neq j; else\ 0$. The constraint matrix is defined as the identity matrix, $\mathbf{B} = \mathbf{I}$.

Laplacian Eigenmaps (LE) is a widely used local nonlinear manifold learning method that can also be represented under the graph embedding framework. The adjacency matrix $\mathbf{W}$ encodes neighborhood information for each vertex, the Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$ and the constraint matrix $\mathbf{B} = \mathbf{D}$. Thereby, under the graph embedding framework, the resultant manifold coordinates are derived by solving for $\mathbf{L}y = \lambda \mathbf{D}y$, which is also equivalent to the function of the normalized cut algorithm [71].

The graph embedding framework provides a unified perspective for the graph-based feature extraction algorithms. For other supervised/linear/kernel/tensor extensions to the general graph embedding framework, we refer readers to [69] for detailed descriptions.

Overall, the objective of graph embedding is to represent each vertex of a graph in a low-dimensional subspace that preserves the proximity among neighboring vertices in the original space. The local proximity is measured by a graph proximity matrix which contains information such as statistical or geometric properties of the data. In linear subspace learning, graph embedding derives new vector representations based on the eigen analysis of the graph Laplacian. The low-dimensional vector representations are usually the eigenvectors corresponding to the first eigenvalues of the graph Laplacian matrix with certain constraints. Algorithms such as PCA, LDA, LPP, ISOMAP, LLE, LE, and tensor based algorithms, can all be generalized as manifold learning based graph embedding. Graph embedding is a very useful computational tool for HSI data analysis. It finds a graph-based data transformation so that the alternative representations may redistribute the original data points into a more useful form, where objects become more separable compared to in the original high dimensional manifolds.

3.3.2 Manifold learning in dimensionality reduction

Dimensionality reduction (DR), a popular machine learning and statistical pattern recognition topic, studies how to extract low dimensional structure dimensionality from high dimensional data. Many algorithms for dimensionality reduction have been developed to accomplish these tasks. Principal components analysis (PCA) is one of the classical methods that derives a set of linear approximations to a higher dimensional input observation. However, PCA captures the maximum variability in the data but is limited by its global linearity. The other drawback of PCA is that it only characterize linear subspaces. Other dimensionality reduction algorithms such as Multidimensional Scaling (MDS), factor analysis, Independent Component Analysis (ICA), etc., all suffer from the same drawback. However, the basic geometric intuitions behind PCA still play a very important role in many algorithms for nonlinear dimensionality reduction.

Graph embedding and manifold learning based feature extraction (FE) are two highly correlated topics. One important contribution of graph embedding is to provide a general platform for efficient FE/DR. Manifold learning techniques can be used in producing a compact low-dimensional encoding of a given high-dimensional input data set. Also, it may provide an concise interpretation of high dimensional input data set in terms of intrinsic degree of freedom, as a by-product of dimensionality reduction. ISOMAP and Kernel-PCA are two widely used global manifold learning based FE approaches. Local methods including LLE, LTSA, LE, etc. use a cost function that retains local properties of the data and have shown good performance for FE in the literature.

Other than feature extraction, feature selection (FS) is another category in dimensionality reduction. Instead of finding a transformation of the original data set into a feature space, FS tries to find a subset of the original variables. FS methods can be classified into wrapper, filter, embedding and hybrid methods [4]. The wrapper methods requires a learning algorithm to evaluate the features, however wrapper models suffer from high time and space complexities due to exhaustive searching for the optimal features [4]. The filter model examines intrinsic properties of the data to evaluate the features prior to the learning tasks and relies on the class labels by assessing the correlations between features and the class label [72]. The embedding model determines the best feature subset using a classifier where variable selection is a part of the training process. In HSI data processing, FS usually means spectral band selection. In [73], a graph-based FS framework was proposed such that a feature subset is selected in a trace ratio form. In [72], Laplacian Score (LS) was proposed as a feature selection algorithm which is fundamentally based on LE and LPP algorithm. The work presented in [74] created a band adjacency graph (BAG) to use a clustering scheme for feature selection, where the graph nodes represent the spectral bands and the edges represent the similarities.

3.4 Summary

In this Chapter, we introduced graph theory, a mathematical structure used to model pairwise relations between objects, as a tool to model and analyze remotely sensed data, especially those data lying in high dimensional space. TAD is an algorithm base on graph theory. It detects the underlying structure of the image data points in a high dimensional feature space to identify a background model and detect anomalies by comparing every pixel to the model. Moreover, TAD algorithm not only can be used to model the background structure but can be further improved to model the land cover types in the dataset. The automatic learning of class features based on TAD algorithm can be used to create a self-learning training set. The self generated training set allows us to utilize classification scheme in an unsupervised fashion. From the results we can see that TAD generated training set combined with supervised classification scheme works in a fully automatic fashion and outperforms the other unsupervised algorithms.

Manifold learning is under the spectral graph theory which studies the properties of graphs via the eigenvalues and eigenvectors of their associated graph matrices, such as the graph Laplacian and its variants. It has been a useful tool to represent the topology of the high-dimensional nonlinear data in a lower dimensional space while seeking to preserve the original intrinsic data structure. Spectral embedding techniques represent the vertices (data points) of a graph in a space spanned by the eigenvectors of the graph matrix. This is very useful for remotely sensed data preprocessing because the large scale data captured by remote sensors contains redundancy and is computationally expansive to analyze. By transforming the data points into a lower dimensional space with more compact and efficient embeddings, the data points may become more separable and easier to process.

Chapter 4

Spatial-spectral Data Fusion for HSI Clustering

Data fusion for remotely sensed data aims to integrate the information generated with different methods or captured with different sensors to provide more possibilities for data exploitation. However, the multiple data sources will inevitably result in an increase in data volume, dimensionality of the feature space and interclass separability. More importantly, the diverse information will cause redundancy due to repetitious features could be extracted from different data sources. Therefore, feature mining techniques are necessary for multi-source data fusion to effectively extract useful feature primitives in various feature domain or from various recording sensors.

Feature mining skills can help collect useful feature primitives at low-level, mid-level and high-level. Those features are usually defined extracted in different domains and combine to the overall data representation. However, the diverse information usually leads to a high dimensionality in the feature space. Therefore, dimensionality reduction (DR) is a widely adopted tool which removes the redundancy without a significant loss of the useful information for subsequent utilizations. Usually, DR techniques only deal with first-order data representations as the input (i.e. vector representation for a pixel in HSI with its spectral feature). In our work, we explore tensor-based DR approaches which are compatible with second-order data (matrix representation) and even any arbitrary order data as the input. Tensor-based approaches have seen great success in many applications in image processing and machine learning, such as feature extraction via tensor decomposition, subspace analysis for feature extraction, multi-view image matching, and etc. We exploit the use of tensor-based DR approaches for HSI and other remotely sensed data fusion due to the reason that tensors can naturally represent the data in its multi-dimensional structure, and it provides the capability of preserving the relationships between the neighboring data points.

4.1 Feature Mining in HSI

With the rapid development in the acquisition and storage of remote sensing data, a great number of high dimensional and multi-modal data sets are available for analysis. However, remote sensed data consists of a substantial amount of spatial and spectral information which brings a challenge to traditional image and signal processing. In other words, the use efficiency of the huge quantities of data urgently needs improvement.

Data mining is a necessary task for interpreting and extracting useful image content to predict and discover attributes. Figure 4.1 summarizes the main procedures of data mining on remote sensed hyperspectral data. The preprocessing step is essential to accurate data mining which aims at combining heterogeneous data sources and reducing undesired effects to get the data prepared for mining tasks. In order to create a target data set, the input data features need to be carefully selected and gathered to effectively represent the data. Afterwards, different data mining tasks, such as clustering, classification, regression and etc., can be conducted using various modeling approaches on the preprocessed data set.

Identifying effective features to model the characteristics of classes in remote sensed imagery is critical to any mining jobs. Feature mining, which includes feature generation, feature extraction and feature selection, is considered to be an effective tool for information extraction . Considering the fact, the original feature space for remote sensed data may not be the most effective space for data representation and understanding, feature selection and extraction are two strategies that help in finding an alternative compressed feature space. We refer readers to the previous chapter for more information on feature extraction and feature selection techniques. Before selecting a subset of features or transforming data into a lower dimensional space, generating new features from the raw input data space could avoid redundant information and take advantage of essential information. Particularly, techniques for generating spatial and spectral features for hyperspectral images, will be discussed in this section. Given remotely sensed image, tensor representation naturally utilizes the spatial and spectral information in a neighborhood for a given data point (i.e. pixel) and can be effectively handled using multilinear algebra.

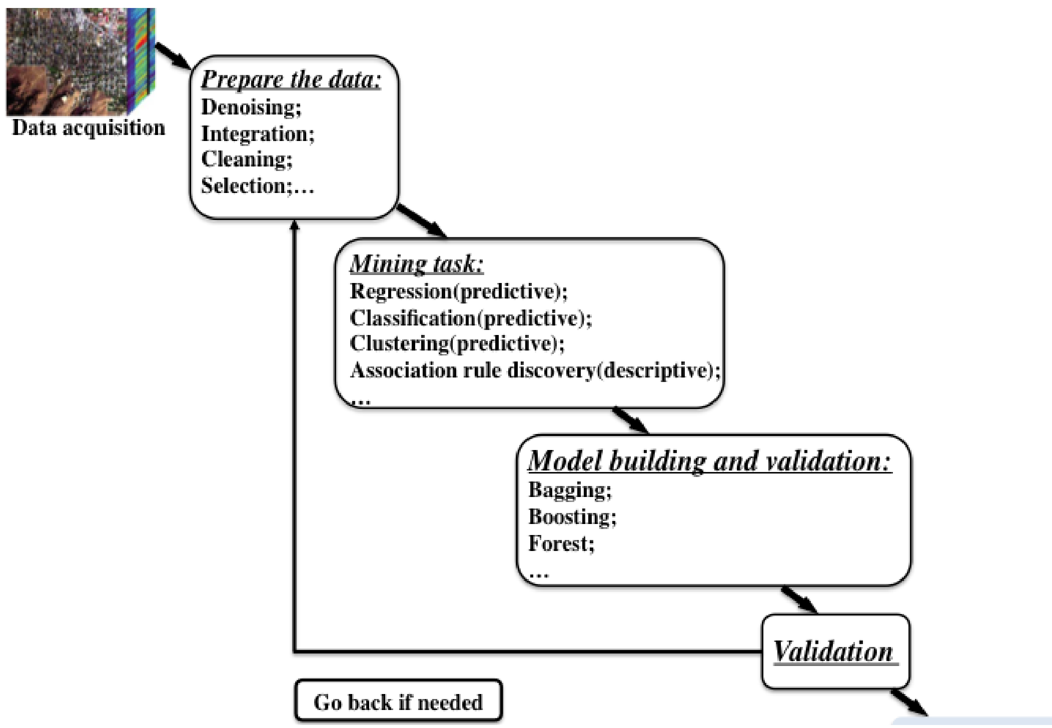


Figure 4.1: Basic data mining procedures for remote sensing imagery.

4.1.1 Spectral feature mining

4.1.1.1 Spectral unmixing

One pixel in an HSI data cube contains detailed spectral information that can usually be used as the feature descriptor. However, one should not ignore that spectral imaging sensors often record scenes in which disparate nearby class instances all contribute to the spectrum measured at one single pixel location, especially if it resides on object boundaries. In other words, the spectrum information collected in one pixel may not be reliable enough to discriminate classes from each other. Given such mixed pixels, not only the spectra of constituent materials need to be identified, but also the proportions in which they appear should be considered.

Spectral unmixing identifies the constituent components of a pixel as the endmembers, and calculates each corresponding fraction as the abundance. The endmembers are normally the natural or man-made objects present in the scene, such as vegetation, soil, water, and etc. It usually consists of three stage processing Figure. 4.2, namely, dimensionality reduction, endmembers determination and inversion stage to obtain the abundance planes [75]. Among the three stages, dimensionality reduction is an optional step which mainly aims at reducing the computational load of subsequent processing.

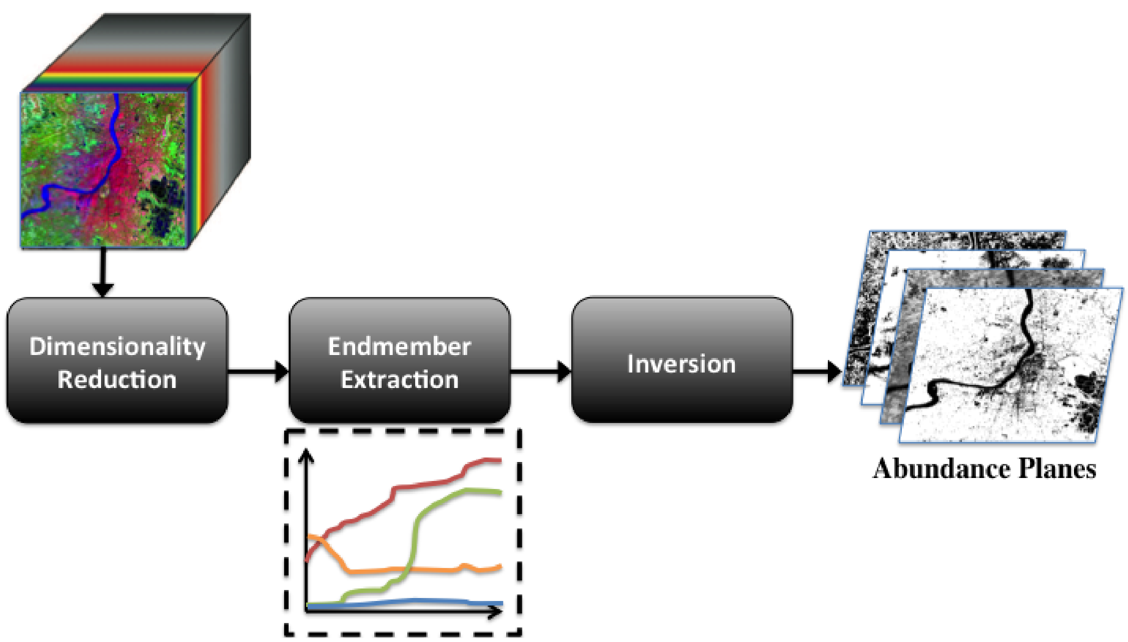


Figure 4.2: Three stages in spectral unmixing.

The constituent components in each pixel are called endmembers because they are usually the vertices of a convex polyhedron formed by the data points in high dimensional space. The pure signature for a class cannot be detected visually from mixed pixels. Also, the quantity of pure pixels could be very limited which also makes endmember extraction a very challenging problem. Many algorithms have been developed for this purpose, we only briefly introduce two most widely adopted endmember extraction techniques: N-FINDR and Pixel Purity Index (PPI).

The N-FINDR algorithm is a selection algorithm [76] which starts with randomly selected pixels as initial endmembers in high dimensional spectral space. Afterwards, new pixels are considered as candidates to replace the old endmembers hoping to have the volume of the simplex increased. This procedure is then repeated until there are no replacements of endmembers left. The N-FINDR algorithm is fully automated with the assumption that the simplex that yields the maximum volume will be the one whose n vertices are the desired purest pixels. Due to the nature of N-FINDR, it is sensitive to the initial endmember set and the existence of noises in the data set.

The PPI algorithm [77] might be the most famous and widely used endmember extraction method, due to its implementation in the ENVI software package. Usually, it performs a dimensionality reduction to the original data cube and a noise whitening step by using the Minimum Noise Fraction (MNF) transformation approach. And then it proceeds to determine the pixel purity by repeatedly projecting data points onto random unit vectors. The extreme pixels in each projection are identified and placed on a endmember list. As the number of unit vectors increases, the list grows and the number of times one data point appears on the list is counted. It should be noted, though PPI has been intensively used, it is not a solution to find the endmembers but should be used as a guide. It usually requires human intervention of a trained analyst to select those extreme pixels that best describe the pure material spectra in the scene.

According to extracted endmembers, the inversion stage produces the abundance maps which can be drawn for each endmembers indicating the fractional presence of pure materials in the mixed pixel spectrum. In this sense, a linear relationship exists between the constituent endmembers comprising the area being imaged and the spectrum being measured at that pixel location. The linear mixing model (LMM) assumes that a spectrum

$\mathbf{x} = [r_1, r_2, ..., r_b]^T$ is a linear combination of endmembers s_i with the abundance of a_i

$$\mathbf{x} = a_1 s_1 + a_2 s_2 + ... + a_n s_n + \mathbf{w} = \sum_{i=1}^{n} a_i s_i + \mathbf{w} = \mathbf{Sa} + \mathbf{w}$$

where n is the number of endmembers, $\mathbf{S}$ and $\mathbf{a}$ are b-by-n matrix representation of endmembers and n-by-1 vector representation of their abundances, respectively. $\mathbf{w}$ is the additive noise vector. The abundance vector $\mathbf{a}$ is usually determined by minimizing a mean-square error cost function, following the non-negativity and sum-to-one constraints:

$$a_i \geq 0, \forall i \in 1, ..., R$$

$$\sum_{i=1}^{n} a_i = 1$$

4.1.1.2 Vegetation index

Different portions in the spectrum contained in each pixel in HSI data have found discriminating power of specific land covers, such as vegetation and water. For live green vegetations, the pigment in plant leaves strongly absorbs visible light for use in photosynthesis, and the cell structure in the leaves, on the other hand, reflects near-infrared light. Therefore, the difference present in the two portions of the measured spectrum is naturally an indicator to determine the distribution of vegetations. The metrics, such as the Normalized Difference Vegetation Index (NDVI), Soil-Adjusted Vegetation Index (SAVI), and Tasseled Cap transformations are useful vegetation measures for remotely sensed data. Before calculating vegetation indices, data must be converted to reflectance as the physical measurement. The vegetation index map can be generated by calculating the value for each pixel. For example,

$$NDVI = (NIR - VIS)/(NIR + VIS)$$

SAVI is similar to NDVI with an additional soil brightness correction factor. It is defined as:

$$SAVI = (1 + L) * (NIR - VIS)/(VIS + VIS + L)$$

where the surface reflectances are used for each band, and L varies by the amount of vegetation cover in the scene. L ranges from 0 to 1 indicating high vegetation region to no vegetation present.

Similarly, some other land cover types can also be monitored based on their unique shape of the spectral absorption and emission curves, such as soil, water, snow, etc. The associated index map of each land cover type can be useful spectral feature for pixel classification.

4.1.2 Spatial feature mining

In recent years, advances in hyperspectral remote sensing allow the simultaneous acquisition of detailed spectral information for each image pixel and increase the possibility for better discrimination between various materials in a scene. Furthermore, the finer spatial resolution of the remote sensing systems enables the analysis of spatial structures in hyperspectral images by taking texture and contextual information into consideration. Useful information can be extracted from the spatial domain, such as the size and the shape of the structure to which one pixel belongs, can also provide the potential to reduce the clustering uncertainty that exits when only spectral information applies. Therefore, spatial feature mining becomes important for better classification and segmentation.

Many techniques for extracting spatial features have been proposed and successfully applied on remote sensed imagery. Those techniques includes statistical based, such as gray-level co-occurrence matrix (GLCM) [78]; signal processing based approaches, such as Gabor filters [79]; model-based approaches, such as MRFs [16]; and geometrical approaches [25].

4.1.2.1 GLCM

GLCM, as the most common technique for texture feature extraction in the remote sensing literature, uses a matrix to represent texture characteristics by counting pixel intensity pairs and represent them as corresponding entries in the matrix as shown in Figure 4.3. The pixel pair $P_{ij}(d, \theta)$ of i and j within the neighborhood $N \times N$ has a displacement of $d = 1, 2, ..., N - 1$ between them with an angle of $\theta = 0°, 45°, 90°, 135°$. i and j are the pixel intensities for the pixel pair, and the columns and rows in GLCM correspond to the level of pixel intensities in an image. The count of a pixel pair of displacement d and angle θ with specific intensities are recorded at the corresponding entry of the the co-occurrence matrix. Based on co-occurrence matrix, Haralick feature descriptors [78], such as entropy, a measure of the degree of disorder in an image; and Inverse Difference Moment (IDM), a measure of the amount of local similarity, and many other texture related features, can be derived for each pixel from its neighborhood. The entropy is defined as

$$-\sum_{i}^{N}\sum_{j}^{N} P_{ij} log P_{ij}$$

Inverse Difference Method is defined as

$$\text{IDM: } -\sum_{i}^{N}\sum_{j}^{N} \frac{P_{ij}}{1 + |i-j|^2}$$

Each feature calculated from the co-occurrence matrix extracts different texture knowledge of the neighboring structure to which one pixel belongs. By stacking all the features for one pixel, a feature vector is obtained to describe the local texture structure.

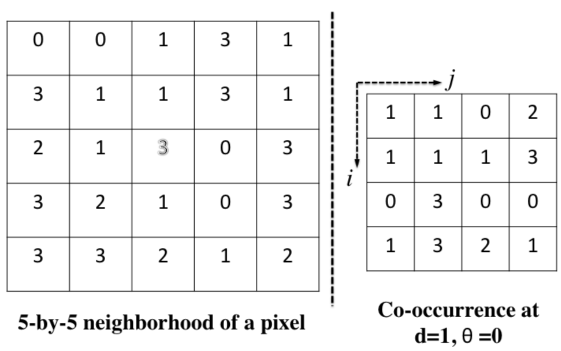


Figure 4.3: 5-by-5 neighborhood patch of one pixel and the co-occurrence matrix for the patch for pixel pair of d=1, $\theta = 0$.

4.1.2.2 Gabor filters

Unlike the statistical based GLCM method, Gabor filtering is a signal processing technique which generates appropriate texture features in a more computationally efficient way than the co-occurrence matrix. Gabor filters have found great success in the texture analysis literature [79] owing to their concise mathematical expression, ease of implementation for multi-channel filtering, and optimal joint localization, or resolution, in both the spatial and the spatial-frequency domains [80].

A Gabor filter bank generates a set of near-independent estimates of the local frequency content in an image in the form of response images. Several Gabor filters with different parameters applied to an image will produce a stack of response images as shown in Figure 4.4. Usually, a Gabor filter is a sinusoid function $s(x, y)$ modulated by a Gaussian envolope $g(x, y)$:

$$\begin{aligned} h(x, y) &= s(x, y)g(x, y) \\ &= e^{-\frac{1}{2}(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2})} e^{-j2\pi(u_0 x + v_0 y)} \end{aligned} \tag{4.1}$$

The σ_x, σ_y are the standard deviation in the Gaussian function, u_0, v_0 contains the spatial frequency and orientation information. A given image can be convolved with multiple Gabor filters according to different orientations and frequencies. Therefore, features such as the mean and standard deviation with respect to the local neighborhood can be characterized by the generated response images.

4.1.2.3 Markov Random Field

Markov random fields (MRFs) have been explored widely in many applications in image processing, such as classification, segmentation, and texture synthesis. They have been demonstrated to be effective tools for texture characterization [81, 16]. In image processing, a MRF model is a two-dimensional lattice, where each point is assigned a value based upon a probabilistic model. In a MRF model, points must exhibit this attribute of Markovianity, that is, the value of a point on the 2D lattice is dependent only on its neighbors. Given one variable in a $N \times M$ lattice L, e.g., $X(c)$ where $c = 1, 2, 3, ..., N \times M$,

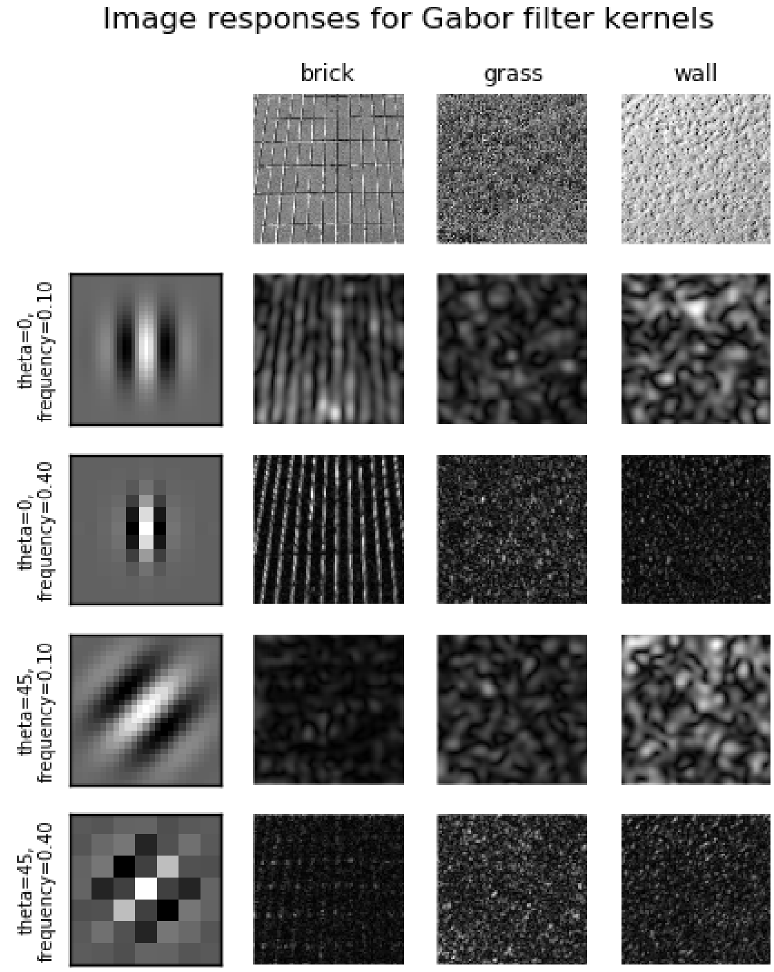


Figure 4.4: Three images (brick, grass, wall) with different texture structures filtered with Gabor filter banks of four different frequency and orientation. Retrieved from http://scikit-image.org/docs/dev/auto_examples/plot_gabor.html

according to Markovianity property, if point $X(m)$ is in the neighborhood δ_c of $X(c)$ then the distribution $p(X(c))$ of the random variable $X(c)$ depends only on the value $X(m)$.

$$P(X(c)|X(m), m = 1, 2, ..., N \times M, c \neq m) = p(X(c)|\delta_c).$$

In a Gaussian Markov Random Field (GMRF), the local conditional probability distribution is assumed to be Gaussian, thereby we have:

$$P(X(c)|X(m), m \in \delta_c) = (2\pi\sigma^2)^{-1/2} exp[-(X(c) - \Sigma\beta_{c,m}\ (X(m) + X(m')))^2/2\sigma]$$

where σ is the standard deviation in Gaussian kernel and $\beta_{c,m}$ is the texture feature for the MRF model. $X(m)$, $X(m')$ is a pair of symmetric neighbors for $X(c)$ and the summation is over all pairs of such symmetric neighbors in the neighborhood δ_c for center pixel $X(c)$. Figure 4.5 shows up to the 5^{th} order symmetric neighbors, each pair of pixels centers at $X(c)$ with opposing angles and increasing distances to $X(c)$. In Figure 4.5, the random field is modeled by five order neighbors and be characterized by level feature parameters in total.

Additional high-order neighbors can be defined by extending the orders in a similar way beyond those in Figure 4.5. In order to obtain the texture feature parameters $\beta_{c,m}$ from a MRF model, as well as estimating the optimal model order for a given dataset [82], a number of techniques have been proposed and discussed [83, 84]. The parameters estimated in a local neighborhood for each pixel thereby forms the texture feature vector.

5.1	**4.2**	**3.1**	**4.3**	**5.2**
4.1	**2.1**	**1.1**	**2.2**	**4.4**
3.2	**1.2**	**X(c)**	**1.2**	**3.2**
4.4	**2.2**	**1.1**	**2.1**	**4.1**
5.2	**4.3**	**3.1**	**4.2**	**5.1**

Figure 4.5: 1^{st} to 5^{th} order symmetric neighbors to central pixel $X(c)$ in a MRF model.

4.1.2.4 Morphological profile

Benediktsson et al. [25] defined the morphological profile (MP) by performing a sequence of closings and openings by reconstruction of increasing sizes (i.e., anti-granulometry and granulometry) on satellite images and stacking each of the processed images together to build a multi-dimensional image as a MP. The advantage of applying openings/closings by reconstruction is it can suppress brighter/darker areas that are smaller than the structuring element (SE). On the other hand, structures that are larger than the SE are well preserved. The SE, which specifies the neighborhood considered for each pixel, defines the amount of contextual information included in the morphological analysis. In recent literature, MPs have been widely used for HSI analysis, especially for image data clustering and classification.

Morphological profile and extended morphological profile

Opening by reconstruction is computed as a sequence of erosion with selected SEs followed by a reconstruction by dilation [85]. By applying opening by reconstruction on a grey-scale image with SE of increasing size and stacking the resulting processed images together, we obtain a morphological opening profile, $\prod_{\phi}$, which is a multi-dimensional image in which each layer contains the processed grey-scale image using a morphological filter according to one specified SE. By duality, a morphological closing profile, $\prod_{\gamma}$, is obtained with a sequence of closing by reconstruction operators. Applying such operators with multiple SEs of increasing sizes, one can extract information about the contrast and the size of the structures present in the image. This concept is called granulometry in mathematical morphology. The MP of a grey-scale image f is then simply the concatenation of its closing and opening profiles with n SEs, written as:

$$MP(f) = \prod_i \begin{cases} \prod_i = \prod_{\phi_\lambda}, & \lambda = (n - i + 1), \forall \lambda \in [1, n] \\ \prod_i = \prod_{\gamma_\lambda}, & \lambda = (i - n - 1), \forall \lambda \in [n + 1, 2n + 1] \end{cases}$$

The resulting morphological profile of size n has been defined as the composition of a granulometry of size n built with opening by reconstruction and an anti-granulometry of size n built with closing by reconstruction, plus the original image which results in a stack of $2n + 1$ images.

Extended morphological profile (EMP) [26] is a generalization of the MP for hyperspectral data by stacking the MP obtained from the first PCA bands, i.e., those that account for most of the variance of the data in the original feature space. Thus, the EMP of c PCAs of HSI I can be formalized by:

$$EMP(I) = \{MP(PC_1), MP(PC_2), ..., MP(PC_c)\}.$$

Although EMP has been demonstrated as a powerful tool for exploring spatial structures of different scales, it only relies on the interaction of a range of SEs of fixed shapes with the image, while ignoring the shape and texture information for the objects in the image. Another limitation of EMP is the computational complexity. The computation time increases linearly with the number of SEs and PCA bands used for the data [27] because each grey-scale image has to be completely processed for each layer in the profile, requiring two separate processes: closing and opening.

Attribute profile and extended attribute profile

Mauro Dalla Mura et al. [27] proposed to use morphological attribute filters for HSI analysis. A morphological attribute filter is a connected filter [86] that only removes self-existent connected components in a binary/grey-scale image that do not fulfill a given attribute criterion. With different attributes, the attribute profile (AP) permits more flexibility to model the spatial information with respect to MP. The attributes measured for each connected component can be geometric (e.g. area, length of the perimeter, image moments, shape factors), textural (e.g, range, standard deviation, entropy), etc. Furthermore, with a Max-tree data representation, AP can be efficiently generated. For the sake of conciseness, we only introduce the definitions of AP and extended AP, and we refer readers to Refs. [27] for a detailed Max-tree implementation.

Analogously to MP, the AP can also be defined as a concatenation of attribute thinning $\prod_{\hat{\gamma}^{u_\lambda}}$ and thickening $\prod_{\hat{\phi}^{u_\lambda}}$ profiles based on a set of ordered attribute criteria $U = \{U_\lambda : \lambda = 0, ..., n\}$:

$$MP(f) = \prod_i \begin{cases} \prod_i = \prod_{\hat{\phi}^{u_\lambda}}, & \lambda = (n - i + 1), \forall \lambda \in [1, n] \\ \prod_i = \prod_{\hat{\gamma}^{u_\lambda}}, & \lambda = (i - n - 1), \forall \lambda \in [n + 1, 2n + 1] \end{cases}$$

Similarly, we can also compute the extended attribute profiles (EAP) on the first c PCAs of HSI:

$$EAP(I) = \{AP(PC_1), AP(PC_2), ..., AP(PC_c)\}.$$

Moreover, by using different attributes for different spatial information from the scene, the idea of EAPs can be further evolved to extended multi-attribute profile (EMAP) by the contatenation of m different EAPs:

$$EMAP(I) = \{EAP_{a_1}, EAP_{a_2}..., EAP_{a_m}\}.$$

Four attributes were considered in the analysis in order to include shape and textural information: area, diagonal of the bounding box, moment of inertia and standard deviation of the pixels in the region (connected component). The benefit of using EAP instead of EMP is the great flexibility in the attributes selection with respect to traditional SEs adopted in EMP. Since in many cases, a compact SE with specified shape may not fit in the actual structure in the image. The other benefit of adopting EAP instead of EMP is the reduced computational complexity via the Max-tree structure, which avoids the multiple computations associated with the levels of profiles.

4.2 Self-tuning spatial-spectral clustering

TAD algorithm, as one of the early work involved in analysis of the topological structure of hyperspectral data, provides a good model to learn the clusters in the data set. In machine learning and computer vision research field, spectral clustering has become one of the most popular modern clustering algorithms which is based on graph representation and topological structure of the data. Recently, spectral clustering algorithms have been applied to hyperspectral data. Those algorithms, based on a graph representation of the data, are easy to implement and have been shown outperforming traditional clustering schemes such as k-means in many cases.

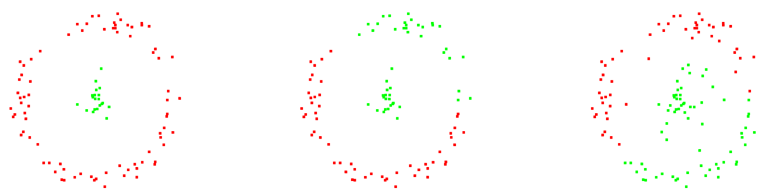
Spectral clustering is the task of grouping similar vertices into clusters, which takes the edge structure into consideration in such a way that there should be many edges within each cluster and relatively few between the clusters. It clusters data points through performing spectral analysis on the matrix representing the graph structure. Specifically, it consists of two stages: (a) build an affinity matrix by creating edges and assigning weights; (b) find optimal partition of the affinity matrix. The first stage is critical to the performance of graph clustering for it encodes the intrinsic structure of the data. Many existing spectral clustering algorithms adopt the Gaussian kernel function as the similarity measure for its simplicity; and use the 4 or 8-connected graph which results in a positive definite similarity matrix that simplifies the analysis of eigenvalue and is easy to implement. However, the scaling parameter σ in Gaussian kernel needs to be specified manually, and is data-dependent. The choice of σ also makes spectral clustering sensitive to outliers. Also, the simple 4-connected or 8-connected graph is to connect data points based on spatial proximity, which only spatially connect pixels to its surrounding neighbors without considering the inherent similarities or accounting for data on "different scales".

Hyperspectral data contains high discriminating power in the spectral space. To better accommodate spectral clustering algorithms to hyperspectral data, we proposed to use an improved affinity matrix which takes the true distribution of the data into account and automate the selection of tuning parameters to handle outliers and data "on different scales" [87]. We apply our proposed affinity matrix on Laplacian Eigenmaps (LE), one of the most widely used spectral clustering algorithm.

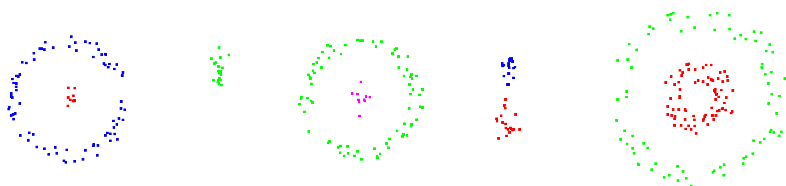
4.2.1 Self-tuning affinity matrix

Hyperspectral imagery offers the potential to combine color constancy with a high discriminating power in the spectral domain. The adaptive k nearest neighbor approach is adopted here instead of the 8-connected graphs that are widely used in RGB or Binary images. The spatial information is included as the smooth term in weights computation.

The similarity measure between every pair of vertices linked by an edge is very important in affinity matrix. The Gaussian kernel $w_{ij} = exp(-\frac{|x_i - x_j|^2}{\delta^2})$, is one of the widely used metric for similarity measure in affinity matrix. It has shown good performance on describing the information of the local consistency. The tuning parameter σ in the Gaussian kernel plays a significant role on characterizing the intrinsic local data structure in a graph. Usually, the value of σ is fixed for every pair of data points. The obvious drawback using a fixed σ is that it is hard to find a value that could maintain optimal for different data sets. In other words, we have to manually specify the value of σ given different data sets. Ng et al. proposed to select the optimal σ by repeatedly running the clustering algorithm using a number of different σ and use the one which gives the best result. But it is very time consuming and still uses the same σ for all data points without considering local density Figure 4.6.



(a) σ=3.3586 (b) σ=6.7173 (c) σ=3.3586(noisy)



(d) σ=3.0482 (e) σ=2.9188 (f) σ=3.3897

Figure 4.6: The example of LE with fixed scaling parameter σ. Top row: a small perturbation in the scaling parameter σ or in the data points gives rise to very different results. Bottom row: the optimal σ for each data set turned out to be different.

In contrast to using a single σ for a similarity measure of every data point, Zelnik-Manor et al.[88] suggested a local scale similarity measure by calculating a scaling parameter for each data point according to its local data density. This is based on the observation that if the input data contains different scales, i.e. data points in different clusters have different local densities, a single σ may not be able to model the separability between clusters. They suggest to use $w_{ij} = exp(-\frac{|x_i - x_j|^2}{\sigma_i \sigma_j})$, where σ_i is the distance between point x_i and its k_{th} nearest neighbor. k is set to be 7 in Refs. [88]. Comparison of the effect using a global σ and local σ_i is illustrated below. The edge thickness reflects the similarity be-

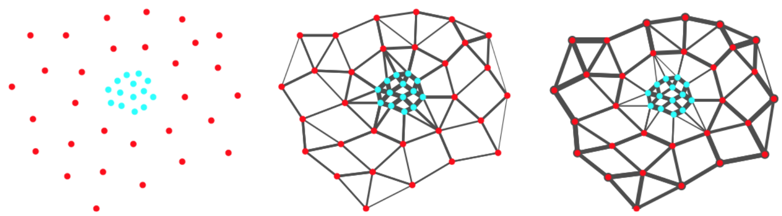


Figure 4.7: (a) Input data points. (b) Graph constructed use Gaussian kernel with a uniform σ. (c) Graph constructed use local scale similarity measure.

tween two data points. Without considering local density, a Gaussian kernel fails creating strong links between data points in the red cluster and penalizing links between red and cyan clusters. In Fig. 4.7.c, we have edges across the two clusters significantly weaker than those within each cluster.

Learned from above, a good similarity measure for spectral clustering should take the local neighborhood into consideration in order to truly capture the cluster separability instead of relying only on the Euclidean distance between data points. When outliers exist in the data set, the local scale parameter σ_i, which is defined according to the distance to the k_{th} nearest neighbor, can not make any contribution to clustering better than using a fixed global σ for similarity measure. The outliers, especially those lying in the overlapping regions between two clusters, usually have different neighborhood density compared to other majority data points. Uniformly defining the local σ_i to be the distance to the k_{th} nearest neighbor for each data point lacks the capability to handle outliers. In Refs. [89] we proposed a self-tuning similarity measure method which effectively handles outliers, and includes combinatorial information as additional connectivity measure.

Our proposed self-tuning similarity is based on the following two observations: 1. For two data points x_i, x_j, local tuning parameter σ_i measured to a further neighbor gives larger similarity via the Guassian kernel $w_{ij} = exp(-\frac{|x_i - x_j|^2}{\sigma_i \sigma_j})$. 2. Two data points tend to stay in the same cluster if they share more common neighbors with each other. The adaptive k nearest neighbor approach has a significant implication with respect to the true distribution of data in the real world. It takes into account the fact that outliers should have fewer neighbors compared to the majority of data points lying within any clusters.

On the other hand, it makes the σ_i selection fully automatic instead of hard-coded as the distant to k_{th} neighbor for all data points.

(a) Data Points (b) large k (c) small k (d) adaptive k

Figure 4.8: Comparison of graphs constructed via Gaussian kernel for similarity measure using local tuning parameters. σ_i for data point x_i is defined as its distance to the k_{th} neighbor. In (b) and (c), k is set to a fixed value for all data points. In (d), k is obtained by adaptive k-nearest neighbor approach.

In Fig. 4.8, the input data set has two clusters marked as red and blue respectively. The points in between are outliers and form an overlapping region for the two clusters. With Zelnik-Manor's method to calculate the local tuning parameter as distance to the k_{th} neighbor, a larger value of k forces stronger connectivity in the overlapping area, while a smaller value of k leads to weaker connections among all data points. Neither way has the ability to assign larger similarity for data points within each cluster and smaller similarity for data points between clusters. On the contrary, adaptive k-nearest neighbor approach finds more neighbors for the data points lying in the center for one cluster, while fewer neighbors for outliers in the overlapping area between two clusters. According to our first observation, the tuning parameter σ_i for outliers is measured as the distance to a closer neighbor using adaptive k nearest neighbor method, thereby it builds weaker connections among them.

The local tuning parameter only captures individual local density without combinatorial information for the computation of similarity between two data points. The number of neighbors two vertices have in common provides complementary information since two data points have more common neighbors indicating higher probability that they belong to the same cluster. This is especially useful for two well-separated clusters with similar local densities. For more discussion, we refer readers to Refs. [89]. The final similarity measure in spectral Euclidean space becomes:

$$w_{ij} = exp(-\frac{|x_i - x_j|^2}{\sigma_i \sigma_j c_{ij}})$$

where σ_i, σ_j are defined as the distance to the adaptive-k nearest neighbor for point x_i, x_j, respectively, and c_{ij} is the number of neighbors x_i, x_j have in common.

4.2.2 Graph-based region merging to reduce over-segmentation

Based on the second smallest eigenvector, LE algorithm computes a multi-partitioning using the recursive splitting scheme until it meets all stopping criteria. However, the data is usually over-segmented because there is no stopping rule that will work perfectly for every partition if no human interaction is included during the process. We proposed to automate the merging step to produce finer segmentation by combining over-segmented partitions.

Here, a greedy region merging approach that gradually merges over-segmented parts into larger and meaningful areas is proposed based on the Region Adjacency Graph (RAG). RAG builds a graph to analyze the relationships between the over-segmented partitions, where nodes represent the partitions that need to be merged and edges represent the likelihood two nodes need to be merged. Once RAG is built, the two partitions that satisfy the merging criterion will be merged. Then RAG is updated and the process repeats until a certain condition is met or terminated manually.

Many merging criteria for RAG have been proposed, such as complete-link, average-link, boundary smoothness, region homogeneity, and etc [90]. Those criteria mostly depend on the pixel intensity difference between two partitions or the spatial boundary information. We introduced a graph based criterion by comparing two metrics between the partitions: one is based on the relative pixel densities in crowding or overlapping area between two partitions. The overlapping area refers to the set of pixels that their edges connect two partitions in the graph used in LE algorithm. The other one is based on the absolute pixel density of each partition. The theory in behind is two partitions A and B should be merged into one single cluster if the pixel density in the overlapping area C is high compared to the pixel density in each partition. Adaptive k nearest neighbor value

n_{pp} has a significant implication of the relative data density around each pixel. And the average edge weight w_{pp} provides the absolute pixel density around each pixel since the weight in our improved affinity matrix captures the spectral similarities, multi-scale data structure and the boundary smoothness. By combing both relative and absolute local density information, the merging criterion for RAG is defined as:

$$D(A,B) = \begin{cases} merge, & if\ D_{diff}(C) \geq Int(A,B) \\ not\ merge, & if\ D_{diff}(C) < Int(A,B) \end{cases}$$

where $D_{diff}(C)$ determines the dissimilarity between A and B by examining the densities of the overlapping area C, while $Int(A,B)$ determines the local density information within each partition A and B. They are defined as:

$$D_{diff}(C) = norm(n_{pp})_c + norm(w_{pp})_c$$

$$Int(A,B) = min(D_{diff}(A), D_{diff}(B))$$

where $norm(n_{pp})_c$, $norm(w_{pp})_c$ scales the values to be on a scale of 0 to 1. Two partitions are considered to be the same class if they have an unclear boundary in between that as the density in overlapping area is higher than that in each partition.

Figure 4.9 shows the results using our proposed spatial-spectral clustering method performed only on pixels in ROIs in Cooke City and a subset of the SalinasA image. The merging step successfully produces a finer segmentation map. Also, Figure 4.10 shows our graph based clustering method outperformed the other three unsupervised segmentation algorithms. Quantitative comparisons could be found in Table 4.1. Our proposed self-tuning clustering method is based on LE algorithm with improved affinity matrix that considers the local density, multi-scale data structure and the boundary smoothness between different classes in the data. We also modified the iterative clustering scheme used in the Ncut algorithm by the use of RAG to further merge subclasses to obtain a finer segmentation map.

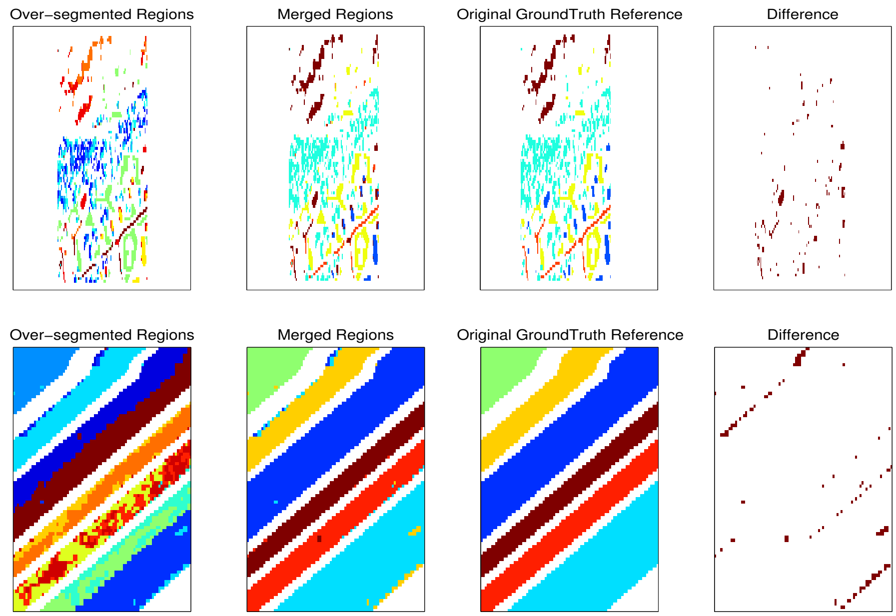


Figure 4.9: Results on Ground Truth only of Cooke City and Salinas Scene.

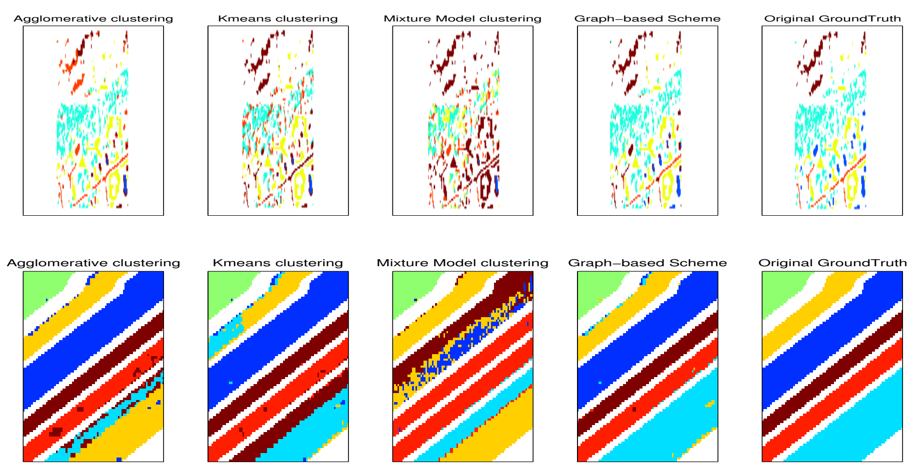


Figure 4.10: Comparison of segmentation results of Agglomerative clustering, K-means and Gaussian Mixture Model and our graph based splitting and merging.

4.3 Spatial-spectral clustering use morphological operations

For remotely sensed data, especially high-resolution HSI data, one pixel contains significant amount of contextual information other than the detailed spectra. For a given pixel

Table 4.1: Accuracy of using agglomerative clustering, K-means, gaussian mixture model and our graph based scheme for Cooke City and Salinas-A data sets.

		Agglomerative	K-means	Mixture Model	graph based
Cooke City	Buildings	0.1745	0.2028	0.1509	0.5000
	Trees	0.8861	0.5942	0.4399	0.9439
	Grasses	1.0	0.9965	0	0.9964
	Roads	0.9856	0.2086	0.3525	0.7122
	Grey	0.3417	0.9964	1.0	0.9928
	Overall Accuracy	**0.7901**	**0.6809**	**0.3687**	**0.9083**
SalinasA	Lettuce_romaine_5wk	1.0	0.9987	0.2269	0.9980
	Corn_senesced_green_weeds	0.2517	0.5607	0.3857	0.9702
	Broccoli_green_weeds_1	0.9974	0.9974	0.9949	0.9923
	Lettuce_romaine_4wk	0.9448	0.6363	0.9691	0.9302
	Lettuce_romaine_7wk	0.9086	0.9674	0.9737	0.9749
	Lettuce_romaine_6wk	1.0	0.9985	0	0.9866
	Overall Accuracy	**0.7919**	**0.8422**	**0.4914**	**0.9779**

we can extract the shape, size and texture information of the structure to which it belongs. This information will complement the spectral signatures to help discriminate and identify different objects in a scene. Consequently, a joint spatial-spectral data fusion strategy is needed to generate more accurate results in HSI image segmentation and classification.

In our previous spatial-spectral graph based clustering scheme, spatial and spectral information together plays a very important role for constructing a good affinity matrix. The spatial information that is used to characterize the similarity of two points is the Euclidean distance on the two dimensional grid structure. As a source of information to measure the spatial similarity of pixels, the Euclidean distance only replies on the location of two given pixels without considering their local structure and relationship to each other. As mentioned in Chapter 4.1.2.4, morphological operators have the ability to suppress brighter/darker areas that are smaller than the structuring element (SE), which specifies the neighborhood considered for each pixel and defines the amount of contextual information included in the morphological analysis. Based on that, the attribute filters are also applied on HSI and other remote sensed data. The attribute operations only remove self-existent connected components in a binary/grey-scale image that do not fulfill a given attribute criterion, thereby it permits more flexibility to model the spatial structure with the use of different attributes. The attribute profile (AP), which is a stack of all the images that have been processed according to different attributes (i.e. geometric attributes: area, length of the perimeter, image moments, shape factors, textural attributes: range, standard deviation, entropy, etc.), may be considered as the additional spatial information source. The Euclidean distance for every pair of pixels in this high dimensional space may be used to characterize similarities that include local geometric and textural information.

4.3.1 Composite kernels for joint spatial-spectral clustering

In our proposed self-tuning spectral clustering scheme, the pairwise similarity of two given pixels is measured from spectral and spatial domain use Gaussian kernel, respectively. As mentioned above, the EAP associated with a hyperspectral image is also a multi-dimensional image that each layer is a grey-level image filtered with a specific attribute filter, which encodes the shape and textural information such as area, bounding box, moment of inertia and standard deviation of the structure to which one pixel should belong.

In other words, an EAP of hyperspectral data contains the spatial information embedded in a high-dimensional feature space, as a complementary data source to the high-dimensional HSI. When addressing multiple data sources, separate affinity matrices can be efficiently combined via composite kernels. The composite kernel approach for joint spatial-spectral clustering and classification has been exploited in the machine learning domain, especially in the context of Support Vector Machines (SVMs). Given the input p-dimensional HSI in space $\mathbb{R}_s^p$ and the input data sets $X_s = \{x_1^s, x_2^s, ..., x_N^s\}$, N samples included, where $x_i^s \in \mathbb{R}_s^p,\ i = 1, 2, ..., N$, we may also generate a c-dimensional spatial data space $\mathbb{R}_a^c$ where each data point is represented by its EAP vector $X_a = \{x_1^a, x_2^a, ..., x_N^a\}$, $x_i^a \in \mathbb{R}_a^c,\ i = 1, 2, ..., N$. The input HSI data space is p dimensional with p equal to the number of spectral bands, while the dimensionality c in the spatial data space is the product of the total number of attribute filters n_{ai} multiplied by the number of SEs n_{as} associated with each filter and the number of PCA bands being considered n_{pca}: $c = n_{pca} \sum\limits_{i,s} n_{ai} n_{as}$. Further, we refer to the self-tuning Gaussian kernel used for spectral graph matrices as $K^s(x_i^s, x_j^s)$. Similarly, the kernel used in graph matrices for spatial data space is denoted as $K^a(x_i^a, x_j^a)$.

Given the spectral and spatial pixel vectors x_i^s and x_i^a in the HSI input space and EAP feature space respectively, together with their kernels $K^s(x_i^s, x_j^s)$, $K^a(x_i^a, x_j^a)$, we will present four composite kernel approaches for the joint consideration of spectral and spatial information in a unified self-tuning spectral clustering framework for HSI data sets.

A. Tune by spatial Euclidean distance method.
We have proposed to include the spatial Euclidean distance $d(x_i - x_j)$ (i.e. the spatial distance between two pixels in the 2D image grid structure) as a smoothing term to the spectral kernel in Refs. [89]. By including the spatial distance, the final affinity matrix tends to assign higher weights only to those pixels that are similar both in the spectral feature space and are located closely in the 2D image grid structure. As a smoothing term, the spatial kernel $exp(-\frac{d(x_i-x_j)^2}{\sigma_d^2})$ is a tuning parameter to the graph which is created in the spectral feature space.

$$w_{ij} = exp(-\frac{|x_i^s - x^s j|^2}{\sigma_i \sigma_j c_{ij}}) \cdot exp(-\frac{d(x_i - x_j)^2}{\sigma_d^2})$$

B. Tune by EAP method.
Similar to the previous approach, a complementary smoothing term is defined according to the Euclidean distance between two data points in the EAP space $d(x_i - x_j)^2_{eap}$. Instead of penalizing two pixels if they are spatially apart, the smoothing term evaluates the multi-scale neighborhood structure associated with two pixels respectively, and tunes the edge weights by assigning higher weights to pixels that share similar contextual information, while lower weights are assigned to pixels if they belong to different shape, or textual structure. This can be written as:

$$w_{ij} = exp(-\frac{|x_i^s - x_j^s|^2}{\sigma_i \sigma_j c_{ij}}) \cdot exp(-\frac{d(x_i - x_j)^2_{eap}}{\sigma_{eap}^2})$$

C. Weighted summation method.
The two methods proposed above create the graph with spectral features and further tune the edge weights according to the spatial information. The weighted summation method

provides a good balance between the graphs defined separately in the spectral space and EAP space. We write the joint kernel as:

$$K(x_i, x_j) = (1-\lambda)K^s(x_i^s, x_j^s) + \lambda K^a(x_i^a, x_j^a)$$

where λ is a tuning parameter ($0 < \lambda < 1$) that controls the trade-off between similarity of points x_i, x_j characterized in spectral and EAP space . This allows us to use *a priori* knowledge about the content of the image, or simply use a default value. In our study, we currently set λ equal to 0.2 with more emphasis on the spectral features. It should be noted that, the summation method is an entry-wise summation that if either of the two graphs have non-zero entries in location x_i, x_j, then we will have a non-empty element in that location in the joint matrix. This is similar to an "or" condition in the individual graphs.

D. Multiplication method.

The graph combination approach can easily be modified as the multiplication of two kernels defined separately in spectral and spatial space. In this way, strong edges only exist between data points that are both similar in the spectral feature space and the EAP spatial feature space. Other edges built in either spectral or spatial domain will be discarded, hence this methods results in a sparser affinity matrix. This is similar to an "and" condition in the individual graph. We write this kernel as:

$$K(x_i, x_j) = K^s(x_i^s, x_j^s) \cdot K^a(x_i^a, x_j^a)$$

The four composite kernel approaches will be experimentally evaluated on HSI data sets in the following section. Here, we give an overview flowchart of our proposed joint spatial-spectral graph based clustering scheme Figure 4.11:

4.3.2 Conductivity matrix: block-diagonal structure amplified affinity matrix

A composite kernel unifies both spatial and spectral information into one affinity matrix. An ideal affinity matrix for spectral clustering should have strong connections within each cluster while weak connections between different clusters. By reordering the rows and columns according to the assigned labels, the block-diagonal structure of the affinity

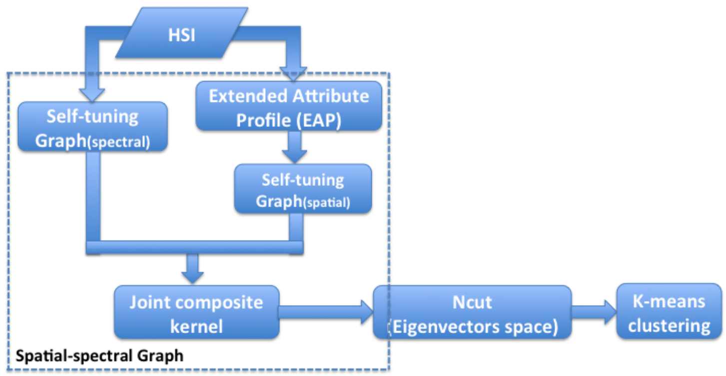
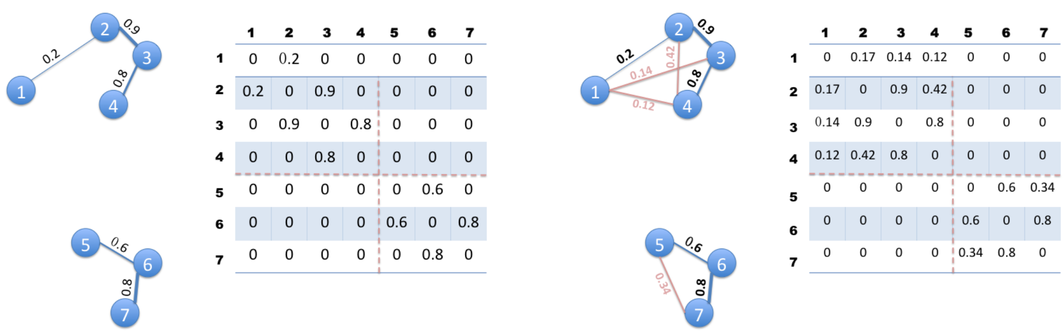


Figure 4.11: Flowchart of the proposed spatial-spectral clustering scheme.

matrix, which has two clusters with no edges in between, could be visualized as in Figure 4.12.



	1	2	3	4	5	6	7
1	0	0.2	0	0	0	0	0
2	0.2	0	0.9	0	0	0	0
3	0	0.9	0	0.8	0	0	0
4	0	0	0.8	0	0	0	0
5	0	0	0	0	0	0.6	0
6	0	0	0	0	0.6	0	0.8
7	0	0	0	0	0	0.8	0

	1	2	3	4	5	6	7
1	0	0.17	0.14	0.12	0	0	0
2	0.17	0	0.9	0.42	0	0	0
3	0.14	0.9	0	0.8	0	0	0
4	0.12	0.42	0.8	0	0	0	0
5	0	0	0	0	0	0.6	0.34
6	0	0	0	0	0.6	0	0.8
7	0	0	0	0	0.34	0.8	0

Figure 4.12: Left: Graph with two clusters and its affinity matrix. Right: Reinforced graph with conductivity matrix.

Real-world data is seldom block-diagonal due to many factors, such as cluster overlapping, in-cluster variation, and etc. From matrix perturbation theory, spectral clustering can produce satisfactory results when the affinity matrix has block-diagonal structure [91]. This theory motivates an improved version of the affinity matrix, named the *conductivity matrix*, with amplified block-diagonal structure. Proposed by Igor Fischer et al. [92], instead of considering two points similar if they are connected by a high-weight edge in the graph, we assign them a high affinity if the overall graph *conductivity* between them is high. Given an affinity matrix A, a *conductivity matrix* can be derived as:

$$G(p,q) = \begin{cases} if \;\; p == 1, & \begin{cases} 1, & if\; q = 1 \\ 0, & else \end{cases} \\ else, & \begin{cases} \sum_{k \neq p} A(p,k), & if\; p = q \\ -A(p,q), & else \end{cases} \end{cases}$$

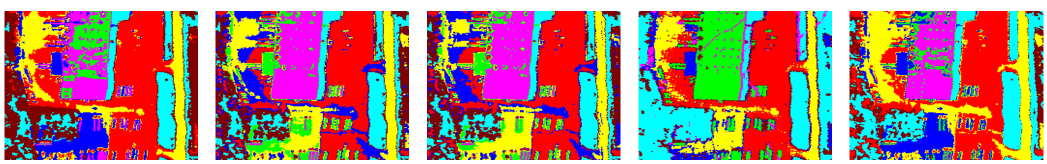
$$C(i,j) = 1/[G^{-1}(i,i) + G^{-1}(j,j) - G^{-1}(i,j) - G^{-1}(j,i)]$$

C is the *conductivity matrix* derived from affinity matrix A with the effect of amplifying the block-diagonal structure. The *conductivity* comes from electrical engineering, where a similar method is named node analysis. We refer readers to Refs. [92] for detailed explanation. Though the resulting *conductivity matrix* C is derived from A, it actually considers the flow between two points and constructs a new graph based on it. C thereby displays a reinforced block-diagonal structure. The simple graph in Fig. 4.12 would have a stronger connection within each cluster with the use of *conductivity matrix* C. The red edges are complementary edges obtained from C to further reinforce the connection within each cluster according to the *conductivity* flow between every pair of data points.

Different attribute filters process the image according to different geometry or texture information in the image. An image filtered with different attributes results in different emphasis on the spatial structure. The "area of the region" attribute filter and the "diagonal of the box bounding" attribute filter provide a measurement of the size of the objects contained in an image. With different sizes of SEs used in the filtering, regions of different sizes would be uncovered in the hierarchical data representation in an EAP. The "moment of inertia" attribute filter measures the elongation of the regions, thereby elongated regions would be highlighted by the EAP. Similarly, regions of different homogeneity can be uncovered from their neighboring structure in an EAP associated with the "standard deviation" attribute filter.

Given an unknown image, which is difficult to apply *a priori* knowledge on the shape and texture information in the image content, one can simply stack the EAP profiles generated by all four attributes. The tested hyperspectral data are chip images from RIT SpecTIR Hyperspectral Airborne Experiment (SHARE) 2012 data campaign [93]. We provided quantitative comparisons between our proposed spatial-spectral self-tuning clustering method with comparison to ISODATA, K-means, Gaussian mixture model and Agglomerative clustering as shown in Fig. 4.13. In the following comparisons, we only adopted two of the composite kernel approaches: tune with EAP graph and weighted summation, which showed superior performance compared to the other composite kernel approaches.

We have manually selected seven region of interests (ROIs) as the reference data to quantitatively evaluate the clustering performance. As shown in Fig. 4.13 and Table 4.2, our self-tuning graph based clustering method performs better than the other four clustering methods by merely using spectral information with an overall accuracy of 89.38%. Furthermore, the two composite kernel methods both displayed superior performances by dramatically improving the overall accuracy with spatial information incorporated.



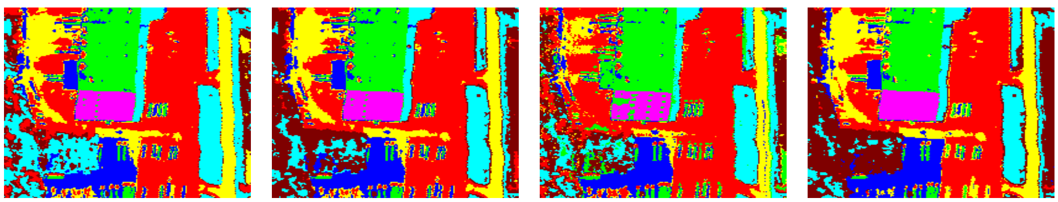
(a) our method (spectral only) (b) ISODATA (c) K-means (d) Gauss Mixture (e) Agglomerative

Figure 4.13: Clustering results comparison. Only spectral information is used in the clustering.

The proposed self-tuning spectral clustering method creates a graph representation for hyperspectral image data and efficiently encodes the intrinsic structure by amplifying connectivities in clusters and reducing connectivities between clusters. Hence, the accuracy of our proposed method based solely on spectral information outperformed the other clustering approaches. By including the spatial information with the use of composite kernel approaches for combining the graph matrices in EAP space, we have seen improvements in the overall accuracy as well as the per-class accuracies. Spatial information efficiently contributes to the spectral information by separating $roof1$ and

ROI	Our method	ISODATA	K-means	Gauss Mixture	Agglomerative
Ground1	1.0000	1.0000	1.0000	1.0000	1.0000
Roof1	0.4686	0.0331	0.0679	0.9146	0.3031
Ground2	1.0000	0.0000	0.0000	0.0324	1.0000
Road	0.9946	0.9973	0.9892	0.9866	0.9973
Grass	0.9861	0.9838	0.9861	0.9781	0.9919
Roof2	1.0000	1.0000	1.0000	0.0000	0.9846
Veg/Tree	1.0000	0.8683	0.8878	0.1073	0.9707
OA	0.8938	0.6782	0.6859	0.7095	0.8609

Table 4.2: Comparison of class accuracy and overall accuracy of each method.



(a) Tuning method (b) w/ conductivity matrix (c) Weighted summation method (d) w/ conductivity matrix

Figure 4.14: Clustering results using our tuning by EAP graph matrices approach and weighted summation method. Results of using conductivity matrix instead of affinity matrix are also included.

ROI	Tune (w/ affinity)	Tune (w/ conductivity)	Summation (w/ affinity)	Summation (w/ conductivity)
Ground1	1.0000	1.0000	1.0000	1.0000
Roof1	0.9948	0.9930	0.9983	0.9983
Ground2	1.0000	1.0000	1.0000	1.0000
Road	0.9973	0.9946	0.9543	0.9946
Grass	0.9919	0.9861	0.9792	0.9838
Roof2	0.9768	0.9768	0.8533	1.0000
Veg/Tree	0.9561	1.0000	0.9317	1.0000
OA	0.9913	0.9920	0.9707	0.9943

Table 4.3: Class accuracy and overall accuracy 1) comparison of using affinity matrix and conductivity with tuning method; 2) comparison of using affinity matrix and conductivity with direct summation method.

$roof2$ in the test image. Also, Fig. 4.14 and Table 4.3 illustrates the effectiveness *conductivity matrix*. It re-adjusts the graph by computing the *conductivity* flow between two data points based on the adjacency matrix, and shows its ability of further amplifying the intra-cluster proximity and produce promising results as well.

4.4 Summary

In this Chapter, we introduced feature mining and feature fusion techniques for HSI data analysis. Though HSI is captured by a single hyperspectral sensor which records the reflectance from earth surface over a range of wavelengths with high spectral resolution, the finer spatial resolution of the remote sensing systems enables the analysis of spatial structures in HSI by taking texture and contextual information into consideration. Feature mining, which includes feature generation, feature selection, feature extraction and etc, is a critical task for HSI data analysis. Though the original data consists of abundant spatial and spectral information, without proper feature mining, the original data space may not be the most effective space for representing the data points. Before selecting a subset of features or transforming the data into a lower dimensional space by feature selection or feature extraction (dimensionality reduction), feature mining could help alleviate redundancy and capture essential information.

The feature mining in spatial and spectral domain could help avoid ambiguous feature descriptors and exploit hidden information that can not be effectively used in the original data space. Particularly, without feature mining skills, spatial information can hardly be effectively utilized together with spectral features. With spatial neighborhood operations, texture features, shape information, and spatial proximities can be investigated as useful spatial information.

Manifold learning techniques use graph representation to model the data points in feature space. Those techniques have been explored for HSI dimensionality and clustering. Within this framework, an affinity matrix is created to encode the intrinsic structure of the input data points. Spatial and spectral features can be applied to discover the intrinsic structures and combined into an affinity matrix. Based on this idea, a self-tuning spatial-spectral clustering approach is proposed to simultaneously use spatial and spectral information contained in a HSI dataset. In the proposed approach, spatial information is included as the smooth term in similarity measure between every pair of vertices in a graph. Gaussian kernel is one of the widely used metric for characterizing similarities in an affinity matrix. However, though the tuning parameter plays a significant role in measuring the intrinsic local data structure, it is hard to predict its value. In this work, a self learning similarity measure based on graph is proposed by taking the local density into consideration to truly capture the cluster separability instead of merely relying on the Euclidean distance as similarity indicator. A graph-based RAG model is also introduced to reduce the over segmentation problem by directly applying the local structures learnt in the graph to recombine sub-clusters.

By adopting feature mining techniques, a spatial feature space can be constructed by extracting texture, shape, size information and stacking them into a feature vector for each data point. In the spatial feature space, every data point has a unique spatial feature descriptor that describes its local neighborhood. To combine the spatial and spectral feature space, a composite kernel approach is proposed for joint spatial-spectral clustering of HSI data. In this work, four composite kernel approaches were tentatively applied to fuse the HSI input space and EAP feature space in a unified self-tuning spectral clustering framework. An improved affinity matrix named *conductivity matrix* is adopted to further improve the performance of spectral clustering. It reinforces the connections within each cluster according to the *conductivity* flow between every pair of data points and amplifies the instrisin block-diagonal structure in the affinity matrix. The tested hyperspectral data are chip images from RIT SpecTIR Hyperspectral Airborne Experiment (SHARE) 2012 data campaign. Results have been generated to demonstrate that the proposed joint spatial-spectral clustering scheme outperformed the other clustering approaches that only use spectral features.

Chapter 5

Multi-feature Fusion with High-order Tensors

5.1 Tensors for Dimensionality Reduction

In HSI analysis, classification and segmentation usually have each pixel represented as a vector in original high-dimensional spectral space and solve the mathematical problem with linear algebra, i.e., the algebra of matrices. This data representation has one obvious shortcoming in that only spectral knowledge is utilized without neighborhood spatial information being exploited. While, tensors provide a natural data representation for data in multiway structure, such as a HSI. Considering the fact that a gray scale image is intrinsically a matrix or a second order tensor, the tensor representation provides a more sophisticated modeling technique and can be efficiently solved use multilinear algebra. Tensor-based image analysis has been explored in recent years. Success in tensor extensions to dimensionality reduction and noise reduction algorithms have been noticed in the literature [94, 95, 96, 97]. By considering the conventional linear analysis as a special case, tensor-based analysis offers a unifying mathematical framework to address a variety of image processing problems.

5.1.1 Tensor algebra

The order of a tensor is the number of dimensions, also called *ways*. A first-order tensor is a vector, a second-order tensor is a matrix, and a higher-order tensor consists of N vector spaces, which can be represented as $\mathcal{X} \in \mathbb{R}^{L_1 \times L_2 \times \ldots L_N}$, where the i-th order of the tensor is of size L_i, and each order is also called the i-th mode. Here, we briefly introduce several notations and definition of tensor operations in multilinear algebra:

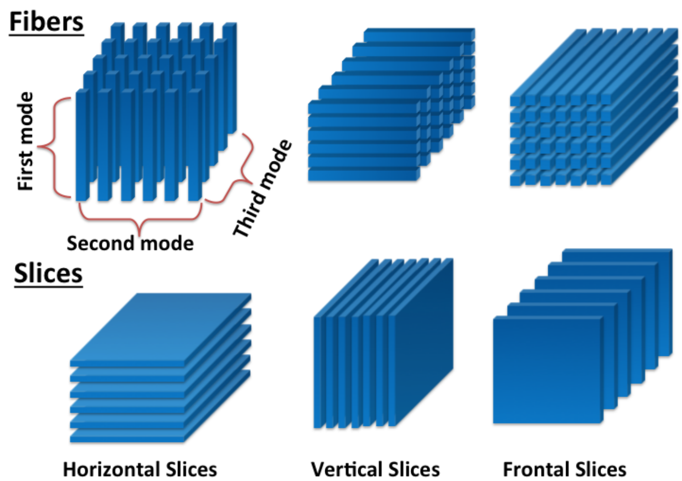


Figure 5.1: A simple illustration of fibers and slices in a third order tensor. Third-order tensors have column, row, and tube fibers; horizontal, lateral, and frontal slices.

1. **Fibers and Slices**: Fibers are the high-order analogue to row and column in a matrix. A mode-n fiber is defined by fixing every index but the n^{th} index. Slices are two-dimensional sections of a tensor, thereby they are defined by fixing every index but two indices.

2. **Inner product**: The inner product of two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{R}^{L_1 \times L_2 \times \ldots L_N}$ of the same dimension is defined as: $< \mathcal{A}, \mathcal{B} >= \sum_{i_1 \ldots i_N} \mathcal{A}_{i_1 \ldots i_N} \mathcal{B}_{i_1 \ldots i_N}$.

3. **Outer product**: The outer product of two tensors $\mathcal{A} \in \mathbb{R}^{L_1 \times L_2 \times \ldots L_N}$ and $\mathcal{B} \in$

$\mathbb{R}^{K_1 \times K_2 \times \ldots K_M}$ is given by:

$$\mathcal{C} = \mathcal{A} \circ \mathcal{B} \in \mathbb{R}^{L_1 \times L_2 \times \ldots L_N \times K_1 \times K_2 \times \ldots K_M}.$$

where:

$$c_{l_1, l_2, \ldots, l_n, k_1, k_2 \ldots, k_m} = a_{l_1, l_2, \ldots, l_n} b_{k_1, k_2 \ldots, k_m}.$$

As a special case, the outer product of two vectors $a \in \mathbb{R}^I$, $b \in \mathbb{R}^J$ yields a rank-one matrix $C = a \circ b = a \cdot b^T \in \mathbb{R}^{I \times J}$. The outer product for general tensors is also called the tensor product.

4. **Matricization**: Matricization of a tensor is also known as unfolding or flattening of a tensor. It is the process of reorganizing the elements in an N-th order tensor $\mathcal{X} \in R^{L_1 \times L_2 \times \ldots \times L_d \times \ldots L_N}$ into the form of a 2D matrix $\mathbf{X} \in \mathbb{R}^{L_d \times \overline{L_d}}$, $\overline{L_d} = \prod_{i=1, i \neq d}^{N} L_i$. In other words, the mode-d matricization of tensor $\mathbf{X}$ is to arrange the mode-d fibers in the columns of the resulting matrix.

5. ***k*-mode product**: To multiply a tensor by a matrix of mode-k is called the k-mode product. $\mathcal{X} \times \mathbf{U}$, where $\mathcal{X} \in \mathbb{R}^{L_1 \times L_2 \times \ldots L_N}$ and $\mathbf{U} \in \mathbb{R}^{L_k \times H (k=1,2,\ldots,N)}$, results in a tensor of size $\mathbb{R}^{L_1 \times L_2 \times \ldots L_{k-1} \times H \times L_{k+1} \times \ldots L_N}$.

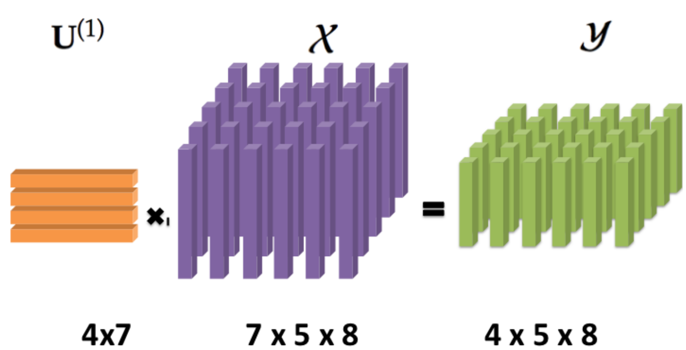


Figure 5.2: Illustration of the mode-n multiplications of a third-order tensor by a matrix. Mode-1 multiplication $\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)}$.

5.1.2 Tensor decomposition and factorization

Large amounts of data with multiple factors and high dimensionality are generated daily in all kinds modern applications nowadays. Tensors (i.e., multi-way arrays) provide a natural representation for such data. Tensor decompositions and factorizations were initiated in 1928 by Hitchcock [98]. Two of the most popular tensor decomposition models are the canonical polyadic (CP) model, also known as Parallel Factor (PARAFAC) analysis, and the Tucker model, respectively. Most of the early work devoted to tensor factorizations and decompositions appeared in the psychometrics and chemometrics literature. Although tensor decomposition models and analysis have been explored for a long time, they have recently attracted the interest of researchers from modern applications, such as mathematics, image analysis signal processing, neuroscience and etc. Two of the most commonly used decomposition models: Tucker and CP decomposition are often considered as higher-order generalizations of the matrix singular value decomposition (SVD) or principal component analysis (PCA).

Standard matrix factorization, such as PCA/SVD is a valuable tool for feature selection, dimensionality reduction, noise reduction and etc. Tensor decompositions are particularly promising for big and multi-way data analysis as multiple data can be analyzed in a unified framework that allows us to simultaneously explore the complex interactions across multiple factor components. Tensor decompositions have recently been explored in applications such as data compression, low-rank approximation, visualization, and feature extraction. To obtain meaningful and unique representations, additional constraints such as orthogonality, sparsity, and non-negativity constraints are often imposed on hidden factors and the core tensor. Non-negative constraint is necessary for certain applications such as chemical concentrations in experimental results or pixel intensities in digital images. Such decomposition is meaningful and have physical interpretation.

5.1.2.1 Tucker Model

Tucker decomposition [99], shown in Figure 5.3 for a 3-way case, is a basic model for high-order tensor decomposition that effectively allows for dimensionality reduction and feature extraction. Given a tensor data $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, Tucker decomposition can be

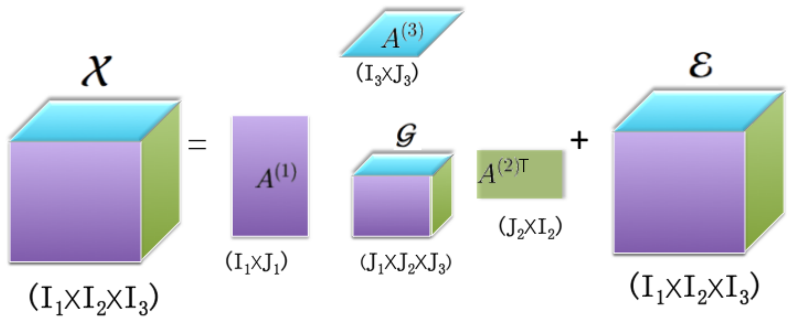


Figure 5.3: Illustration for 3-way Tucker decomposition. 3-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ is decomposed into three basis matrices $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}$ and a core tensor $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$, $J_1 \leq I_1, J_2 \leq I_2, J_3 \leq I_3$. Also, there is a term $\mathcal{G}$ denotes the approximation error.

formulated as follows:

$$\begin{aligned} \mathcal{X} &= \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} + \mathcal{E} \\ &= \sum_{p=1}^{J_1} \sum_{q=1}^{J_1} \sum_{r=1}^{J_1} g_{pqr}\, a_p^{(1)} \circ a_q^{(2)} \circ a_r^{(3)} + \mathcal{E} \end{aligned} \tag{5.1}$$

where $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ is called the core tensor of reduced dimension, and $\mathbf{A}^{(1)}, \mathbf{A}^{(2)}, \mathbf{A}^{(3)}$ are three component matrices representing common factors with different dimension in each mode. $\mathcal{E}$ denotes the approximation error. Given a tensor data $\mathcal{X}$ and three index numbers $\{J_1,\ J_2,\ J_3\} << \{I_1,\ I_2,\ I_3\}$, a core tensor $\mathcal{G}$ and three factor matrices can be obtained by performing tucker decomposition. By imposing non-negativity constraints to the decomposition, the problem of estimating the loading matrices and core tensor $\mathcal{G}$ becomes a generalized nonnegative matrix factorization problem called the Nonnegative Tucker Decomposition (NTD) [100].

It is noted the symbol $'' \times_n ''$, $n = 1, 2, 3$ denotes the n-mode product of tensor $\mathcal{G}$ with a matrix along the mode-n. The Alternating Least Squares (ALS) algorithm is the most widely adopted approach to solve Tucker decomposition. This Tucker decomposition model for 3-way tensor data illustrated in Fig 5.3 is also called the Tucker3 model because it is decomposed into three loading factors. A Tucker2 model can be obtained from the Tucker3 model by absorbing one factor by the core tensor, and a Tucker1 model can be obtained by absorbing two factor matrices by the core tensor.

5.1.2.2 CP Model

The CP model [101], which is similar to the singular value decomposition (SVD) of matrices, decomposes the a tensor into the sum of rank-one tensors. Given a 3-way data tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, and an index J, three loading matrices/factors $\mathbf{A}^{(1)} = [a_{11}, a_{12}, ..., a_{1j}] \in \mathbb{R}^{I_1 \times J}, \mathbf{A}^{(2)} = [a_{21}, a_{22}, ..., a_{2j}] \in \mathbb{R}^{I_2 \times J}, \mathbf{A}^{(3)} = [a_{31}, a_{32}, ..., a_{3j}] \in \mathbb{R}^{I_3 \times J}$ are combined as follows:

$$\mathcal{X} = \sum_{j}^{J} a_{1j} \circ a_{2j} \circ a_{3j} + \mathcal{E}. \tag{5.2}$$

where $a_{1j} \in \mathbb{R}^{I_1}$, $a_{2j} \in \mathbb{R}^{I_2}$, $a_{3j} \in \mathbb{R}^{I_3}$ are column vectors from the three loading matrices respectively. In fact, the CP model decomposes a given tensor into a sum of rank-one tensors.

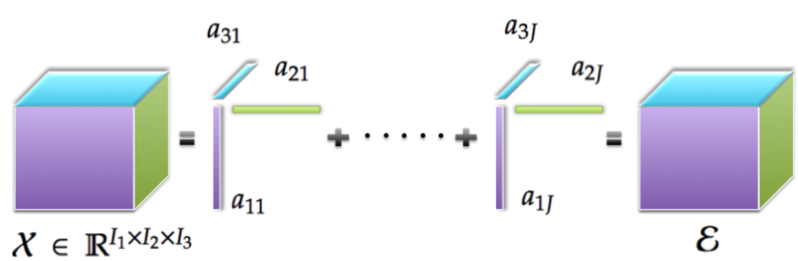


Figure 5.4: A graphical representation of the third-order CP decomposition as a sum of rank-one tensors $\mathcal{X} = \sum_{j}^{J} a_{1j} \circ a_{2j} \circ a_{3j} + \mathcal{E}$. Each vector a_{ij} is a column vector of the corresponding loading matrix.

CP decomposition sometimes fail to give compact and unique representation of multiway data. Non-negativity constraints are usually imposed on the factor matrices to obtain unique solutions. It is also named Nonnegative Tensor Factorization (NTF). The other way to interpret CP decomposition assumes that all vectors have unit length, so the CP model can be written as the weighted outer products of the column vectors in an alternative equivalent representations:

$$\mathcal{X} = \sum_{j}^{J} \lambda_j a_{1j} \circ a_{2j} \circ a_{3j} + \mathcal{E} \tag{5.3}$$

where λ_j is the scaling factor associated with each mode. By using the mode-n multiplication of a tensor by a matrix, the CP model can be written as:

$$\mathcal{X} = \Lambda \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} + \mathcal{E}. \tag{5.4}$$

where $\Lambda \in \mathbb{R}^{J \times J \times J}$ is a the core tensor with λ_j on the super-diagonal. This becomes a special case of Tucker decomposition, where the core tensor has non-zero elements only on the super-diagonal and with some mild constraints imposed.

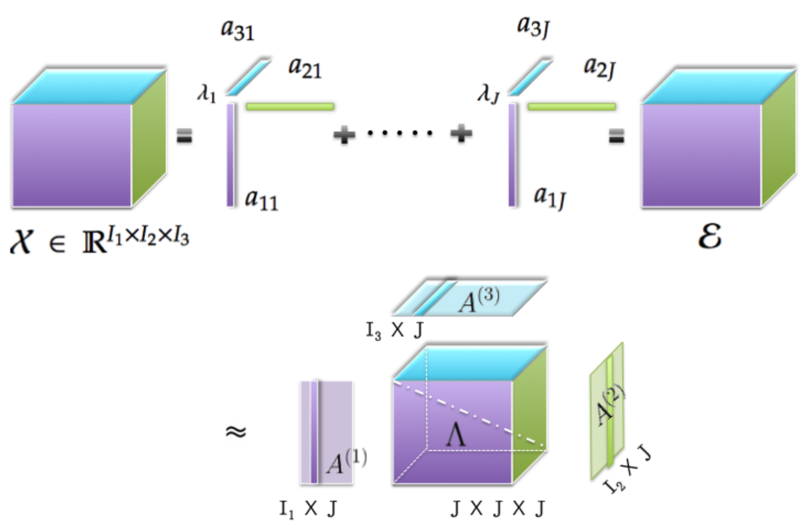


Figure 5.5: CP model as a special Tucker decomposition with a super-diagonal core tensor. In this model all vectors are normalized to unit length.

For conventional dimensionality reduction algorithms, such as Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), etc. the input data instances are represented by vectors of length l. The purpose of dimensionality reduction is to reduce redundant information contained in each vector so that the new vector representation for each data point is of length r, with $r < l$. In contrast to conventional vector-based representation, tensor-based representation represents an input data point as a 2D matrix or higher-order tensor. Tensor decomposition therefore aims to find the basis factors and coefficients of smaller sizes to efficiently represent the original data.

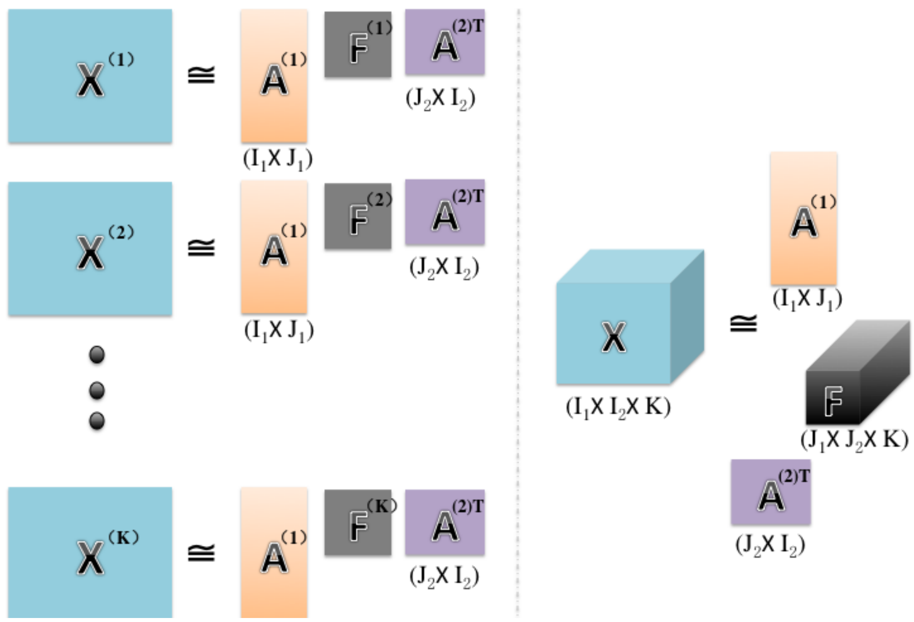


Figure 5.6: Left: simultaneous approximate matrix factorizations, given a set of second order tensors $\mathbf{X}^k \in \mathbb{R}^{I_1 \times I_2}$, $(k = 1, 2..., K)$. Right: Tucker-2 decomposition taken all the second order tensors as a single unified third order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times K}$.

In remote sensing research area, HSI and other multi-source data usually contains high dimensionality. The input data instance (pixel/super-pixel) may be described using a second order or higher order tensor as an extension to conventional vector representation. Similar to PCA which finds reduced-length vector representation for every input data point, simultaneous decomposition [102] of a set of input tensor instances can also generate a set of core tensors with reduced dimensionality. For example, given a set of K second order tensors (matrices) $\mathbf{X}^k \in \mathbb{R}^{I_1 \times I_2}$, $(k = 1, 2..., K)$, the objective is to perform simultaneous model reduction and to extract the core tensors as new features $\mathbf{G}^k \in \mathbb{R}^{J_1 \times J_2}$, with $J_1 << I_1, J_2 << I_2$, as in Figure 5.6. The simultaneous decomposition of the concatenated data set considers all input data instances all at once, and retrieves common factors within the whole data set. The extracted core tensor $\mathbf{G}^k$ contains the feature information of the interaction among basis components, in the subspace of $\mathbf{A}^{(n)}$. This is demonstrated in [102] to be equivalent to Tucker-N decomposition which concatenates a set of K input tensors $\mathbf{X}^k \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$, $(k = 1, 2..., K)$ as a unified single tensor, and obtain a reduced core tensor with the same dimension K along the mode of subjects.

5.1.3 Tensor subspace analysis

Tensor decomposition is an extension to SVD (PCA) of higher order data, which can be generalized for dimensionality reduction for tensorial data as described in the previous section. In linear algebra, a linear subspace (vector subspace) is a vector space that is a subset of some other (higher dimensional) vector space. Therefore, linear subspace learning algorithms, such as principal component analysis (PCA), independent component analysis (ICA), linear discriminant analysis (LDA), canonical correlation analysis (CCA) and etc., can be conveniently adopted for dimensionality reduction. Similarly, multilinear subspace (tensor subspace) learning are higher-order generalization to linear subspace learning that can be extended to solve for tensorial data dimensionality reduction problem.

Linear subspace learning algorithm reduces the input data dimensionality by first representing the input data as vectors and solving for an optimal linear mapping to a lower-dimensional subspace. When the input data is in multi-dimensional form, the linear subspace learning framework will break the natural structure and correlation within the original data by reshaping them into vector representations. Recently, researchers are showing interests in multilinear subspace learning (MSL), a new approach to dimensionality reduction that can preserve data in its natural multi-dimensional form.

Spectral methods are recently explored as a powerful tool for nonlinear dimensionality reduction and manifold learning for remotely sensed high dimensional data. By use of multilinear algebra, many manifold learning based DR algorithms can also be generalized to accept data directly from their tensorial representations. In multilinear subspace learning, input tensorial data are mapped to lower dimensional space through multilinear projections. Traditional linear dimensionality reduction algorithms, such as PCA, find a vector to vector projection (VVP) in a lower dimensional subspace. Such linear projections take a vector $x \in \mathbb{R}^I$ and projects it to another vector $y \in \mathbb{R}^J$ using a projection matrix $\mathbf{U} \in \mathbb{R}^{I \times J}$, $I << J$:

$$y = \mathbf{U}^T x = x \times_1 \mathbf{U}^T \tag{5.5}$$

Similarly, a high order tensor data can be projected to tensor subspace of the same order, named as tensor to tensor projection (TTP). N projection matrices $\mathbf{U}^{(n)} \in \mathbb{R}^{I_n \times J_n}$, $n = 1, 2, ..., N$, are generated to project a tensor data $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times ... \times I_N}$ to a lower dimensional tensor space $\mathcal{Y} \in \mathbb{R}^{J_1 \times J_2 \times ... \times J_N}$, where $J_n < I_n$ for all n, as the following:

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{U}^{(1)T} \times_2 \mathbf{U}^{(2)T} \times ... \times_N \mathbf{U}^{(N)T} \tag{5.6}$$

It is obvious that VVP is a special case of TTP when N equals 1. In multilinear subspace learning there are N sets of parameters to be estimated, one in each mode, and the results for one set often depends on the other sets. A suboptimal approach is usually adopted through an iterative procedure originated from the ALS algorithm. It solves for the projections by alternating between solving for one set of parameters at a time. At each iteration, by fixing the parameters in all other modes, the one mode can be efficiently solved through unfolding tensors into matrices use linear algebra. The parameter estimation process is sequentially and iteratively proceeded until convergence or a maximum number of iteration reaches. The general workflow of a multilinear subspace learning algorithm is shown in Fig 5.7.

5.2 Represent Multi-feature Remotely Sensed Data with High-order Tensors

The emergence of hyperspectral imagery has opened up new possibilities for image analysis and information extraction due to its capability of recording hundreds of spectral

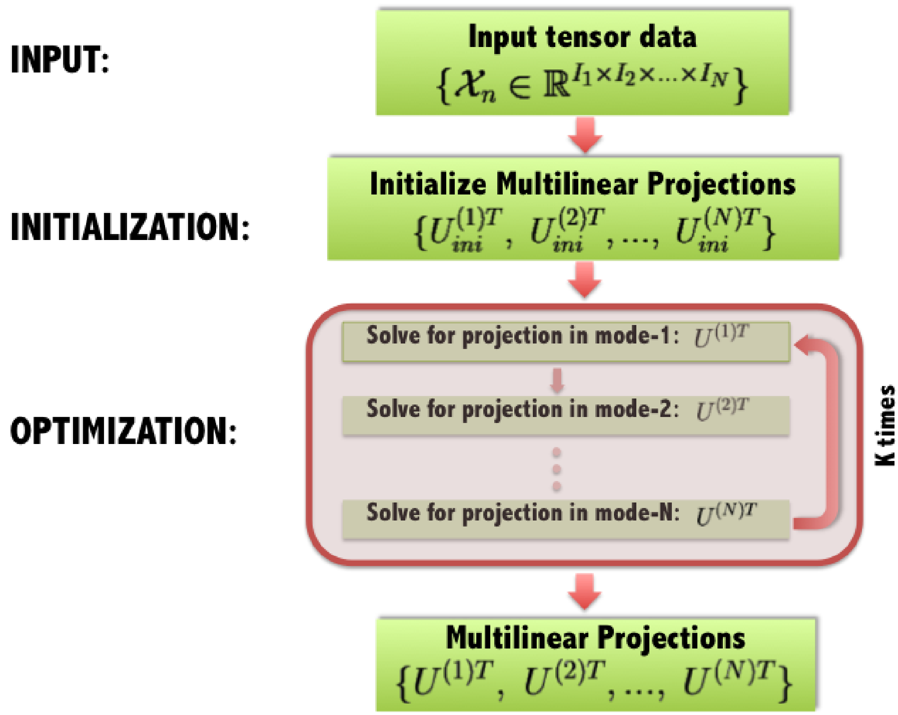


Figure 5.7: A simple illustration of a typical multilinear subspace learning algorithm workflow.

bands. In traditional applications, the input HSI is processed in a vector-based manner, which does not exploit the spatial correlations between neighboring pixels. Thereby it ignores the shape and texture information embedded in the image spatial structure. For each pixel in a HSI, the intensity over each spectral channel forms a spectral vector, which is the descriptor used in pixel-based feature extraction, classification, anomaly detection, and other applications. However, the HSI as a whole can be naturally represented as a 3-order tensor with the joint spatial-spectral dimensions. The analysis of a HSI, which is a set of D spectral images with spatial dimensions of size $W \times H$, is a problem in multilinear algebra. Inspired by the idea of representing an image cube as a tensor, pixel and superpixel can also be represented with tensors to include local neighborhood structure. In this chapter, we will discuss how to apply tensors in remotely sensed image processing.

5.2.1 Image cube as third-order tensor

HSI is a multi-dimensional data with a huge number of spectral bands. Tensor, as a higher order generalization to matrix, provides an intuitive representation of multi-dimensional data. A HSI cube can be conveniently represented as a third order tensor with three modes (width, height, and spectral bands) and discover the joint hidden spatial–spectral structures using multilinear algebra. Traditionally, a HSI is processed as a nonnegative matrix, $X \in \mathbb{R}^{N \times D}$, where N denotes the number of pixels and D denotes the number of spectral bands. With tensor modeling techniques, a HSI could be interpreted as nonnegative third order tensor: $\mathcal{X} \in \mathbb{R}^{W \times H \times D}$, where W and H denotes the width and height of the image. Therefore, instead of rearranging data into classical matrix-based linear algebra problems, the multilinear algebra provides a powerful mathematical framework for directly analyzing the multi-dimensional data. Based on Tucker decomposition, HSI image cube can be decomposed into a core image tensor, three component matrices and a reconstruction error term which reflects the difference between decomposed components and original HSI image cube. In Ref. [96], simultaneous denoising and dimensionality reduction of an input HSI were achieved under such tensor decomposition framework.

The idea behind taking a HSI cube as a single processing unit is to split a third order data, which is a superposition of signal and noise, use tensor decomposition techniques. Lower Rank Tensor Approximation (LRTA) algorithm takes a third order tensor data $\mathcal{X} \in \mathbb{R}^{W \times H \times D}$ as the input, and it finds a lower rank approximation to it with redundant information removed. Its Rank-(R_1, R_2, R_3) tensor approximation $\mathcal{B}$, with ($R_1 < W$, $R_2 < H$, $R_3 < D$), can be calculated by minimizing its quadratic Frobenius distance: $\|\mathcal{A} - \mathcal{B}\|$. $\mathcal{B}$ can be expressed with Tucker model:

$$\mathcal{B} = \mathcal{D} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)} \tag{5.7}$$

The best lower Rank approximation of $\mathcal{A}$ is then given by:

$$\mathcal{B} = \mathcal{A} \times_1 P^{(1)} \times_2 P^{(2)} \times_3 P^{(3)} \tag{5.8}$$

with $\mathbf{P}^{(n)} = \mathbf{U}^{(n)}\mathbf{U}^{(n)T}$, the orthogonal projector on the subspace generated by the $R_n I_n$-dimensional column vectors of $U^{(n)}$, in the n-mode initial vector space [**?**].

The LRTA algorithm extracts R_3 spectral components from the original tensorial data $\mathcal{X} \in \mathbb{R}^{W \times H \times D}$ to get the output tensorial data $\mathcal{Y} \in \mathbb{R}^{W \times H \times R_3}$. By applying the orthogonal projectors $\mathbf{P}^{(1)}$, $\mathbf{P}^{(2)}$ while decompose the data only along the spectral mode, the input HSI cube is simultaneously denoised and truncated [96].

5.2.2 Spatial-spectral feature fusion for HSI

Since the emergence of multimodal remote sensing techniques, feature fusion has been more and more important for remotely sensed data processing. For classification and clustering task, it is desirable to combine multiple feature descriptors to improve the performance because different features may contain complementary information to fully describe the image content. The simple combination of multimodal data result in an increase in noise and dimensionality, hence effective feature fusion becomes an essential step. Features in different domain could be on different scales. Besides, a set of bad features may deteriorate the performance of a good classification algorithm. These issues make feature fusion a challenging task.

Conventional feature fusion methods usually concatenate various features into a single long vector. Despite the simplicity of such methods, the combined vector may not perform better than using a single feature. This is because the information conveyed by different feature descriptors is not equally measured. A simple concatenation may cause the long feature vector significantly unbalanced. Besides, concatenation may lead to curse of dimensionality if the feature vector is large.

A more careful feature weighting according to their level of importance is proposed to avoid one or two features may dominate the entire system performance. The simplest way is to normalized each feature to the same scale so that the combined vector could be more balanced. However, this feature weighting scheme does not help with the curse of dimensionality problem. And in many cases, different features that are extracted from the same datum are very correlated with each other. Therefore, the simple combination scheme is not sufficiently helpful for pixels or object discrimination.

An alternative way to overcome the shortcomings of simple feature vector concatenation method is feature selection. This is to combine the features to generate a smaller but more effective feature set. One of the common feature selection method is to find a linear combinations of all features. If $\mathbf{x}$ is the original d-dimensional feature vector and $\mathbf{W}$ is a d-by-m transform matrix with $m < d$, then the new m-dimensional feature vector $\mathbf{y}$ is given by:

$$\mathbf{y} = \mathbf{W}^{\mathbf{T}}\mathbf{x} \tag{5.9}$$

The elements in the transformation matrix $\mathbf{W}$ are the coefficients of the linear combination. A frequently adopted approach to this problem is to use PCA to find such a transform matrix by maximizing the variances. More generally, subspace learning is the tool to jointly combine and select a subset of features to provide a more effective and concise representation of them. PCA is one of the most well-known linear subspace learning techniques that have been widely used in image processing. Subspace learning refers to the transform of the original input features to a lower dimensional subspace. It includes linear subspace learning, nonlinear subspace learning, multilinear subspace learning and etc. By forming multiple features into a vector, linear and nonlinear subspace learning techniques seek a low dimensional subspace where the correlation conveyed in the original feature spaces is preserved.

Multilinear algebra provides the possibility to merge multiple features along different modes into a higher order tensor. By adopting multilinear subspace learning, we are able to fuse multiple features and reduce dimensionality at the same time. It is also capable of dealing with the computational difficulty introduced by existing feature fusion methods when the number feature is large. The tensor structure, which contains complex relationships between different features, may also introduce more powerful properties to bring respective feature spaces together so as to boost the discriminating power. Besides, tensor can learn more compact representations than its linear counterpart. It needs to estimate a much smaller number of parameters and it has fewer problems in the small sample size scenario [103].

In remotely sensed image, the spatial resolution refers to the area of the ground captured by a single pixel. With the advances in hyperspectral remote sensing, HSI may contain both high spatial and spectral resolution. The high spatial resolution is helpful to locate the objects with better accuracy, and the high spectral resolution can be utilized to identify the materials. Useful information can be extracted from the spatial domain, such as the size and the shape of the structure to which one pixel belongs, can also provide the potential to reduce the clustering uncertainty that exits when only spectral information applies.

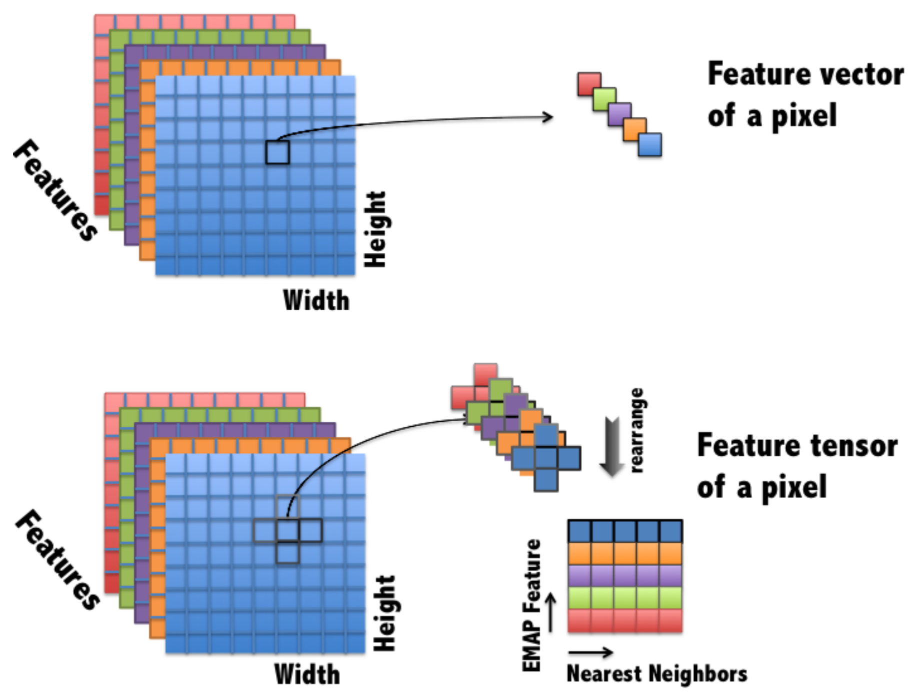


Figure 5.8: Tensor representation for pixels in a high dimensional image (HSI/EMAP). Comparison between a conventional vector representation for a pixel and a second order tensorial representation with four spatial neighbors considered.

As mentioned in Chapter 4, various spatial feature mining techniques have been proposed and applied to HSI. Among the spatial feature mining approaches, the extended morphological attribute profiles (EMAP) associated with a HSI is also a multi-dimensional image that each layer is a grey-level image filtered with a specific attribute filter, which encodes the shape and textural information such as area, bounding box, moment of inertia and standard deviation of the structure to which one pixel should belong. Every pixel in this spatial feature space is described by a feature vector that contains both spatial and spectral information that can be utilized as an effective descriptor for image content understanding.

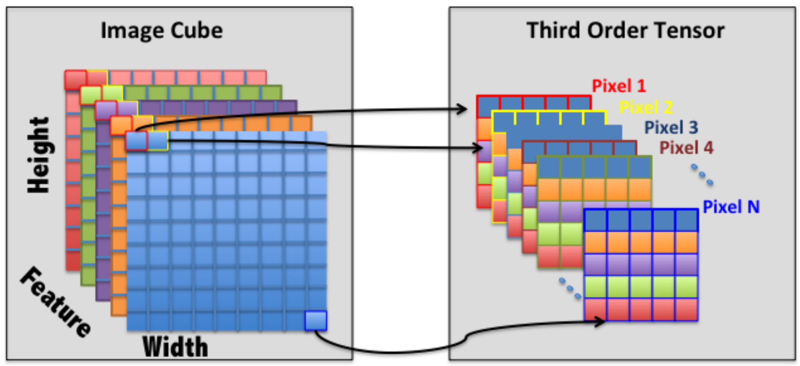


Figure 5.9: The second order tensorial representations for all pixels in an image cube are concatenated along the third mode to form a third order tensor.

Moreover, pixels in a local spatial neighborhood around a center pixel usually share common spatial and spectral patterns. The combination of one pixel's local neighborhood may contribute to local consistency and smoothness during classification and clustering. Direct neighbors are included in a tensor representation to include the local spatial structures of the centric pixel.

Therefore, we fuse the spatial and spectral information extracted by feature mining techniques by arranging the EMAP feature vectors and the direct local neighbors into a second order tensorial representation: $\mathbf{X} \in \mathbb{R}^{k \times d}$, where k is the number of spatial nearest neighbors to a centric pixel and d is the dimensionality of EMAP feature space. Fig 5.8 shows detailed data organizational structure of vector and tensor representations. Our goal is to utilize the relationship between these spatial and spectral features for better classification performance and to lower the computational burden. A third order tensor can be formed by concatenating all pixels along the third mode, as shown in Fig 5.9. Therefore, tensor decomposition and subspace analysis techniques can be conveniently applied to the tensor data and be used as an efficient tool to perform feature selection and dimensionality reduction. Also, by preserving as many as possible the original spatial-spectral structures, tensor representation helps to reduce the number of parameters to be estimated in a projection in linear or multilinear dimensionality reduction. In other words, it will be more computational efficient by forming local spatial structures and extracted spatial-spectral features into tensorial representation.

5.2.3 Spatial-Spectral-LiDAR feature fusion with high order tensor

HSI is a very important tool in remote sensing for its capability of providing both high spectral and spatial resolution of the scene. By exploiting the spatial correlation between neighboring pixels, the discriminating power in the HSI data can be further improved by extracting spatial texture and shape knowledge.

Despite the fast development in spatial-spectral classification of HSI data, a single imaging sensor cannot provide comprehensive knowledge of the scene being captured. LiDAR sensor delivers three dimensional point clouds with the intensities of the returned signals. Unlike cameras, LiDAR sensor does not require light to capture data. It provides rich information on the vertical structures that could be exploited to help distinguish objects with similar spectral signature and horizontal structure.

The fusion of LiDAR data and HSI has been investigated for a variety of applications recently, ranging from generation of Digital Surface Model (DSM)/Digital Elevation Model (DEM), 3D object recognition/extraction, forest boundary detection, ground object modeling and mapping, and etc. LiDAR data has also been used for land cover classification. The inclusion of LiDAR data contributes to the differentiation of classes that have similar spectral characteristics. With a single source of spectral data, it cannot differentiate objects composed of the same material, such as roofs and roads both made of concrete. On the other hand, LiDAR data source only contains geometric structures and may not be used to identify materials. Therefore, the data fusion of HSI and LiDAR data may provide more reliable image interpretation.

The abundant spatial, spectral and geometric information embedded in LiDAR and HSI data sources facilitates image classification by providing higher order attributes (features), which enhance the description of the data points. A LiDAR point cloud is defined as a set of points $P_i, i = 1, ..., n$, embedded in three-dimensional Cartesian space.

To utilize LiDAR data with HSI, a georeferenced point cloud is given in an earth-fixed coordinate system. Each point P_i has three coordinates, $(x_i, y_i, z_i)^T \in \mathbb{R}^3$, where (x_i, y_i) is the pixel location as in a HSI, and z_i indicates the elevation. Higher level features of LiDAR data points can either be measured directly, calculated within a local neighborhood, or computed in combination with other source of data.

The first level of features obtained from LiDAR point clouds for individual data points are the geometric locations recorded by the sensor. The vertical location, i.e., the height, provides valuable and complementary information to HSI image interpretation. Some other attributes like intensity, number of returns, colors may also be assigned to each data point and can be used as the first level features. The second level of features refer to the features computed from the neighborhood of a point that provide surface properties, such as surface normals and curvatures.

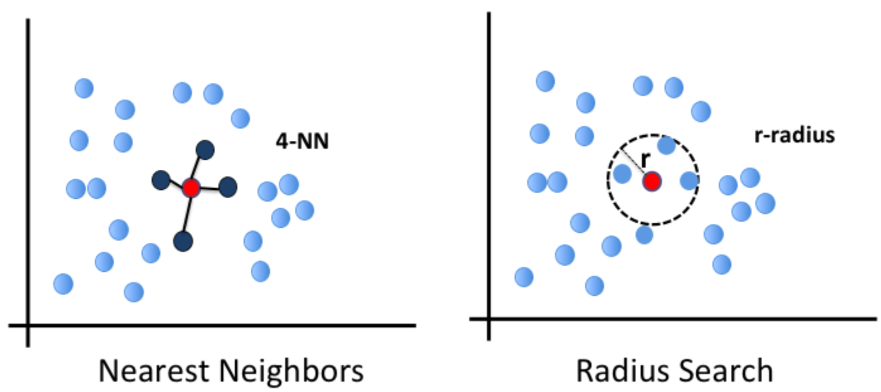


Figure 5.10: Two types local neighborhood searching: k nearest neighbor searching ($k = 4$ in the figure) and radius searching (r is the radius in the figure).

5.2.3.1 Local analysis and feature extraction

The neighborhood of a center pixel P_i is a subset of the point clouds defined by the spatial relationships to it. The rule of finding a neighborhood of a data point is to detect a number of neighbors that is sufficient to represent a small surface patch for local analysis. In other words, the local neighborhood $N_i = \{P_1, P_2, ..., P_k\}$ depends mainly on the three dimensional geometric locations to the center pixel P_i. Points are considered in the neighborhood either if their distance to P_i is below a certain threshold or if they belong to the k nearest points to P_i, as seen in Fig. 5.11. In either criterion, the distance between data points must be measured in the first place. Intuitively, k-nearest neighbor method finds k nearest neighbors according to Euclidean distance and identifies them as the local neighborhood to a data point. Many algorithms have been proposed to search nearest neighbors based on different auxiliary indexing data structures, such as $kd-$tree and box-decomposition trees. The ATRIA algorithm adopted triangle inequality which further optimizes and speeds up the nearest search strategy. The fixed radius searching method is a variant of the nearest neighbor search problem. The size of the neighborhood relies on the density of points, so the number of neighbors may vary around different data points. The optimal size of neighborhood for computing features is often empirically chosen. Methods for determining the optimal neighborhood size for normal and curvature feature estimations have been studied in the literature [104].

After local neighborhood of a data point is determined, surface properties such as normal vector and curvature can be estimated using statistical analysis. Surface normal vectors indicate the orientation of the surface the data point reflected off. It is usually estimated by fitting a plane that minimizes the Euclidean distance of the neighboring points to the plane. The problem of estimating the normal vectors is approximated by the problem of finding a virtual plane tangent to the surface, which becomes a least-square plane fitting problem. The normal vector estimation problem is then reduced to a numerical analysis problem. Eigen analysis of the covariance matrix formed by the data points in its local neighborhood is performed to find the normal vector.

Let $\overline{P}$ be the 3D centroid of the k nearest neighbors $\{P_1, P_2, ..., P_k\}$. A covariance matrix C_i is formed as the following:

$$C_i = \frac{1}{k}\sum_{i=1}^{k}(P_i - \overline{P}) \cdot (P_i - \overline{P})^T \tag{5.10}$$

The eigenvectors $\vec{e_i}$ and eigenvalues λ_i can be calculated as:

$$C_i \cdot \vec{e_j} = \lambda_i \cdot \vec{e_j},\ j = 0, 1, 2 \tag{5.11}$$

PCA can be used to find a set of orthogonal basis that best approximate a given data set. Therefore finding the normal vector $\vec{n} = \{n_x, n_y, n_z\}$ is equivalent to performing a PCA of the covariance matrix C_i and choosing the principal component with the smallest covariance [105].

Curvature is a measurement related to object local shape. It is the second derivative of a surface. A positive curvature indicates the surface is upwardly convex at a point location. A negative curvature indicates the surface is upwardly concave. A value of zero indicates the surface is planar. Based on the definition, we know that the curvature of data points located on vegetations have higher absolute value. While those on building surfaces have relatively small value in curvature and change rate. This is helpful to distinguish manmade objects, such as buildings, roads, bridges, and etc., from natural land covers, i.e., trees, soil slopes, and etc.

For a data point P_i, given the surface normal $\vec{n}$, and $\vec{u}$, $\vec{v}$, the orthogonal unit vectors in the tangent plane to the surface at P_i, the curvature is defined in terms of the 2×2 second fundamental form in the surface coordinate frame defined by $\vec{u}$, $\vec{v}$:

$$\mathrm{I\!I} = (du\ dv) \begin{bmatrix} A & B \\ B & C \end{bmatrix} \begin{pmatrix} du \\ dv \end{pmatrix} \tag{5.12}$$

where A, B and C are defined in terms of derivatives of the normal vector $\vec{n}$:

$$A = -\frac{\partial \vec{n}(x, y)}{\partial x} \cdot \vec{u} \tag{5.13}$$

$$B = -\frac{\partial \vec{n}(x, y)}{\partial x} \cdot \vec{v} = -\frac{\partial \vec{n}(x, y)}{\partial y} \cdot \vec{u} \tag{5.14}$$

$$C = -\frac{\partial \vec{n}(x, y)}{\partial y} \cdot \vec{v} \tag{5.15}$$

The eigenvalues of the matrix $\mathrm{I\!I}$ are called the principal curvature values κ_1, κ_2. Their product $\mathbb{K} = \kappa_1 \kappa_2$ is called the Gaussian curvature [106].

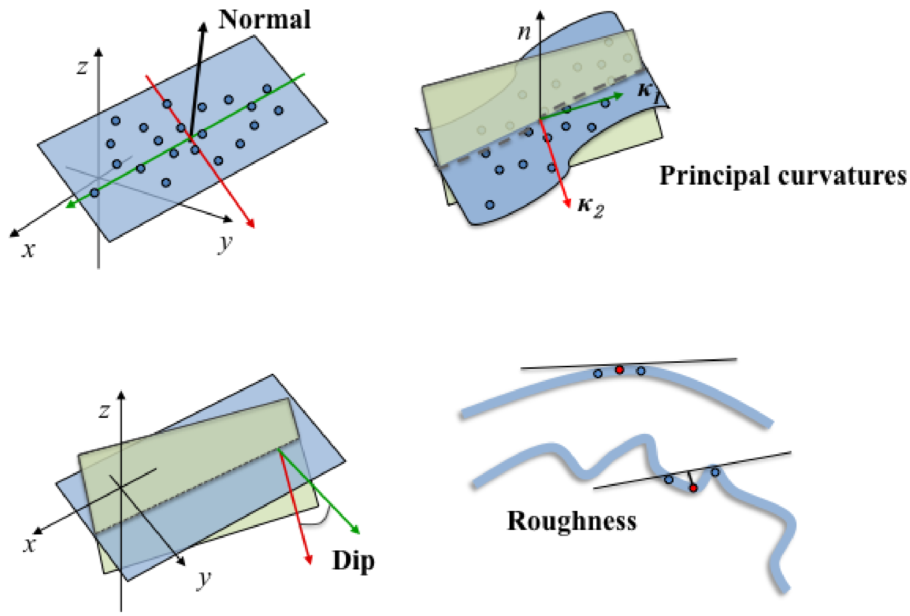


Figure 5.11: Simple illustration of some of the LiDAR point cloud local features.

The surface variation of a data point P_i can be used as a a curvature indicator (flatness), which is defined as:

$$\delta_{p_i} = \frac{\lambda_0}{\lambda_0 + \lambda_1 + \lambda_2} \tag{5.16}$$

The open-source Point Cloud Library (PCL) also provides methods for approximating the curvature value for each data point based on a PCA study of the point normals of a surface patch in the tangent plane of the given data point. It returns the approximate principal curvature (eigenvector of the maximum eigenvalue), along with the maximum and minimum eigenvalues as the principal curvatures (κ_1, κ_2).

Dip is a geologic feature which measures the steepest angle of descent of a surface relative to a horizontal plane. Given the normal vector of a data point, its Dip is measured as 90° minus the angle between $\vec{n}$ and $\vec{n}_{proj}$, where $\vec{n}_{proj}$ is the projection of $\vec{n}$ onto the xy-

plane.

Other useful local features include the volume density and roughness. To compute the volume density for P_i, we first look for its nearest neighbor and consider that P_i is inside a ball with radius equal to the distance between the two points. The density is simply defined as 1/volume(ball(r)), with unit pts/m^2. The roughness is equal to the distance between a data point P_i and the best fitting tangent plane computed from local neighborhood.

5.2.3.2 Fusing features with tensors

In order to capture comprehensive information from the scene, we utilized three sets of features separately representing spatial, spectral and geometric attributes derived from HSI and LiDAR sensors. As mentioned in the previous section, to concatenate all the different feature vectors into a unified long vector representation usually lead to high dimensionality problem and result in poor computational performance. Tensors provide a convenient representation for multiple features by arranging information along different modes. The tensor structure, for high-order feature patterns, can jointly reduce dimensionality and extract compact features of the fused feature ensemble by factor decomposition and subspace learning. For one data point P_i, feature mining algorithms are applied to separately extract spatial, spectral and geometric feature vectors from HSI and LiDAR data source. The three set of features $\{f_s, f_p, f_l\}$ of P_i are formulated into a second order tensor (matrix) $F_i \in \mathbb{R}^{l\times m}$, where l , m are the length of individual feature vector and number of features ($m = 3$ in this case), respectively. Its local neighborhood N_i, with size n, is important for deriving smooth and consistent classification result. Hence, we combine F_k generated for data P_k, with $k \in N_i$, and arrange them along the third mode to form a third order tensor $\mathcal{F}_i \in R^{l\times m\times n}$.

5.2.4 Superpixel represented with high-order tensor

Recently, the predominantly pixel-spectra based method for remotely sensed image analysis has evolved to multi-scale object-based contextual model which is more similar to the way that humans interpret images. Object-based methods can make better use of spatial information implicit within remotely sensed imagery. In its most fundamental level,

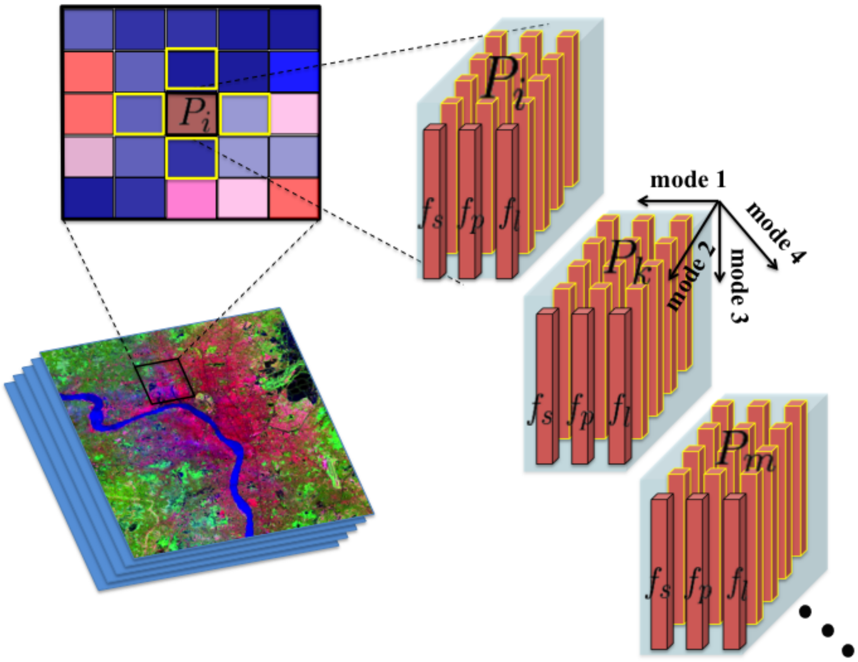


Figure 5.12: A HSI image cube is represented by the concatenation of the 3^{rd} *order* tensor representations of all pixels. The spatial, spectral and geometric features for each pixel is represented as a matrix. With local neighborhood information included, a 3^{rd} *order* tensor is formulated to fuse HSI and LiDAR features.

object-based methods require image segmentation as the first step for building objects or sub-objects. An alternative way is to generate superpixels, which has been widely used in the area of computer vision. A superpixel is a small and coherent cluster that contains several pixels of similar attributes and are spatially adjacent. Superpixels are a form of segmentation which focus more on obtaining over segmented regions rather than meaningful objects. Superpixel-based techniques can reduce the influence of noise and improve the computational speed. To represent a superpixel, instead of calculating the mean feature vector of all pixels in it, one can use tensors to include the abundant information included in a superpixel.

The Simple linear iterative clustering (SLIC) [107] superpixel algorithm constructs a feature vector for each pixel using its spatial location (x, y) and its color/spectral vector $\mathbf{I}(x, y)$. The superpixel generation process of SLIC is similar to k-means clustering which starts with an initial set of cluster centers then gradually updates the cluster centers and pixel labels until criteria are satisfied. With its parameters S and m which control the size and regularity of superpixels, superpixels are iteratively optimized based on the distance metrics defined in the feature space. Different to k-means clustering, SLIC only searches cluster centers within a $2S \times 2S$ neighborhood which speeds up the convergence. SLIC has the potential of producing very regular and compact superpixels for high dimensional images. The superpixels as the input to dimensionality reduction and image classification results in great improvement of the computational efficiency because the number of superpixels are much fewer compared to the quantity of original pixels, although at some cost of lost accuracy especially when details are smaller than the size of a superpixel. A superpixel is an atomic image patch that contains both two-dimensional spatial correlations and spectral information. However, in HSI processing, a superpixel is usually represented with its mean or centroid spectral vector which loses spatial structures and results in a further loss of accuracy in post processing. To the contrary, tensors could be conveniently combined with SLIC for DR and classification of high dimensional image where the input superpixels are formularized in their natural multi-dimensional form as tensors.

In the previous section, tensor representation of a data point utilized the spatial neighborhood around its pixel location as well as the spectral signatures. Similarly, superpixels divide an image plane into non-overlapping atomic patches. Pixels that belong to a superpixel are spatially related and share similar color or texture distributions. Therefore, tensor representation can be adapted to represent superpixels. Compared to a single pixel which carries very little perceptual meaning, superpixel contains both spatial correlations and spectral features. Besides, it reduces the computational complexity of the images by dropping hundreds of thousands of pixels to only a few hundred superpixels. Instead of representing each superpixel as a vector which lost most of the contextual relationships among pixels, tensor representation could be conveniently incorporated into the superpixel image processing framework without losing the important spatial correlations. The spatial neighborhood structures are well preserved and fully utilized with tensor representation for superpixel analysis.

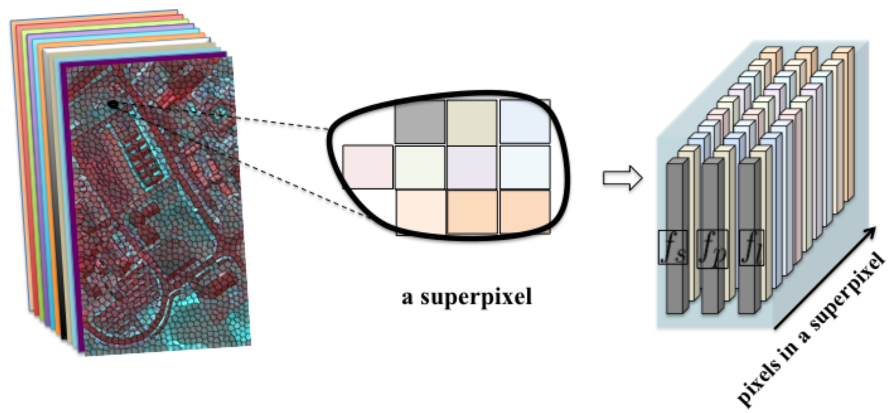


Figure 5.13: Features extracted from a superpixel can be formulated into high order tensor representation.

5.3 Tensor-based Dimensionality Reduction Algorithms

Multilinear algebra is a powerful tool to handle the multi-dimensional remotely sensed data. Tensor decomposition and multilinear subspace learning have been used for feature extraction (dimensionality reduction) by capturing multilinear data structures in higher-order dataset. Traditional feature extraction algorithms such as PCA, treat the data as matrices or vectors and are often not computationally efficient. Since the curse of high dimensionality is often a major issue of many practical applications, dimensionality reduction is a necessary step to classification, data mining, pattern recognition, and etc. In this thesis, tensor decomposition and multilinear subspace learning algorithms were introduced to model and process remotely sensed high-order data for feature fusion and dimensionality reduction.

5.3.1 Tensor decomposition and low-rank approximation

In HSI, every pixel contains a spectra feature vector and all together form into a three dimensional image cube, or a third-order tensor, with the first two dimensions indicating the spatial domain and the third dimension indicating the spectral domain. To avoid the rearranging step to vectorize the image cube as a two-way data for traditional vector-based processing algorithms, tensor decomposition can be used to directly decompose a high-order image cube use multilinear algebra. Renard et al. [96] proposed the Lower Rank Tensor Approximation (LRTA) method that takes the image cube as a third order tensor to simultaneously reduce the spectral dimension and jointly denoise all ways.

LRTA

LRTA algorithm was proposed by Nadine Renard et al. [96] which directly models the three-dimensional HSI cube as a three-way tensor and adopts the TUCKER3 decomposition to reduce the input tensor to a lower rank approximation. The LRTA algorithm takes into account the cross-dependency of information contained in spatial and spectral modes. With Tucker decomposition directly applied on the input image cube, it captures intact multi-way structure to derive the n-mode projectors jointly.

There are two spatial modes for rows and columns and one mode for the spectral bands in a HSI tensor. TUCKER3 decomposes the HSI cube $\mathcal{X}_{HSI} \in \mathbb{R}^{width \times height \times bands}$ into a core tensor with reduced dimensions $\mathcal{G}_{HSI} \in \mathbb{R}^{J_1 \times J_2 \times J_3}$ ($J_1 \leq width, J_2 \leq height, J_3 \leq bands$), and three factor matrices $\mathbf{A}^{(1)} \in \mathbb{R}^{width \times J_1}$, $\mathbf{A}^{(2)} \in \mathbb{R}^{height \times J_2}$, $\mathbf{A}^{(3)} \in \mathbb{R}^{bands \times J_3}$, written as:

$$\mathcal{X}_{HSI} = \mathcal{G}_{HSI} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \mathbf{A}^{(3)} + \mathcal{E} \tag{5.17}$$

$\mathbf{A}^{(n)}$ (n=1,2,3) is the orthogonal n-mode matrix holding the J_n eigenvectors associated with the J_n largest eigenvalues of the flattened matrix $\mathbf{X}_n$. Renard proposed to use new matrices generated from $\mathbf{A}^{(n)}$ to keep the size of the two spatial dimensions unchanged while only the dimensionality along the third mode will be reduced:

$$\mathbf{P}^{(n)} = \mathbf{A}^{(n)} \mathbf{A}^{(n)T} \tag{5.18}$$

$$\mathcal{Y}_{HSI} = \mathcal{X}_{HSI} \times_1 \mathbf{P}^{(1)} \times_2 \mathbf{P}^{(2)} \times_3 \mathbf{\Lambda}^{-1/2} \mathbf{A}^{(3)T} \tag{5.19}$$

where $\mathbf{\Lambda}$ is the diagonal eigenvalue matrix holding the J_3 largest eigenvalues associated with $\mathbf{A}^{(3)}$. Therefore, only the number of bands in the input HSI is reduced to size of J_3. The classification result depends not only on the number of extracted spectral features J_3

but also on the dimension of spatial subspaces J_1, J_2.

Given remotely sensed data, local features such as shape, texture, and geometric descriptors can be individually extracted for every data point. Tensor representation is also a convenient tool to fuse and model all the features, as well as keep their multi-way structure, for every individual data point. Tucker and CP are two of the most popular tensor decomposition models. By aligning the tensor representations for all data points along one mode, a high order tensor representation for the entire dataset is constructed and can be analyzed with tensor decomposition techniques to replace the original data by a lower dimensional approximate representation obtained via a simultaneous factorization of all data points.

Because the image data are usually nonnegative, it is preferable to take the nonnegative constraint into account in the analysis to extract factors with reasonable interpretation. With the nonnegative constraints, Tucker and CP decomposition become the Nonnegative Tucker Decomposition (NTD) and Nonnegative Tensor Factorization (NTF), respectively [108].

NTF

NTF is a multi-way decomposition method, as the extension to nonnegative matrix factorization (NMF). It finds the basis factors of the dataset based on linear transform in terms of CP model and assumes that the elements are all nonnegative. In multi-feature applications, the tensor representation of each data point could be of any dimensions, depending on the number and the length of individual features. If a data point is represented as a s-way tensor by fusing multiple features along different modes, then all the data points will be concatenated and formed into a $(s+1)$-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{(s+1)}}$.

As introduced in the previous section, for an input data $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{(s+1)}}$, the NTF model can be formulated as:

$$\mathcal{X} = \sum_{j}^{J} a_{1j} \circ a_{2j} \circ \ldots \circ a_{(s+1)j} + \mathcal{E}.$$

where J is the size of each decomposed subspace. Each vector $\mathbf{a}_{ij}$ is a column vector of the corresponding loading matrix. In the form of tensor products, this NTF model can also be described as:

$$\mathcal{X} = \Lambda \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_. .. \times_{(s+1)} \mathbf{A}^{(s+1)} + \mathcal{E} \tag{5.20}$$

where $\mathbf{\Lambda} \in \mathbb{R}^{J \times J \times \ldots \times J}$ is a the identity core tensor. Each factor matrix $\mathbf{A}^{(i)} = [\mathbf{a_{i1}}, \mathbf{a_{i2}}, ..., \mathbf{a_{iJ}}]$ explains the input tensor data along one mode. Each row in the factor matrix can be considered as the extracted feature vector of the input data onto the subspace spanned by the others. Therefore, the first s factor matrices contain extracted features along one mode separately, while each data point is characterized by each row of $\mathbf{A}^{(s+1)}$.

Let the extracted and shortened feature vector of each data point combines as a feature matrix $\mathbf{F} \in \mathbb{R}^{I_{(s+1)} \times J}$. The dimensionality reduction problem for the input $(s+1)$-way remotely sensed tensor data, where all data points are arranged along the $(s+1)$-mode, then can be formulated as a NTF decomposition problem by identifying the factor matrix along the $(s+1)$-mode as the following:

$$\mathcal{X} = \Lambda \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_. .. \times_{(s+1)} \mathbf{F} + \mathcal{E} \tag{5.21}$$

The last factor matrix $\mathbf{F}$ consists of the extracted reduced features in vector representations onto multi-domain feature spaces, such as spatial, spectral, geometric and etc., depending on what features were fused and formed into the s-way tensor representation of each input data point. to interact with each other

NTD

In contrast with NTF model, which decomposes the input tensor data into rank-one tensors, NTD [109] is based on Tucker model that decomposes a tensor into a core tensor multiplied a matrix along each mode with non-negativity constraints on them. In other words, NTD is a special case to Tucker model. By using a multiplicative updating algorithm, NTD iteratively matricizes a tensor along each mode and then solve nonnegative matrix factorization (NMF) problem. tensor. The main difference between NTF and NTD model is the core tensor in NTD allows column vectors in different factor matrices to interact with each other to reconstruct the original tensor, while the NTD model confines such interactions to occur among the components. Given a $(s+1)$-way tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_{(s+1)}}$ which consists N data points that each data point is represented by a s-way tensor. These data points are then concatenated along the $(s+1)$-mode. the NTD model is given by:

$$\mathcal{X} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times \ldots \times_{(s+1)} \mathbf{A}^{(s+1)} + \mathcal{E} \tag{5.22}$$

The core tensor $\mathcal{G} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_{(s+1)}}$ contains the variance of the data, and it controls the interactions between the modes component matrices $\mathbf{A}^{(i)}$. In other words, the core tensor is embedded with the compressed features, and the factors are bases of the feature subspace. To extract features from input tensor, we decompose $\mathcal{X}$ into a core tensor and s factor matrices except for the last mode.

By performing mode-n matricizing on the tensor:

$$\mathcal{Y} = \mathcal{X} \times_1 \mathbf{A}^{(1)T} \times_2 \mathbf{A}^{(2)T} \times \ldots \times_s \mathbf{A}^{(s)T} \tag{5.23}$$

with respect to the last mode (subjects mode), we can extract features from input samples and reduce the dimensionality of them in the form of matrix: $\mathbf{Y} \in \mathbb{R}^{(J_1 \times J_2 \ldots \times J_s) \times N}$, where each row vector is a new extracted feature and each column is the feature vector associated with a data point.

5.3.2 Multilinear subspace learning

Multilinear subspace learning aims to find a linear transform that maps the original input data to a lower dimensional subspace with multilinear algebra. The subspace learning has been widely used for data dimensionality reduction. Unsupervised dimensionality reduction have been used in pattern recognition and image processing. The typical methods include linear methods, such as PCA, which finds a projection matrix that maximizes that total variance of the projected data; and nonlinear methods, such as manifold learning which calculates the new embeddings in a lower dimensional space by learning the distance matrix along the manifold of the original input data. Famous manifold learning methods for dimensionality reduction include Locally linear embedding (LLE), Laplacian Eigenmaps (LE), Locality Preserving Projection (LPP), Neighborhood Preserving Embedding (NPE) and etc. LPP and NPE can be viewed as the optimal linear approximation to LLE and LE algorithm, respectively. By use of multilinear algebra and tensor representation, PCA, LPP and NPE can be generalized to accept input data in high-order tensor representations without converting them into vectors. These subspace learning methods explore the multilinear projections for direct mapping from input high-dimensional tensorial space to low-dimensional representations.

5.3.2.1 Multilinear Principal Component Analysis (MPCA)

The objective of PCA algorithm is to perform dimensionality reduction while preserving as much of the randomness in the original data as possible by projecting the data along the directions of maximal variances. Given a set of data points $\{\mathbf{x_1}, \mathbf{x_2}, ..., \mathbf{x_n}\},\ \mathbf{x_i} \in \mathbb{R}^{p\times 1}$, and the reduced data: $\{\mathbf{y_1}, \mathbf{y_2}, ..., \mathbf{y_n}\},\ \mathbf{y_i} \in \mathbb{R}^{q\times 1}$. Let $\mathbf{X}$ and $\mathbf{Y}$ be $p \times n,\ q \times n$ matrices, respectively. The $\mathbf{X},\ \mathbf{Y}$ are related by a linear transformation $\mathbf{W},\ \mathbf{Y} = \mathbf{W^T X}$. The objective function of PCA algorithm to find the projection matrix is as follows:

$$\arg\max_W \sum_i^n (\mathbf{y_i} - \bar{\mathbf{y}})^2 = \arg\max_W \mathbf{W^T C W} \tag{5.24}$$

where $\bar{y} = \frac{1}{n}\sum y_i$ and $\mathbf{C}$ is the data covariance matrix. The basis factors are the eigenvectors of the covariance matrix $\mathbf{C}$. By selecting the first q eigenvector and stacking them along the rows, we can obtain the projection matrix $\mathbf{W}$. It can be mathematically seen that the choice of $\mathbf{W}$ is to diagonalize the covariance matrix $\mathbf{C}$ in the projected space so to decorrelate the data.

A two-dimensional PCA (2DPCA) algorithm is proposed in [110] to solve for a linear transformation that projects a 2D image to a low-dimensional matrix while maximizing the variance measure. However, 2DPCA actually only seeks one linear transformation in the 2^{nd}-mode while the projection along the 1^{st}-mode is ignored. A more general higher order PCA algorithm named the generalized low rank approximation of matrices (GLRAM) was introduced in [111]. GLRAM calculates two linear projections along both the row and column of the input image. Compared to 2DPCA for dimensionality reduction of input image dataset, the GLRAM further reduces the computational cost. But, both algorithms are restricted to accept matrix as input data type. Multilinear

Principal Component Analysis (MPCA) [**?**] further generalize the PCA algorithm to work for tensors of arbitrary order. The objective is to find tensor-to-tensor projections that captures most of the original tensorial data variations. GLRAM is a special case to MPCA when the individual input data is of 2^{nd} order.

MPCA is basically the Tucker decomposition of a k^{th}-order tensor with one mode uncompressed. Analogous to PCA, MPCA aims to find a TTP that captures most of the original tensorial data variations. Given a set of input tensors $\mathcal{X}_1, ..., \mathcal{X}_m \in \mathbb{R}^{n_1 \times n_2 \times ... \times n_k}$, the mean tensor $\bar{\mathcal{X}} = \frac{1}{m}\sum_i^m \mathcal{X}_i$ is subtracted to make a zero-centered data set $\widetilde{\bar{\mathcal{X}}}\mathcal{X} = \frac{1}{m}\sum_i^m \widetilde{\mathcal{X}}_i$. Then an initial set of projection matrices $\mathbf{U^{(1)}}$, $\mathbf{U^{(2)}}$ are obtained by solving the eigen-decomposition of:

$$\Phi^{(n)*} = \sum_i^m \widetilde{\mathbf{X}}_{i(n)} \cdot \widetilde{\mathbf{X}}_{i(n)}^T,\ n = \{1, 2, ..., k-1\} \tag{5.25}$$

where $\widetilde{\mathbf{X}}_{i(n)}$ is the n-mode unfolded matrix of $\widetilde{\mathcal{X}}_i$. Afterwards, iterative optimization is performed until the result converges or a maximum number of iterations is reached. For more details about MPCA, we refer readers to work of Haiping Lu et al. [112].

5.3.2.2 Tensor Locality Preserving Projection (TLPP)

The Locality Preserving Projection (LPP) algorithm is a linear approximation to the Laplacian Eigenmap (LE) algorithm. LLP builds a graph by incorporating the local neighborhood information of each data point. It then finds a linear transformation matrix which maps the original data points into a new space based on the Laplacian of the graph. LPP takes the input data points as vectors and aims at preserving the local structure of the data points in high-dimensional space. It supposes there exists a linear mapping between the input data $\mathbf{x_i}$ and the output data $\mathbf{y_i}$, *i.e.* $\mathbf{y_i} = \mathbf{A^T x_i}$, where $\mathbf{A^T}$ is the mapping matrix. The LPP algorithm may adopt k nearest neighbor method to connect the data points and a Gaussian kernel function to compute the edge weight w_{ij} between them. Given the adjacency matrix $\mathbf{G}$, degree matrix $\mathbf{D}$ and the graph Laplacian matrix $\mathbf{L}$, the mapping matrix is calculated by minimizing the objective function:

$$\begin{aligned}\sum_{i,j}(\mathbf{y_i}-\mathbf{y_j})^2 w_{ij} &= \sum_{i,j}(\mathbf{A^T x_i}-\mathbf{A^T x_j})^2 w_{ij} \\ &= \arg\min_{\mathbf{A^T X D X^T A = 1}} \mathbf{A^T X L X^T A}\end{aligned} \tag{5.26}$$

This equals a generalized eigen problem:

$$\mathbf{XLX^T a} = \lambda \mathbf{XDX^T a} \tag{5.27}$$

The solutions to the eigen problem are the column vectors $[\mathbf{a_0}, ..., \mathbf{a_p}]$ as the eigenvectors, and $\lambda_0 < \lambda_1 < ... < \lambda_p$ as the corresponding eigenvalues. The mapping matrix is defined as the concatenation of the first q eigenvectors, $q << p$: $\mathbf{A^T} = [\mathbf{a_0}, ..., \mathbf{a_q}]$. This algorithm was generalized by He et al. [95] to accept tensors instead of one-dimensional vectors. Given a set of data points represented by second order tensors $\mathcal{X}_1, ..., \mathcal{X}_m \in \mathbb{R}^{n_1 \times n_2}$, similar to the idea of tensor decomposition, the TLPP algorithm aims to find two transformation matrices that map the m tensors from space $\mathbb{R}^{n_1 \times n_2}$ to a subspace of $\mathbb{R}^{I_1 \times I_2}$ ($I_1 < n_1$, $I_2 < n_2$). This can be written as $\mathcal{Y}_i = \mathbf{U^T} \mathcal{X}_i \mathbf{V}$, $i \in (1, 2..., m)$. Given the input 2^{rd}-order tensorial data, the objective function for TLPP algorithm becomes:

$$\min_{\mathbf{U,V}} \sum_{\mathbf{i,j}} \|\mathbf{U^T X_i V} - \mathbf{U^T X_j V}\|^2 w_{ij} \tag{5.28}$$

Using multilinear algebra, solving for the above objective function can be reduced to simultaneously minimize $tr(\mathbf{U^T}(\mathbf{D_v} - \mathbf{S_v})\mathbf{U})$ and $tr(\mathbf{V^T}(\mathbf{D_u} - \mathbf{S_u})\mathbf{V})$, where $\mathbf{D_U} = \sum_\mathbf{i} \mathbf{D_{ii}} \mathcal{X}_\mathbf{i}^\mathbf{T} \mathbf{U U^T} \mathcal{X}_\mathbf{i}$ and $\mathbf{S_U} = \sum_\mathbf{ij} \mathbf{S_{ij}} \mathcal{X}_\mathbf{i}^\mathbf{T} \mathbf{U U^T} \mathcal{X}_\mathbf{j}$. To further maximize the global variance in the tensor subspace, the optimization problems become:

$$\begin{aligned}&\min_{U,V} \frac{tr(\mathbf{U^T}(\mathbf{D_V} - \mathbf{S_V})\mathbf{U})}{tr(\mathbf{U^T D_V U})} \\ &\min_{U,V} \frac{tr(\mathbf{V^T}(\mathbf{D_U} - \mathbf{S_U})\mathbf{V})}{tr(\mathbf{V^T D_U V})}\end{aligned} \tag{5.29}$$

He, et al. proposed an iterative method to compute for the two transformation matrices $\mathbf{U}, \mathbf{V}$ that finds the mapping between the original tensor data and the reduced model in a nonlinear sub-manifold. This concept can be further generalized to accept input data in arbitrary order. The minimization problem can then be formulated as:

$$\begin{aligned}\arg\min \mathbf{Q}(\mathbf{U_1}, \mathbf{U_2}, ..., \mathbf{U_k}) &= \sum_{ij} \|\mathcal{Y}_i - \mathcal{Y}_j\|^2 w_{ij} \\ &= \sum_{ij} \|\mathcal{X}_i \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_. .. \times_{(k)} \mathbf{U}^{(k)} - \\ &\quad \mathcal{X}_j \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_. .. \times_{(k)} \mathbf{U}^{(k)}\|^2 w_{ij}\end{aligned} \tag{5.30}$$

Where $\mathbf{U}^{(f)}$ is the projection matrix along the f-mode. Similar to the two mode case, each projection matrix can be estimated in an iterative scheme by solving the following eigen problem, assuming the other projection matrices $\mathbf{U^{(1)}}, ..., \mathbf{U^{(f-1)}}, \mathbf{U^{(f+1)}}, ..., \mathbf{U^{(k)}}$ have been calculated:

$$\mathbf{H_2 U^{(f)} = H_1 U^{(f)} \Lambda} \tag{5.31}$$

where we have:

$$\begin{aligned}\mathbf{H_1} &= \sum_{\mathbf{ij}} \mathbf{d_{ij} Y_i^{(f)} Y_j^{(f)T}} \\ \mathbf{H_2} &= \sum_{\mathbf{i}} \mathbf{Y_i^{(f)} Y_i^{(f)T}}\end{aligned} \tag{5.32}$$

d_{ij} is the element in the degree matrix $\mathbf{D}$, and $\mathbf{Y}_i^{(f)}$ represents the mode-f unfolding for output data $\mathcal{Y}_i$. Given an initial set of $\mathbf{U_0^{(1)}}, ..., \mathbf{U_0^{(k)}}$, the projected subspace can be iteratively learnt for each mode until the result converges or a maximum iteration met.

5.3.2.3 Tensor Neighborhood Preserving Embedding (TNPE)

The Neighborhood Preserving Embedding (NPE) algorithm is a linear approximation to the LLE algorithm. It characterizes the local neighborhood structure of each input data point by linear coefficients $\mathbf{w}_{ij}$ such that one data point x_i can be approximately reconstructed from its k neighbors by a linear combination: $\hat{x}_i = \sum_1^K \mathbf{w}_{ij} x_j$. NPE finds the coefficients $\mathbf{w}_{ij}$ that best preserve the local neighborhood in a lower dimensional subspace and seeks a linear mapping such that the local structure is preserved in the projected subspace. A graph is first constructed to identify the local neighborhood of each data point. Then the linear coefficients $\mathbf{w}_{ij}$ for every data point are obtained by the following optimization:

$$\min \sum_i \|x_i - \sum_j^k \mathbf{w}_{ij} x_j\| \tag{5.33}$$

Those coefficients are calculated by solving a least-squares problem subject to the constraints: $\sum_1^K \mathbf{w}_{ij} = 1$, for each $j = 1, 2, ..., K$. Based on the learnt local neighborhood structure, a linear mapping: $\mathbf{Y} = \mathbf{A^T X}$, is obtained by preserving the same neighborhood characteristics inthe output subspace, such that:

$$\begin{aligned} \mathbf{y}^* &= \arg\min_y \sum_i \|y_i - \sum_j^K \mathbf{w}_{ij} y_j\| \\ &= \arg \min_{\mathbf{y:y^Ty=1}} \mathbf{y^T(I-W)^T(I-W)y} \\ &= \arg \min_{\mathbf{a:a^TXX^Ta=1}} \mathbf{a^TXMX^Ta} \end{aligned} \tag{5.34}$$

where I is the identity matrix and M is the $n \times n$ matrix given by:

$$\mathbf{M} = (\mathbf{I} - \mathbf{W})^\mathbf{T}(\mathbf{I} - \mathbf{W}) \tag{5.35}$$

This optimization problem for the calculation of the projection matrix $\mathbf{A}$ can be solved by the following generalized eigen problem:

$$\mathbf{XMX^Ta} = \lambda \mathbf{XX^Ta} \tag{5.36}$$

The column vectors $[\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_d}]$, ordered according to their eigenvalues, $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_d$, form into the projection matrix $\mathbf{A} = [\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_d}]$. And the new embedding: $\mathbf{y}_i = \mathbf{A^T x_i}$, is a d-dimensional vector.

TNPE extends the linear projection to multilinear projection to find k optimal projections $\mathbf{U_i} \in \mathbb{R}^{\mathbf{I_1 \times I_2 \times ... \times I_k}}$, such that the local topological structure learnt with graph model is preserved. Given the input data in tensorial representations $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n$, with $\mathcal{X}_i \in \mathbb{R}^{n_1 \times n_2 \times ... \times n_k}$, the optimization function for TNPE becomes:

$$\begin{aligned}\arg\min \mathbf{Q}(\mathbf{U_1}, \mathbf{U_2}, ..., \mathbf{U_k}) &= \sum_i \|\mathcal{Y}_i - \textstyle\sum_j^K w_{ij}\mathcal{Y}_j\| \\ &= \sum_i \|\mathcal{X}_i \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_. .. \times_{(k)} \mathbf{U}^{(k)} - \\ &\textstyle\sum_j^K w_{ij}\, \mathcal{X}_j \times_1 \mathbf{U}^{(1)} \times_2 \mathbf{U}^{(2)} \times_. .. \times_{(k)} \mathbf{U}^{(k)}\|\end{aligned} \tag{5.37}$$

This optimization problem can be solved approximately by adopting an iterative scheme which was originally proposed for low-rank approximation of second-order tensor. For each mode f, assuming the projection matrices for all other modes are known, the f-mode unfolding of $\mathcal{Y}_i$ can be obtained as the following:

$$\begin{aligned}\mathbf{Y_i^{(f)}} &= \mathcal{Y}_{(n)} \\ &= \mathcal{X}_i \times_\mathbf{1} \mathbf{U^{(1)}} \times_\mathbf{f-1} \times_1 \mathbf{U^{(f-1)}} \times_\mathbf{f+1} \mathbf{U^{(f+1)}} ... \times_\mathbf{k} \mathbf{U^k}\end{aligned} \tag{5.38}$$

Based on $\mathbf{Y_i}$, the factor matrix $\mathbf{U^{(f)}}$ can be estimated by solving for the eigenvectors corresponding to the first smallest eigenvalues in the generalized eigenvalue equation:

$$\mathbf{H_1 U^{(f)}} = \mathbf{H_2 U^{(f)} \Lambda_f} \tag{5.39}$$

where:

$$\begin{aligned}\mathbf{H_1} &= \sum_\mathbf{ij} \mathbf{m_{ij} Y_i^{(f)} Y_j^{(f)T}} \\ \mathbf{H_2} &= \sum_\mathbf{i} \mathbf{Y_i^{(f)} Y_i^{(f)T}}\end{aligned} \tag{5.40}$$

$\mathbf{m_{ij}}$ is the element in the graph $\mathbf{M}$ matrix describing the pairwise relationship between data point i and j. The other projections can be similarly calculated in an iterative procedure by solving the corresponding generalized eigenvalue problems.

The workflow of the three tensor subspace learning algorithms for dimensionality reduction of tensorial input data: MPCA, TLPP and TNPE, are summarized in the following charts. We refer readers to the original authors to learn the details of the algorithms and their mathematical backgrounds.

Algorithm 1: MPCA

Input : n input data points in tensorial representations: $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n$, with $\mathcal{X}_i \in \mathbb{R}^{n_1 \times n_2 \times ... \times n_k}$. Also, the subspace dimensions $\mathbb{R}^{m_1 \times m_2 \times ... \times m_k}$.

Output: The subspace projection matrices: $\mathbf{U^{(1)}}, ..., \mathbf{U^{(k)}}$, with $\mathbf{U}^{(i)} \in \mathbb{R}^{m_i \times n_i}$.

Center each input data as $\{\widetilde{\mathcal{X}}_i = \mathcal{X}_i - \bar{\mathcal{X}}_i,\ \bar{\mathcal{X}}_i = \frac{1}{n}\sum_1^n \mathcal{X}_i,\ i = 1, ..., n\}$;
for $f = 1 : k$ **do**
    1. Construct: $\Phi^{(f)*} = \sum_1^n \widetilde{X}_{i(f)} \cdot \widetilde{X}_{i(f)}^T$;
    2. Solve for eigenvectors: $\Phi^{(f)*}\widetilde{\mathbf{U}}^{(n)} = \lambda \widetilde{\mathbf{U}}^{(n)}$, where $\widetilde{\mathbf{U}}^{(n)}$ consists of the eigenvectors corresponding to the most significant m_f eigenvectors;
end
Obtain $\widetilde{\mathbf{Y}}_i = \widetilde{\mathbf{X}}_i \times_1 \widetilde{\mathbf{U}}^{(1)T} ... \times_k \widetilde{\mathbf{U}}^{(k)T}$;
Calculate $\Psi_{\mathcal{Y}_0} = \sum_1^n \|\widetilde{\mathcal{Y}}_i\|_F^2$;
for $t = 1 : t_{max}$ **do**
    for $f = 1 : k$ **do**
        1. Solve the eigenvectors of the matrix: $\Phi^{(f)} = \sum_1^n \mathcal{X}_{i(f)} \cdot \widetilde{\mathbf{U}}_{\Phi^{(f)}} \cdot \widetilde{\mathbf{U}}_{\Phi^{(f)}}^T \cdot \mathcal{X}_{i(f)}^T$ corresponding to the largest m_f eigenvectors, where $\widetilde{\mathbf{U}}_{\Phi^{(f)}} = (\widetilde{\mathbf{U}}^{\mathbf{(f+1)}} ... \otimes \widetilde{\mathbf{U}}^{\mathbf{(k)}} \otimes \widetilde{\mathbf{U}}^{\mathbf{(1)}} \otimes \widetilde{\mathbf{U}}^{\mathbf{(2)}} ... \otimes \widetilde{\mathbf{U}}^{\mathbf{(f-1)}})$;
        2. Set $\widetilde{\mathbf{U}}^{(f)}$ to consist of the first m_f eigenvectors. ;
        3. Derive updated $\widetilde{\mathcal{Y}}_i, i = 1, 2, ..., n$, and obtain updated $\Psi_{\mathcal{Y}_t}$;
    end
    if $\Psi_{\mathcal{Y}_t} - \Psi_{\mathcal{Y}_{t-1}} < \eta$ **then**
        break;
    end
end

Algorithm 2: TLPP

Input : 1. n input data points in tensorial representations: $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n$, with $\mathcal{X}_i \in \mathbb{R}^{n_1 \times n_2 \times ... \times n_k}$.
2. The subspace dimensions $\mathbb{R}^{m_1 \times m_2 \times ... \times m_k}$.
3. The graph degree matrix $\mathbf{D}$.

Output: The subspace projection matrices: $\mathbf{U^{(1)}}, ..., \mathbf{U^{(k)}}$, with $\mathbf{U}^{(i)} \in \mathbb{R}^{m_i \times n_i}$.

```
Initialize the projection matrices  U_1^0 = I ∈ R^{n_1×n_1}, ..., U_k^0 = I ∈ R^{n_k×n_k} ;
for t = 1 : t_max do
    for f = 1 : k do
        1. Calculate for every data point:
        Y_i^f = X_i ×_1 U^(1) ... ×_{f−1} U^(f−1) ×_{f+1} U^(f+1) ... ×_k U^(k);
        2. Unfold Y_i^f along the f-mode to get matrix Y_i^(f);
        3. Derive H_1, H_2 as the following:
        H_1 = Σ_ij d_ij Y_i^(f) Y_j^(f)T
        H_2 = Σ_i Y_i^(f) Y_i^(f)T;
        4. Solve for the first largest eigenvectors of:
        H_2 U_t^(f) = H_1 U_t^(f) Λ,  U_t^(f) ∈ R^{m_f×n_f} ;
        if ||U_t^(f) − U_{t−1}^(f)||_F < η for each f then
            break;
        end
    end
end
```

5.4 Pixel-level and Superpixel-level Spatial-spectral HSI Classification with Tensor Representation

Hyperspectral data with high-spatial resolution have become available, which provides wealthy spatial and spectral information. For an accurate interpretation of HSI data, it is desirable to simultaneously exploit the radiometric information in the spectral domain and the structural information in the spatial domain. As described earlier, tensor representation provides the capability in joint spectral-spatial expatiation of HSI data. By first extracting multi-domain features from the original image cube, a tensor can be adopted to embed features along different modes to keep the high-dimensional data structure.

The general workflow of the proposed tensor-based spatial-spectral HSI classification problem is summarized in Fig. 5.14. In pixel-wise DR and classification, each pixel is

---

Algorithm 3: TNPE

Input : 1. n input data points in tensorial representations: $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_n$, with $\mathcal{X}_i \in \mathbb{R}^{n_1 \times n_2 \times ... \times n_k}$.
2. The subspace dimensions $\mathbb{R}^{m_1 \times m_2 \times ... \times m_k}$.
3. The graph matrix derived from the adjacency matrix: $\mathbf{M} = (\mathbf{I} - \mathbf{W})^{\mathbf{T}}(\mathbf{I} - \mathbf{W})$.

Output: The subspace projection matrices: $\mathbf{U}^{(\mathbf{1})}, ..., \mathbf{U}^{(\mathbf{k})}$, with $\mathbf{U}^{(i)} \in \mathbb{R}^{m_i \times n_i}$.

Initialize the projection matrices $\mathbf{U}_1^0 = \mathbf{I} \in \mathbb{R}^{n_1 \times n_1}, ..., \mathbf{U}_k^0 = \mathbf{I} \in \mathbb{R}^{n_k \times n_k}$;
for $t = 1 : t_{max}$ **do**
  for $f = 1 : k$ **do**
    1. Calculate for every data point:
    $\mathcal{Y}_i^f = \mathcal{X}_i \times_1 \mathbf{U}^{(1)} ... \times_{f-1} \mathbf{U}^{(f-1)} \times_{f+1} \mathbf{U}^{(f+1)} ... \times_k \mathbf{U}^{(k)}$;
    2. Unfold $\mathcal{Y}_i^f$ along the f-mode to get matrix $\mathbf{Y}_i^{(f)}$;
    3. Derive $\mathbf{H_1}$, $\mathbf{H_2}$ as the following:
    $\mathbf{H_1} = \sum_{\mathbf{ij}} \mathbf{m_{ij}} \mathbf{Y_i^{(f)}} \mathbf{Y_j^{(f)T}}$
    $\mathbf{H_2} = \sum_{\mathbf{i}} \mathbf{Y_i^{(f)}} \mathbf{Y_i^{(f)T}}$;
    4. Solve for the first smallest eigenvectors of:
    $\mathbf{H_1 U_t^{(f)}} = \mathbf{H_2 U_t^{(f)}} \boldsymbol{\Lambda}$, $\mathbf{U}_t^{(f)} \in \mathbb{R}^{m_f \times n_f}$;
    if $||\mathbf{U_t^{(f)}} - \mathbf{U_{t-1}^{(f)}}||_F < \eta$ *for each* f **then**
      break;
    end
  end
end

---

modeled as a 2-way tensor of size 5-by-99. Four spatially nearest neighbors are included in the tensor modeling. Similarly, in superpixel-wise DR and classification, each superpixel is modeled by reshaping the atomic region into a 2-way tensor of size N_a-by-99, where N_a is the total number of pixels in a superpixel. Attention needs to be paid toward LRTA algorithm since it takes the whole image cube as a 3-way tensor and directly decomposes it with TUCKER3, it cannot be directly folded into the superpixel DR framework. We will simply use the mean spectrum of each superpixel and treat them as pixels on a 2D spatial grid to form a 3-way tensor. We will use it as the pseudo 3-way tensor representation for the HSI, which has superpixels as the smallest processing unit, and apply LRTA on it. However, the classification accuracy may be affected due to the irregular shape of superpixels.

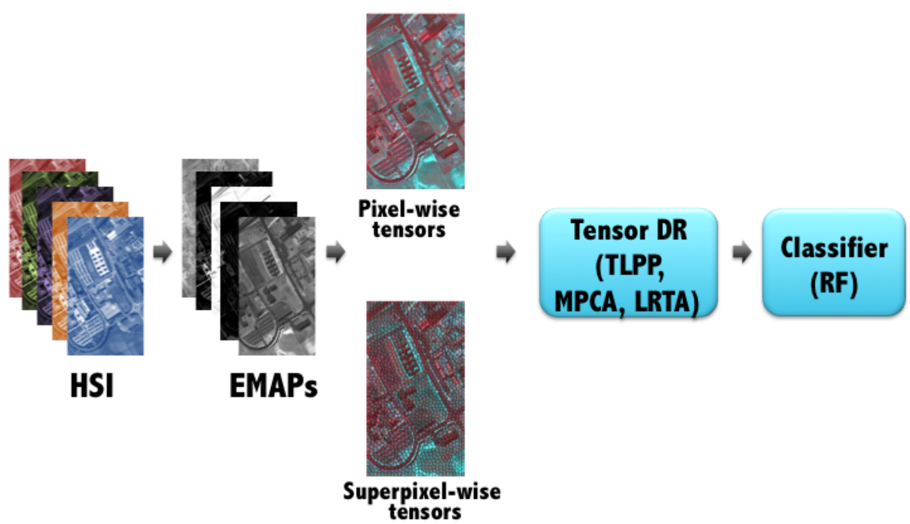


Figure 5.14: Workflow of spatial-spectral pixel-wise and superpixel-wise image classification with 2^{nd} tensor representation for every HSI data point.

The experiments were carried out on the popular Pavia University dataset. The scenes contained in Pavia University dataset are mainly an urban area, which were acquired by the hyperspectral airborne sensor ROSIS (Reflective Optics Systems Imaging

Spectrometer) during a flight campaign over Pavia, Northern Italy. The sensor acquired 115 spectral bands, ranging from 0.43 mm to 0.86 mm with the geometrical resolution of 1.3 m. The data acquired over the university area has a resolution of 610×340 and is composed of 103 bands (12 bands were abandoned due to noise). Nine classes were identified in this scene: trees, asphalt, bitumen, gravel,metal sheets, shadows, self-blocking bricks, meadows and bare soil. The true-color image and its ROIs for training and testing are shown in Fig. 5.15.

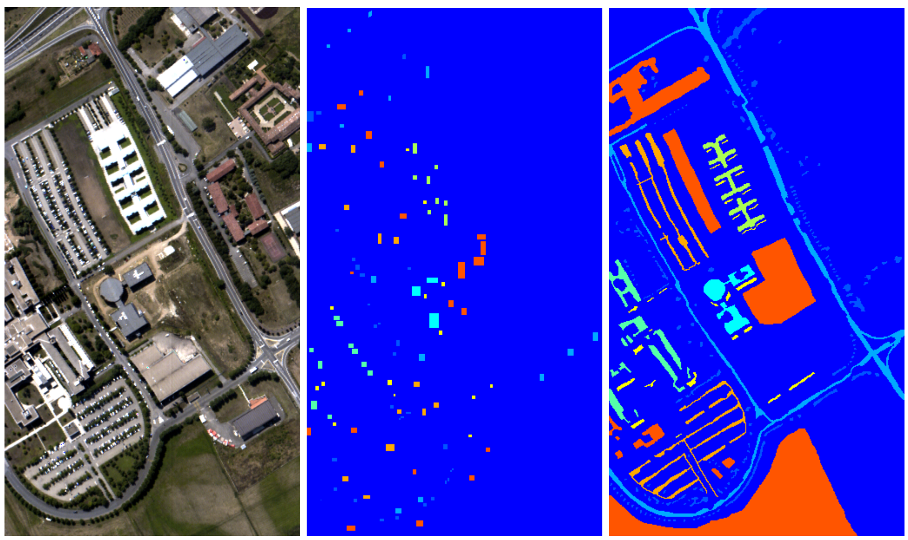


Figure 5.15: Left: Pavia University image in RGB. Middle: Training data. Right: Testing data.

The spatial features were calculated in EMAP feature space. The first four PCs were used for the calculation of EMAPs for Pavia University dataset for they contain more than 99% variance in the original data. Four attributes are measured in EMAP feature space (i.e., area of the region λ_a, diagonal of the box bounding λ_d, moment of inertia λ_i and standard deviation λ_s) and the values for each attributes are: $\lambda_a = [500, 1000, 1500, 2000]$, $\lambda_d = [10, 40, 70, 100]$, $\lambda_i = [0.2, 0.4, 0.6, 0.8]$, $\lambda_a = [10, 60, 110, 160]$, respectively. In cases where a filtered image suppressed too much spatial content, which may result in an all-black or all-white image, we add a threshold such that if the processed image (closing/opening profile) contains less than 10% non-0 and non-255 data, it will be abandoned. The result EMAPs have a total of 99 bands, which is close to the dimension of the original HSI. Some of the processed images contained in EMAPs are displayed in Fig. 5.15.

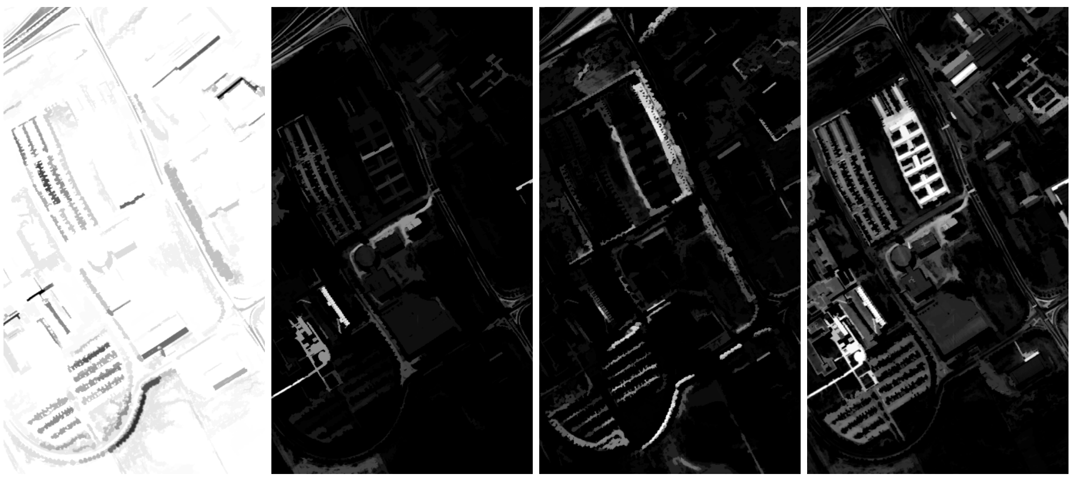


Figure 5.16: Some feature images in EMAPs for Pavia University.

To compare the vector-based dimensionality reduction (LPP, PCA) for fused spatial-spectral feature vectors with the introduced tensor-based methods (TLPP, MPCA and LRTA), we reduce the dimension to ten percent of original dimension for vector-based methods. And for tensor-based algorithms, we reduce the dimension along the spatial mode to one and the dimension along the spectral mode to ten percent of original number of bands. In other words, the vectorized data after either DR algorithm (vector-based/tensor-based) will have the same dimension before they are input to a classifier. Particularly for the LRTA algorithm, according to Renard's observation, the smaller spatial ranks may have a stronger positive influence on the classification accuracy. We therefore set J_1 and J_2 to be merely ten percent of the width and height of the original spatial dimensions, respectively.

The EMAP feature space contains the local shape information to which one pixel belongs and the spectral features contained in the PCA bands. We compared the vector-based methods LPP and PCA to reduce the feature dimensionality in the original HSI space and EMAP space. The comparison can be found in the Fig. 5.17. Results show that EMAP feature space produced better accuracies for both LPP and PCA compared to them applied on original HSI for that EMAP contain both extracted spatial correlations from the neighborhood of each pixel, as well as spectral information. LPP algorithm did not overperform the PCA in EMAP space for this dataset. However, they reported similar accuracy performances on HSI with respect to the AA, OA and Kappa coefficient.

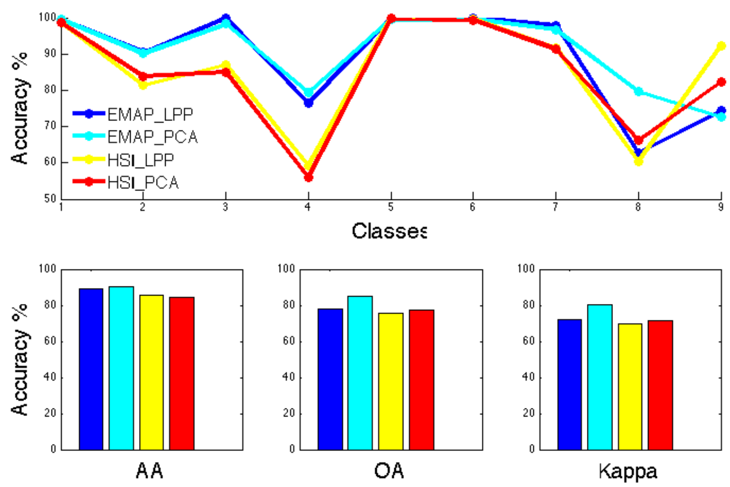


Figure 5.17: Top: Plots of the accuracy for each class in the ROIs for LPP and PCA applied on original HSI and EMAPs. Bottom: Bar plots for: Left: Average accuracy, Middle: Overall accuracy, Right: Kappa coefficient.

Vector-based DR methods: PCA and LPP, and their tensor counterparts: MPCA and TLPP, are compared, as well as the LRTA algorithm, which takes the entire image cube as a third order tensor without reformulating pixel-wise tensor representations along the last mode. From the results shown in Fig. 5.18, TLPP generated the highest OA/AA/Kappa accuracy among the five algorithms.

Tensors can also be used to model superpixels. In pixel tensor representations, a tensor incorporates the neighborhood information of each pixel. Similarly, we can rearrange pixels that form a superpixel along one mode as the neighborhood structure mode. By

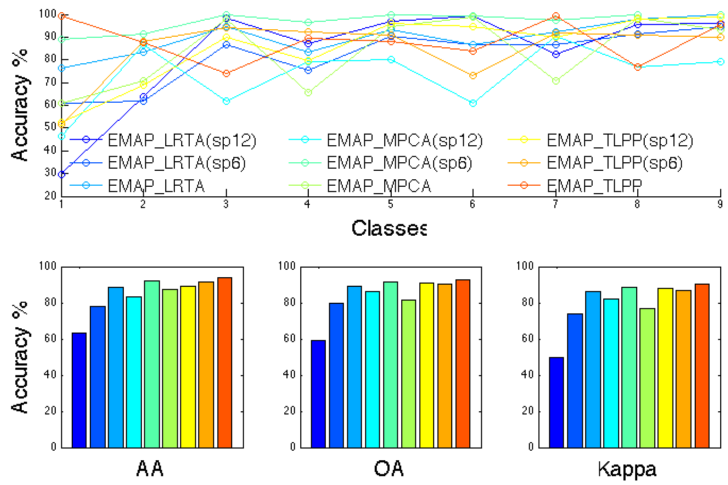


Figure 5.18: Top: Plots of the accuracy for each class in the ROIs for TLPP, MPCA and LRTA applied on EMAPs with pixel-wise DR and superpixel-wise DR. Bottom: Bar plots for: Left: Average accuracy, Middle: Overall accuracy, Right: Kappa coefficient.

processing superpixels, the number of data points to be processed is greatly reduced. We compared the three tensor-based DR methods on their performances in superpixel-wise classification. Two superpixel sizes were used $S = 6$, $m = 0.01$ and $S = 12$, $m = 0.01$. Their class-specific accuracies and AA/OA/Kappa are shown in Fig. 5.19. It was noted that TLPP performs equally well for both pixel-wise and superpixel-wise DR and classification. LRTA generates poor results for superpixel-wise DR and classification, and its accuracy drops significantly as the superpixel size increases. As mentioned earlier, LRTA is based on the decomposition of the whole high dimensional image which is not naturally compatible with the superpixel processing framework. Also, in Fig. 5.19, the MPCA with superpixel size of $S = 6$, $m = 0.01$ shows the best performance compared to pixel-wise DR and superpixel DR of size $S = 12$, $m = 0.01$. This is possibly because a $S = 6$, $m = 0.01$ superpixel enables more spatial correlations being included into a tensor representation compared to pixel-wise tensor representation where only four nearest neighbors were considered in the spatial mode. However, a larger superpixel size will

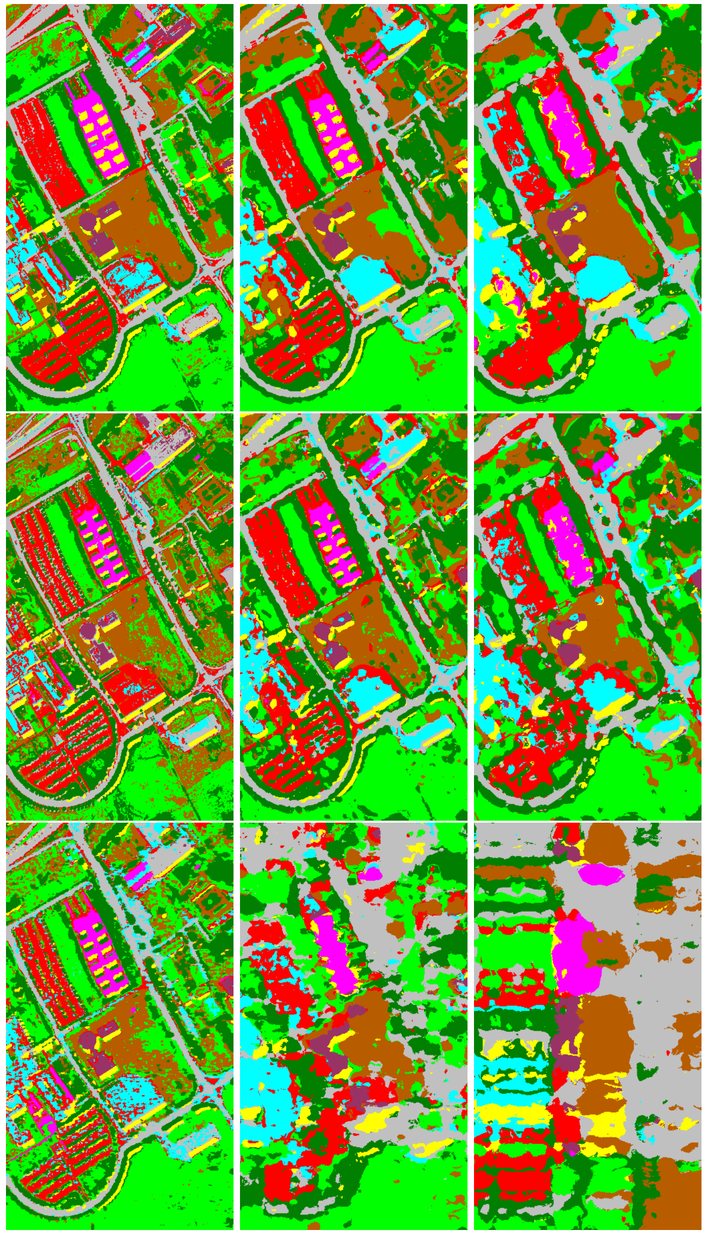


Figure 5.19: Classification maps of the Pavia University dataset obtained using DR methods of the following: First row: TLPP; Second row: MPCA; Third row: LRTA. In each row from left to right: pixel-wise classification, superpixel-wise classification of size $S = 6$, $m = 0.01$, superpixel-wise classification of size $S = 12$, $m = 0.01$.

result in less spatial and spectral correlations among all the superpixels, and therefore may lead to a decrease in the accuracy.

5.5 Results of HSI and LiDAR Fusion for Classification Based on Tensors

The HSI data has been explored to generate both spectral and spatial feature for accurate image interpretation. LiDAR sensor, provides more possibilities in measuring different aspects of the objects on the Earth's surface. The combination of both HSI and LiDAR data sources can contribute to a more comprehensive interpretation of the ground objects. For example, spectral feature itself cannot discriminate between objects of the same material (e.g., rooftops and roads made with the same asphalt), while the elevation information measured by LiDAR sensor can easily tell them apart. On the other hand, LiDAR data alone may fail to differentiate between objects that are different in nature but similar in their altitude or geometric structure. Therefore, we formulate a fourth-order tensor including modes of spatial-spectral-LiDAR features, size of features, local neighborhood and subjects to fully utilize both HSI and LiDAR data for feature fusion.

In HSI and LiDAR fusion area, features being exploited includes spatial and spectral features extracted from HSI, altitude information obtained directly from LiDAR sensor, and EMAP features generated on the elevation map. Considering the fact that many local features can be estimated to characterize the geometric surfaces in LiDAR point cloud, we proposed to use geometric features calculated with local neighborhood operators from LiDAR *.las* file, instead of simply using the elevation information only.

The dataset used for experiments on HSI and LiDAR data fusion were acquired by the NSF-funded Center for Airborne Laser Mapping (NCALM) [113]. The dataset was acquired over the University of Houston campus and the neighboring urban area on June 22, 2012. A hyperspectral image and a LiDAR point cloud las file, both at the same spatial resolution (2.5 m) are applied in this paper. The HSI image has 144 bands with wavelength range from 380 nm to 1050 nm. The whole scene of both the data, consisting of the full 349×1905 pixels, as shown in the following Fig. 5.21. 15 classes of interest are reported in the following Table. 5-1.

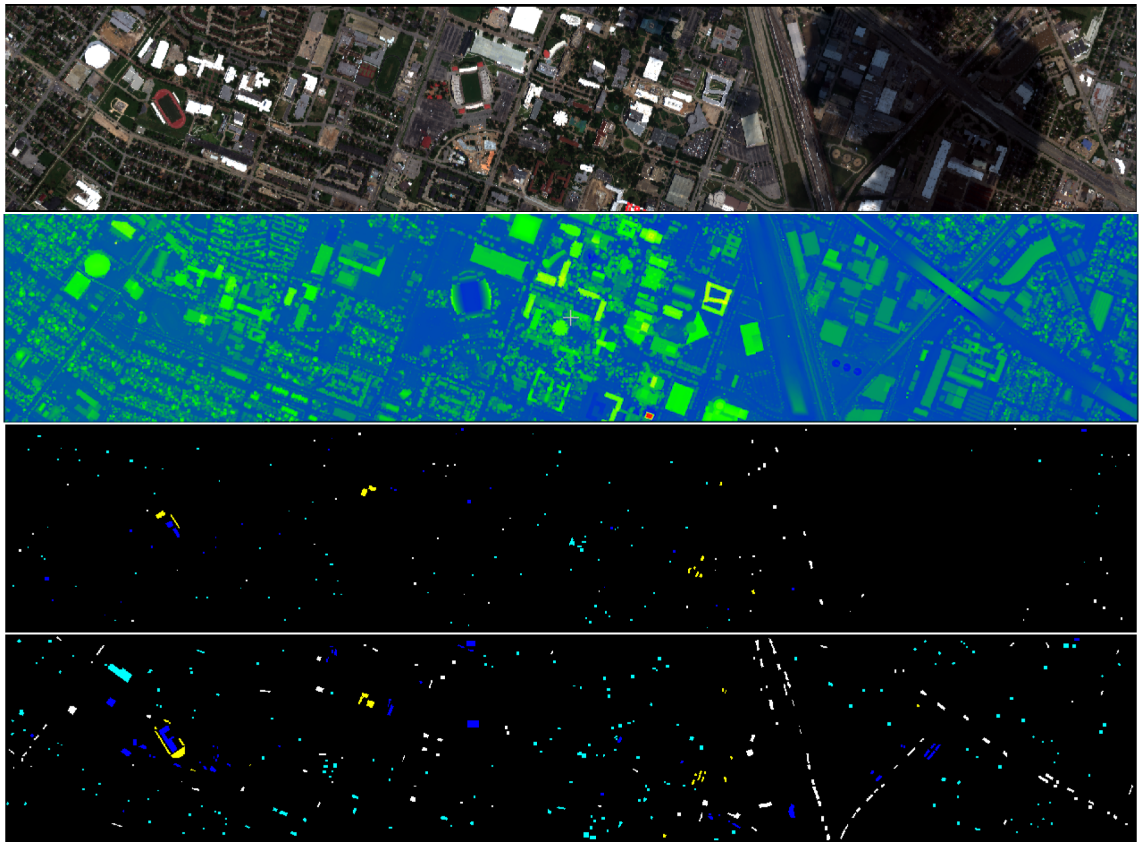


Figure 5.20: Data sets for HSI and LiDAR fusion: (a) HSI, (b) the LiDAR height map, (c) training ROI, and (d) validation ROI.

For every data point, three sources of features are extracted including the spectral features in PCA bands, spatial features in EMAP feature space, and LiDAR geometric features. The LiDAR geometric feature vector consists of ten elements (described in Section 5.2.3): elevation, Dip (degree), Dip (direction), Gaussian curvature, Normal change rate, Volume density, Roughness, and Normal vector magnitude in x, y, z directions, respectively. Each feature vector $\mathbf{x}^{\mathbf{spectral}} \in \mathbb{R}^{\mathbf{l}\times\mathbf{1}}$, $\mathbf{x}^{\mathbf{spatial}} \in \mathbb{R}^{\mathbf{l}\times\mathbf{1}}$, $\mathbf{x}^{\mathbf{LiDAR}} \in \mathbb{R}^{\mathbf{l}\times\mathbf{1}}$, is normalized to have unit length before being concatenated into a matrix. Four spatial neighbors are considered to form the local neighborhood for every data point along the third mode in a

Table 5.1: The training and testing data provided with HSI image

Classes	Training Set	Test Set	Class Color
grass_healty	198	1053	green2
grass_stressed	190	1064	chartreuse
grass_synthetic	192	505	sea green
tree	188	1056	green3
soil	186	1056	sienna
water	182	143	cyan
residential	196	1072	white
commercial	191	1053	thistle
road	193	1059	red
highway	191	1036	red3
railway	181	1054	black
parking_lot1	192	1041	yellow
parking_lot2	184	285	orange2
tennis_court	181	247	purple3
running_track	187	473	coral

tensorial representation $X^{all} \in \mathbb{R}^{l \times p \times q \times n}$, where l is the number of elements in each feature vector, p is the number of features, q is the size of local neighborhood and n is the number of subjects. Overall, all the data points are arranged along the fourth mode and combined into a 4^{th} order tensor with spatial, spectral and LiDAR features all embedded. The size of the 4^{th} order tensor in this experiment will then be $10 \times 3 \times 5 \times 664845$.

We introduce a method to fuse the multiple sources of features and reduce the dimensionality simultaneously with tensorial representation. The aim of this work is to show that tensors can be conveniently used as an effective tool for data fusion of remotely sensed data. As an explorative study of tensor model in data representation and fusion, we are more interested in discovering how the tensor-based algorithms perform compared to traditional vector-based algorithms, rather than tuning the feature mining techniques for getting stronger features. Compared to stacking multi-domain features into a long vector representation, tensors preserve the multilinear structure of the input data and alleviate the curse of dimensionality issue associated with vector representation. Therefore, the algorithm parameters selected in our framework, such as the number of nearest neighbors, the number of elements in an individual feature domain, the size of the subspaces and etc., were determined mainly in low computational cost consideration.

Five tensor-based dimensionality reduction algorithms were introduced for the spatial-spectral-lidar feature fusion of remotely sensed data for the first time. Details of the five algorithms can be found in the previous section. Two of the algorithms NTF and NTD are based on factor analysis models, and the other three algorithms, MPCA, TLPP and TNPE, are based on multilinear subspace analysis. The multilinear subspace learning algorithms are high order extensions to the linear subspace learning algorithms: PCA, LPP and NPE. We have generalized the algorithms to accept tensorial input data with any arbitrary order. The TLPP and TNPE algorithms adopt graph models to study the pairwise relationship between data points. These two Laplacian-based multilinear subspace learning algorithms were generalized with multilinear algebra to accept high order tensorial data to simultaneously exploit the multiple information and find appropriate subspace for the input data. In this spatial-spectral-lidar fusion experiment, the input data is of 4^{th} order.

It can be found that using single feature source is not sufficient for a reliable classification. Each feature source has its advantages on different classes. Some of the extracted features after fusion and dimensionality reduction with tensor-based algorithms are shown below. We can see that, in the tensor model derived feature subspace, some feature spaces are not affected by the cloud shadow present during the acquisition of the HSI. While, some other feature subspaces mainly capture specific geometric information, such as edge structures. The extracted tensor feature subspaces provides concise yet comprehensive representation of the HSI and LiDAR dataset.

The feature vectors obtained with the tensor-based DR algorithms are input to the RF classifier for classification of the input dataset. The classification results are quantitatively evaluated by measuring the OA, the AA, and the Kappa coefficient on the test samples. Table. 5.2 shows the accuracies obtained from our experiments, whereas the classification maps for the five tensor-based DR algorithms are illustrated in Fig. 5.22. The training set was split randomly into two non-overlapping subsets of the same size for training and testing, respectively. From the results, tensor-based DR algorithms generate much higher accuracy in terms of the overall accuracy than vector-based methods, except for the MPCA algorithm. The tensor factor analysis based method NTD generates the highest classification accuracy in this experiment.

Table 5.2: Performance comparison of the two tensor factor analysis algorithms NTF, NTD; the three tensor multilinear subspace learning algorithms TLPP, MPCA, TNPE; and the three traditional linear subspace learning algorithms LPP, PCA and NPE.

Algorithms	Average Accuracy	Overall Accuracy	Kappa Coefficient
TLPP	98.54%	98.46%	0.9835
MPCA	92.78%	92.49%	0.9195
TNPE	98.16%	98.10%	0.9796
NTF	98.45%	98.37%	0.9825
NTD	99.32%	99.28%	0.9922
LPP	96.65%	96.47%	0.9622
PCA	96.44%	96.20%	0.9593
NPE	95.29%	95.11%	0.9476

Table 5.3: Confusion Matrix - CP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Producer Acc(Precision)
grass_healthy	70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100.00
grass_stressed	0	72	0	0	0	0	0	0	0	1	0	0	0	0	0	98.63
grass_synthetic	0	0	68	0	0	0	0	0	0	0	0	0	0	0	0	100.00
tree	0	1	0	76	0	0	0	0	0	0	0	0	0	0	0	100.00
soil	0	0	0	0	61	0	0	0	0	1	0	0	0	0	0	98.39
water	0	0	0	0	0	81	0	0	0	0	0	0	0	0	0	96.43
residential	0	0	0	0	0	1	62	0	0	0	0	0	0	2	1	100.00
commercial	0	0	0	0	1	0	0	76	0	0	0	0	0	0	0	100.00
road	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	100.00
highway	0	0	0	0	0	0	0	0	0	72	0	0	0	0	6	92.31
railway	0	0	0	0	0	0	0	0	0	1	70	0	0	0	0	100.00
parking_lot1	0	0	0	0	0	0	0	0	0	1	0	82	0	0	0	100.00
parking_lot2	0	0	0	0	0	0	0	0	0	1	0	0	63	2	1	100.00
tennis_court	0	0	0	0	0	0	0	0	0	1	0	0	0	72	0	94.74
running_track	0	0	0	0	0	2	0	0	0	0	0	0	0	0	65	89.04
User Acc(Recall)	100.00	98.63	100.00	98.70	98.39	100.00	93.94	98.70	100.00	92.31	98.59	98.80	94.03	98.63	97.01	

The validation set is more challenging due to a large cloud shadow was present during the acquisition of the HSI. No training samples were included in the shadow region, however, a number of validation samples were selected to test the efficacy of the LiDAR and HSI fusion algorithms in dealing with shadow area. In order to enhance the algorithm performance on the validation set, we selected 20 features from LiDAR, HSI spatial and HSI spectral domain, respectively. 10 LiDAR geometric features generated from local neighbor analysis and 10 EMAP features generated on the elevation map were selected as the LiDAR features. 20 EMAP features generated on the first two PCA images and 20 PCA images were selected as the HSI spatial and spectral features, respectively. For all graph-based algorithms, the value of k-nearest neighbors is always set to 5. All the

Table 5.4: Confusion Matrix - MPCA

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Producer Acc(Precision)
grass_healthy	70	2	0	0	0	0	0	0	0	0	0	0	0	0	0	100.00
grass_stressed	0	70	0	0	0	0	0	0	0	1	2	0	0	0	0	95.89
grass_synthetic	0	0	68	0	0	0	0	0	0	0	0	0	0	2	0	100.00
tree	0	1	0	76	0	0	0	0	0	0	0	0	0	0	0	100.00
soil	0	0	0	0	61	0	0	0	0	10	0	0	0	0	0	98.39
water	0	0	0	0	0	81	0	0	0	0	0	0	0	0	0	96.43
residential	0	0	0	0	0	1	62	0	0	3	0	4	0	5	0	100.00
commercial	0	0	0	0	0	0	0	76	0	0	0	0	0	0	0	100.00
road	0	0	0	0	0	0	0	0	74	0	0	0	0	0	0	92.50
highway	0	0	0	0	1	0	0	0	0	60	0	0	0	3	8	76.92
railway	0	0	0	0	0	0	0	0	0	0	68	0	0	0	0	97.14
parking_lot1	0	0	0	0	0	0	0	0	0	0	0	78	0	2	0	95.12
parking_lot2	0	0	0	0	0	0	0	0	0	0	0	0	63	2	4	100.00
tennis_court	0	0	0	0	0	1	0	0	4	3	0	0	0	58	3	76.32
running_track	0	0	0	0	0	1	0	0	2	1	0	0	0	4	58	79.45
User Acc(Recall)	97.22	95.89	97.14	98.70	85.92	100.00	82.67	100.00	100.00	83.33	100.00	97.50	91.30	84.06	87.88	

Table 5.5: Confusion Matrix - NTD

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Producer Acc(Precision)
grass_healthy	70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100.00
grass_stressed	0	72	0	0	0	0	0	0	0	0	0	0	0	0	0	98.63
grass_synthetic	0	0	68	0	0	0	0	0	0	0	0	0	0	0	0	100.00
tree	0	1	0	76	0	0	0	0	0	0	0	0	0	0	0	100.00
soil	0	0	0	0	62	0	0	1	0	1	0	0	0	0	0	100.00
water	0	0	0	0	0	84	0	0	0	0	0	0	0	0	0	100.00
residential	0	0	0	0	0	0	61	0	0	0	0	0	0	5	1	98.39
commercial	0	0	0	0	0	0	0	75	0	0	0	0	0	0	0	98.68
road	0	0	0	0	0	0	0	0	78	1	0	0	0	0	0	97.50
highway	0	0	0	0	0	0	0	0	1	73	0	0	0	1	0	93.59
railway	0	0	0	0	0	0	0	0	0	0	69	0	0	0	0	98.57
parking_lot1	0	0	0	0	0	0	0	0	0	1	0	82	0	0	0	100.00
parking_lot2	0	0	0	0	0	0	1	0	0	1	0	0	63	0	2	100.00
tennis_court	0	0	0	0	0	0	0	0	0	0	0	0	0	70	2	92.11
running_track	0	0	0	0	0	0	0	0	1	1	1	0	0	0	68	93.15
User Acc(Recall)	100.00	100.00	100.00	98.70	96.88	100.00	91.04	100.00	98.73	97.33	100.00	98.80	94.03	97.22	95.77	

Table 5.6: Confusion Matrix - TLPP

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Producer Acc(Precision)
grass_healthy	70	2	0	0	0	0	0	0	0	0	0	0	0	0	0	100.00
grass_stressed	0	71	0	0	0	0	0	0	0	0	0	0	0	0	0	97.26
grass_synthetic	0	0	68	0	0	0	0	0	0	0	0	0	0	0	0	100.00
tree	0	0	0	76	0	0	0	0	0	0	0	0	0	0	0	100.00
soil	0	0	0	0	62	0	0	0	0	2	0	0	0	0	0	100.00
water	0	0	0	0	0	84	0	0	0	0	0	0	0	0	0	100.00
residential	0	0	0	0	0	0	62	0	0	0	0	0	0	3	0	100.00
commercial	0	0	0	0	0	0	0	75	0	0	0	0	0	0	1	98.68
road	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	100.00
highway	0	0	0	0	0	0	0	0	0	74	0	0	0	1	4	94.87
railway	0	0	0	0	0	0	0	0	0	1	69	0	0	0	0	98.57
parking_lot1	0	0	0	0	0	0	0	0	0	0	0	82	0	0	0	100.00
parking_lot2	0	0	0	0	0	0	0	0	0	0	0	0	63	1	1	100.00
tennis_court	0	0	0	0	0	0	0	0	0	1	0	0	0	70	5	92.11
running_track	0	0	0	0	0	0	0	1	0	0	1	0	0	1	62	84.93
User Acc(Recall)	97.22	100.00	100.00	100.00	96.88	100.00	95.38	98.68	100.00	93.67	98.57	100.00	96.92	92.11	95.38	

Table 5.7: Confusion Matrix - TNPE

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Producer Acc(Precision)
grass_healthy	70	2	0	0	0	0	0	0	0	0	0	0	0	0	0	100.00
grass_stressed	0	71	0	0	0	0	0	0	0	0	0	0	0	0	0	97.26
grass_synthetic	0	0	68	0	0	0	0	0	0	0	0	0	0	0	0	100.00
tree	0	0	0	76	0	0	0	0	0	0	0	0	0	0	0	100.00
soil	0	0	0	0	62	0	0	0	0	2	0	0	0	0	0	100.00
water	0	0	0	0	0	84	0	0	0	0	0	0	0	0	0	100.00
residential	0	0	0	0	0	0	62	0	0	0	0	0	0	3	0	100.00
commercial	0	0	0	0	0	0	0	75	0	0	0	0	0	0	0	98.68
road	0	0	0	0	0	0	0	0	80	0	0	0	0	0	0	100.00
highway	0	0	0	0	0	0	0	0	0	74	0	0	0	1	4	94.87
railway	0	0	0	0	0	0	0	1	0	1	69	0	0	0	0	98.57
parking_lot1	0	0	0	0	0	0	0	0	0	0	0	82	0	0	0	100.00
parking_lot2	0	0	0	0	0	0	0	0	0	0	0	0	63	2	1	100.00
tennis_court	0	0	0	0	0	0	0	0	0	1	0	0	0	70	5	92.11
running_track	0	0	0	0	0	0	0	0	0	0	1	0	0	0	63	86.30
User Acc(Recall)	97.22	100.00	100.00	100.00	96.88	100.00	95.38	100.00	100.00	93.67	97.18	100.00	95.45	92.11	98.44	

three set of features were concatenated into a 4^{th} order tensor data of size $20 \times 3 \times 5 \times 664845$. Total of 60 features were extracted from the HSI and LiDAR data. We obtained 89.2433% overall accuracy, 91.19383% average accuracy with TLPP algorithm.

5.6 Summary

In this Chapter, we introduced tensors as an alternative representation to traditional vectors, which have been adopted as the basic feature descriptor form for remotely sensed data. Considering the fact that remotely sensed data is usually multi-modal in nature, tensors provide an intuitive representation of the multi-dimensional structure.

The basic mathematical foundations of tensor algebra was introduced. Two most popular tensor models, i.e., the Tucker model and CP model, are explained as higher-order generalizations of the singular value decomposition (SVD) or principal component analysis (PCA) for matrix. The high-order factor analysis also can be utilized in the multi-linear subspace learning framework, which can be further generalized for dimensionality reduction of multi-dimensional input data. By the use of multilinear subspace learning, input tensorial data can be mapped to lower dimensional space.

The HSI data cube is a third order tensor with two spatial modes and one spectral mode. Tensors can be used to discover the joint hidden spatial-spectral structures using multilinear algebra. Conventionally, each image pixel is represented with a vector by simply concatenating all features associated with the pixel into a long vector. However, a

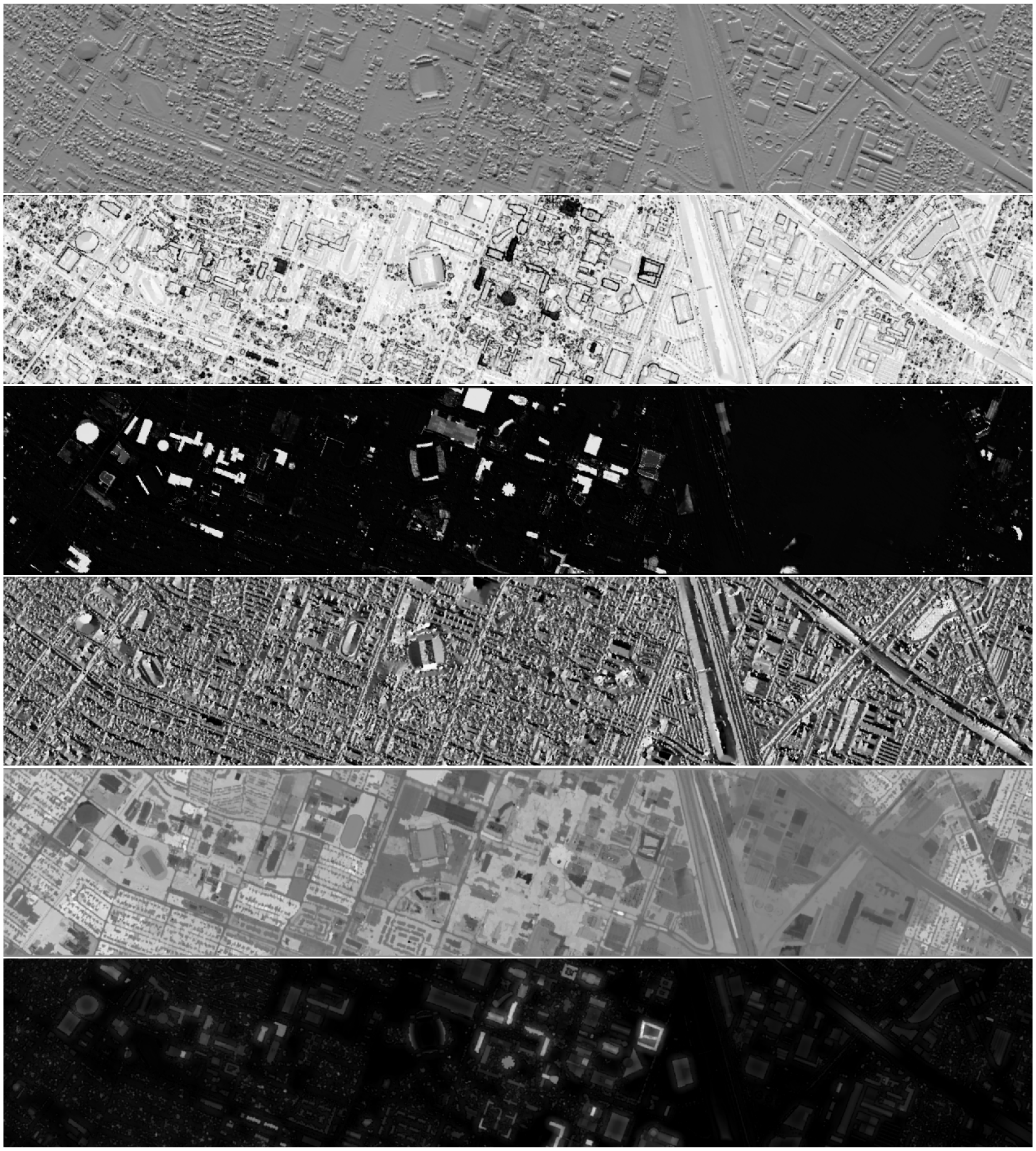


Figure 5.21: Some of the features being extracted after tensor-based dimensionality reduction.

simple concatenation may cause the long feature vector significantly unbalanced. Besides,

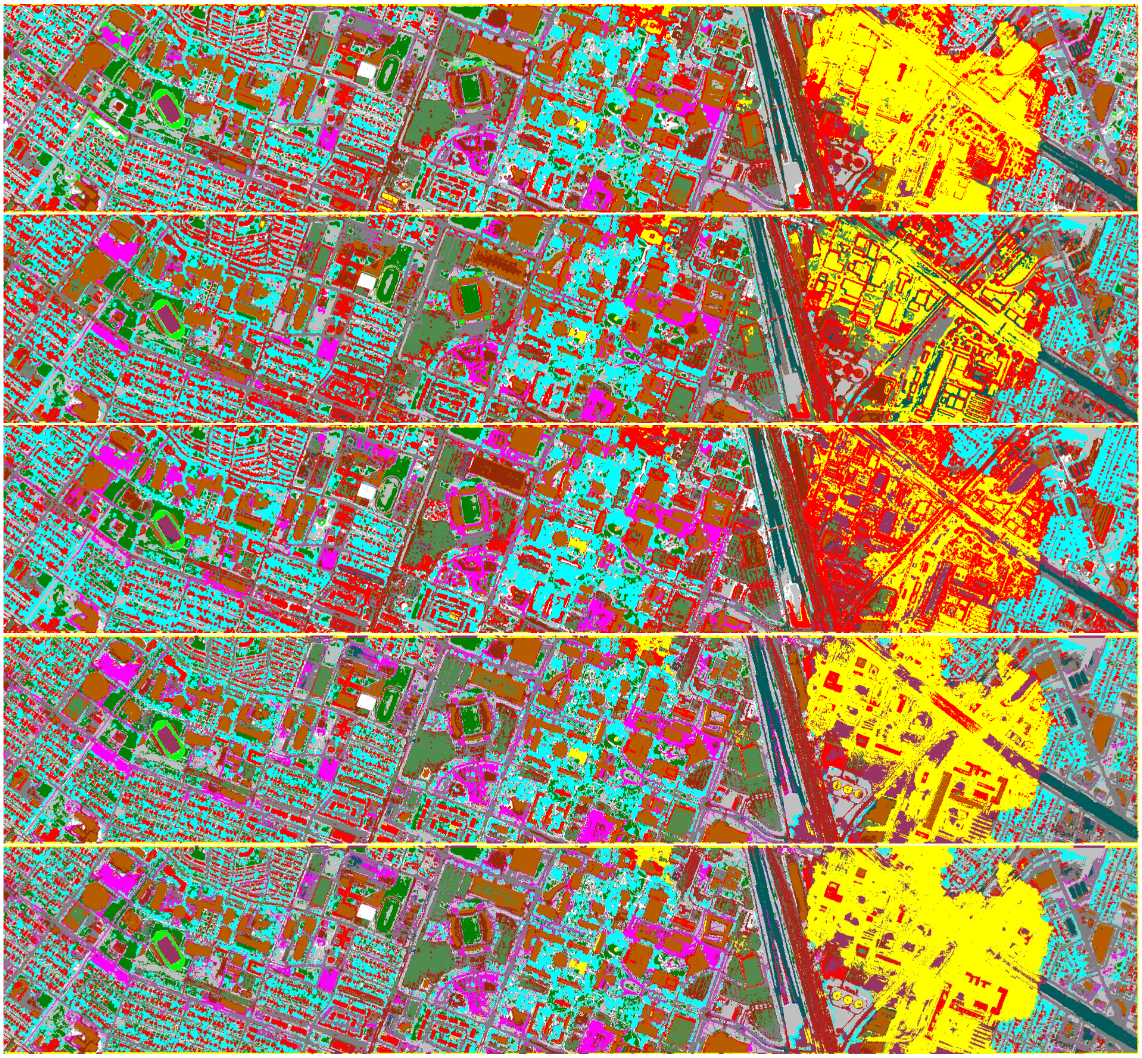


Figure 5.22: Classification map of the five tensor based dimensionality reduction algorithms: NTF, NTD, MPCA, TLPP and TNPE.

concatenation may lead to curse of dimensionality if the feature vector is large. While the multilinear algebra provides the possibility to merge multiple features along different modes into a higher order tensor. By adopting multilinear subspace learning, we are able to fuse multiple features and reduce dimensionality at the same time. Therefore, tensors can also be used for feature and data fusion of remotely sensed images.

Tensor-based dimensionality reduction algorithms were introduced and generalized to handle arbitrary order remotely sensed data in this work. Tensors were applied for pixel-based and superpixel-based HSI classification and shown very promising performance compared to traditional vector-based approaches. Also, the idea of tensor representation was also explored for multiple source data fusion, i.e., HSI and LiDAR fusion for data classification. Five tensor dimensionality reduction algorithms were introduced and generalized to solve remotely sensed multi-source data fusion problems for the first time.

Chapter 6

Summary

This thesis work has been focused on data and feature fusion in remote sensing area using graph-based data modeling and learning techniques. Hyperspectral imaging systems have fine spectral resolution to determine and characterize land cover classes based on their spectral signatures. It also contains abundant contextual information in the spatial domain that can be utilized to complement the spectral information for object differentiation. LiDAR sensors collect 3D point clouds that can be exploited to generate local geometric and shape information to further provide separabilities between different structures. The development of novel remote sensing data fusion techniques to address new challenging applications is demanding. Graph theory has been used in data modeling for its strong mathematical foundations in studying the discrete data relationships. We explored the graph modeling techniques, such as spectral clustering and subspace manifold learning, for remotely sensed data fusion in this work. With multilinear algebra, traditional vector-based algorithms are further generalized to handle tensorial representations. The tensorial representations can preserve the natural multi-dimensional data structure without rearranging data into traditional vector representations. It also help alleviate the curse of dimensionality problem associated with vector-based algorithms. Besides, tensors can be an efficient tool to embed various information in data fusion work.

6.1 Conclusions

There are a number of contributions in this thesis for remotely sensed data fusion community. Graph-based data modeling techniques and multilinear algebra are introduced for the processing of HSI and LiDAR data in this work.

6.1.1 Topology-based semi-supervised classification

TAD algorithm, an algorithm proposed to detect anomalies in a spectral image by identifying a background component, is modified for estimation of land cover types so to be used for supervised classification. Random subsamples of pixels are selected to learn the topological structure of the input dataset. Based on the assumption that pixels belong to the same material usually form strongly connected component in a graph, and with appropriate data trimming and mapping skills, the identified components by TAD are used to represent the features of land covers in a supervised image classification scheme. Overall, by learning the topological structure of the input data, we automate the selection of training data to be combined in a supervised learning scheme without the need for human generated ground truth data being provided.

6.1.2 Self-tuning spectral clustering

Spectral clustering algorithm, which is based on graph representation and topological structure of the data, has been used for HSI segmentation in recent years. Laplacian Eigenmaps, as one of the most popular spectral clustering algorithms, groups similar vertices in a graph into clusters by performing spectral analysis on the matrix representing the graph structure. Pixels in a HSI become vertices in a graph, and edges are created to connect vertices by learning pairwise relationships based on distance metrics.

Gaussian kernel is a widely used metric for similarity measure. The tuning parameter σ in the Gaussian kernel plays a significant role on characterizing the intrinsic local data structure in a graph. In this work, we proposed to automatically estimate this parameter by learning the local point density and co-density so that the edges between two different clusters are significantly weaker than those within each cluster. Also, by utilizing the local density information, a graph-based region merging scheme was proposed to further reduce over segmentation problem.

6.1.3 Graph-based spatial-spectral fusion with composite kernels

In high-resolution HSI, a given pixel can be characterized by spatial and spectral features, extracted separately with different feature mining techniques. Under the self-tuning spectral clustering framework, graph model can be constructed in a spatial and spectral feature domain, respectively. In our work, we use EMAP features to extract the contextual information, including the shape, size and texture information of the structure, to which one pixel belongs. Given the spectral and spatial feature vectors $x_i^{spatial}$, $x_i^{spectral}$, and graph kernels: $K^{spatial}(x_i^{spatial}, x_j^{spatial})$, $K^{spectral}(x_i^{spectral}, x_j^{spectral})$, measured in spatial and spectral feature domain, respectively. Four composite kernel approaches for the joint consideration of spectral and spatial information in a unified self-tuning spectral clustering framework for HSI are introduced. The fused affinity matrix is further transformed into a block-diagonal amplified matrix, named conductivity matrix, to reinforce the connections of the vertices that belong to the same class. By applying the proposal spatial-spectral scheme on chip images from RIT SpecTIR Hyperspectral Airborne Experiment (SHARE) 2012 data campaign, it was demonstrated that the joint spatial-spectral clustering of HSI outperformed other clustering approaches that only use spectral features.

6.1.4 Tensor for pixel/superpixel spatial-spectral HSI classification

Tensor is a high order generalization to vector and matrix. Considering the fact that a gray scale image is intrinsically a matrix, or a second order tensor, tensor representation provides a more sophisticated modeling technique and can be efficiently solved with multilinear algebra. With tensors, the different sources of information, such as the spatial feature, spectral feature, and local structure information extracted for every data point, can be embedded into the multi-dimensional data structure. We introduced tensor-based algorithms for HSI spatial-spectral classification. Spatial-spectral features and neighborhood information can be embedded along different modes in a tensor. A superpixl is a small and coherent cluster that contains several pixels. Instead of representing each superpixl as a vector which lost most of the contextual relationships in it, tensor representation could be conveniently adopted to represent a superpixl. Overall, the local spatial neighborhood structure is well preserved and utilized with tensor representation for pixel-based and superpixl-based image processing.

6.1.5 Tensor for HSI and LiDAR feature fusion

Multilinear algebra is a powerful tool to handle the multi-dimensional data structure. Tensor decomposition and multilinear subspace learning have been used for feature extraction by capturing multilinear structure in higher-order dataset. The HSI data has been explored to generate both spectral and spatial features to contribute to accurate image interpretation. LiDAR sensor, provides more possibilities in measuring different aspects of the objects on the Earth's surface. The HSI and LiDAR fusion can contribute to a more comprehensive interpretation of the ground objects. In our work, for every data point, three sources of features (i.e., the spatial, spectral and LiDAR features) and the local neighborhood information are combined into a third order tensor representation. Two tensor factor analysis based algorithms NTD and NTF, and three multilinear subspace learning based algorithms MPCA, TLPP and TNPE, are introduced for HSI and LiDAR fusion and feature extraction. This is an explorative study of tensor model being used in multi-source information fusion, particularly in the remote sensing area. Given the experimental results on the HSI and LiDAR data acquired over the University of Houston campus and the neighboring urban area on June 22, 2012, we can see the tensor-based algorithms can simultaneously embed the multi-domain features and reduce dimensionality, as well as show good performance in terms of classification accuracy.

6.2 Limitations and Future Work

Though the strong mathematical foundation behind the graph-based spectral clustering and subspace learning algorithms, those algorithms still suffer from some limitations for remotely sensed data clustering and dimensionality reduction. The spectral clustering methods are global in nature, yet the affinity measure is based only on local information. The other limitation with the eigen analysis of graph laplacian is the difficulty in determining the number of eigenvectors for the new subspace that will achieve the best performance. When solving for the eigen problem of large dataset, the matrix consumes a lot of space which is both time consuming and memory inefficient.

The multiple extracted features from spatial, spectral, LiDAR domain are fused into a high order tensor representation for every individual data point. There may exist multiple ways to arrange different features into a tensor. For example, the HSI spatial-spectral second order tensor is formulated by arranging the EMAP feature row vectors (includes the PCA bands that the EMAP features are generated from) associated with the data points from the local neighborhood along the first mode into a second order tensor (matrix). Or, a third order spatial-spectral tensor can be formulated by arranging the EMAP features extracted from the first PCA band as row vectors, and stacking those vectors of the data points from the same local neighborhood along the first mode into a matrix. Afterwards, by concatenating such matrices associated with other PCA bands (i.e., second, third, fourth, etc.), a third order tensor is generated. Obviously, the two different tensor representations consist of the same information but in different tensorial forms. Undoubtedly, how to effectively formulate features into tensorial structure should be taken into consideration for better performance.

Two popular factor analysis based algorithms NTD and NTF are utilized for tensor low rank approximation for feature fusion and dimensionality reduction. Three multilinear mapping algorithms MPCA, TLPP and TNPE, are introduced for remotely sensed multi-source data fusion. More complex multilinear subspace learning algorithms can be investigated in the future. For example, multilinear extensions to other popular graph-based manifold learning techniques should be studied and compared with their linear vector-based counterparts. Besides, there are still many open issues remaining in the computation of the multilinear projection matrices, such as the optimal initialization, the projection order, and the stopping criterion during the iteration stage.

Dimensionality reduction aims to remove redundancy in the original dataset, and to find an alternative representation that implies a better classification condition, if ground truth is provided. When ground truth is available, training stage can be included in dimensionality reduction to learn the class features to generate more concise representations for the input data such that the distances between different classes can be extended. Supervised linear and multilinear dimensionality reduction algorithms can be investigated for remotely sensed data analysis in the future. Under such supervised scheme, the training samples can be fully exploited not only in the data classification, but also in the feature fusion and dimensionality reduction stage.

Besides the applications of tensor-based feature fusion and extraction algorithms reviewed in Chapter 5, there are a wide range of applications dealing with multiple source information fusion in the remote sensing area, where tensor-based algorithms may be useful. Examples include multi-temporal data processing, RGB and spectral image fusion, feature fusion for target detection, and etc.

Bibliography

[1] Gupta, N., "Hyperspectral imager development at army research laboratory," in [*SPIE Defense and Security Symposium*], 69401P–69401P, International Society for Optics and Photonics (2008).

[2] Manolakis, D., Siracusa, C., and Shaw, G., "Hyperspectral subpixel target detection using the linear mixing model," *Geoscience and Remote Sensing, IEEE Transactions on* **39**(7), 1392–1409 (2001).

[3] Dalponte, M., Bruzzone, L., and Gianelle, D., "Fusion of hyperspectral and lidar remote sensing data for classification of complex forest areas," *Geoscience and Remote Sensing, IEEE Transactions on* **46**(5), 1416–1427 (2008).

[4] Liu, H. and Yu, L., "Toward integrating feature selection algorithms for classification and clustering," *Knowledge and Data Engineering, IEEE Transactions on* **17**(4), 491–502 (2005).

[5] Müller, K.-R., Mika, S., Rätsch, G., Tsuda, K., and Schölkopf, B., "An introduction to kernel-based learning algorithms," *Neural Networks, IEEE Transactions on* **12**(2), 181–201 (2001).

[6] Mangai, U. G., Samanta, S., Das, S., and Chowdhury, P. R., "A survey of decision fusion and feature fusion strategies for pattern classification," *IETE Technical review* **27**(4), 293–307 (2010).

[7] Bioucas-Dias, J. M., Plaza, A., Camps-Valls, G., Scheunders, P., Nasrabadi, N. M., and Chanussot, J., "Hyperspectral remote sensing data analysis and future challenges," *Geoscience and Remote Sensing Magazine, IEEE* **1**(2), 6–36 (2013).

[8] Guidi, G., Beraldin, J.-A., Ciofi, S., and Atzeni, C., "Fusion of range camera and photogrammetry: a systematic procedure for improving 3-d models metric accuracy," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on* **33**(4), 667–676 (2003).

[9] Landgrebe, D., "Hyperspectral image data analysis," *Signal Processing Magazine, IEEE* **19**(1), 17–28 (2002).

[10] Brand, M. and Huang, K., "A unifying theorem for spectral embedding and clustering," in [*Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*], (2003).

[11] Lafon, S., Keller, Y., and Coifman, R. R., "Data fusion and multicue data matching by diffusion maps," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **28**(11), 1784–1797 (2006).

[12] Tuia, D., Volpi, M., Trolliet, M., and Camps-Valls, G., "Semisupervised manifold alignment of multimodal remote sensing images," *Geoscience and Remote Sensing, IEEE Transactions on* **52**(12), 7708–7720 (2014).

[13] Van Leeuwen, M. and Nieuwenhuis, M., "Retrieval of forest structural parameters using lidar remote sensing," *European Journal of Forest Research* **129**(4), 749–770 (2010).

[14] Lefsky, M. A., Cohen, W. B., Parker, G. G., and Harding, D. J., "Lidar remote sensing for ecosystem studies," *Bioscience* **52**, 19–30 (01 2002).

[15] Kettig, R. L., "Computer classification of remotely sensed multispectral image data by extraction and classification of homogeneous objects," (1975).

[16] Solberg, A. H. S., Taxt, T., and Jain, A. K., "A markov random field model for classification of multisource satellite imagery," *Geoscience and Remote Sensing, IEEE Transactions on* **34**(1), 100–113 (1996).

[17] Madhok, V. and Landgrebe, D., "Spectral-spatial analysis of remote sensing data: An image model and a procedural design," (1999).

[18] Descombes, X., Sigelle, M., and Preteux, F., "Estimating gaussian markov random field parameters in a nonstationary framework: application to remote sensing imaging," *Image Processing, IEEE Transactions on* **8**(4), 490–503 (1999).

[19] Fauvel, M., Chanussot, J., and Benediktsson, J. A., "A spatial-spectral kernel-based approach for the classification of remote-sensing images," *Pattern Recognition* **45**(1), 381–392 (2012).

[20] Rellier, G., Descombes, X., Falzon, F., and Zerubia, J., "Texture feature analysis using a gauss-markov model in hyperspectral image classification," *Geoscience and Remote Sensing, IEEE Transactions on* **42**(7), 1543–1551 (2004).

[21] Zhang, L., Huang, X., Huang, B., and Li, P., "A pixel shape index coupled with spectral information for classification of high spatial resolution remotely sensed imagery," *Geoscience and Remote Sensing, IEEE Transactions on* **44**(10), 2950–2961 (2006).

[22] ElMasry, G., Wang, N., ElSayed, A., and Ngadi, M., "Hyperspectral imaging for nondestructive determination of some quality attributes for strawberry," *Journal of Food Engineering* **81**(1), 98–107 (2007).

[23] Bau, T. C., Sarkar, S., and Healey, G., "Hyperspectral region classification using a three-dimensional gabor filterbank," *Geoscience and Remote Sensing, IEEE Transactions on* **48**(9), 3457–3464 (2010).

[24] Qian, Y., Ye, M., and Zhou, J., "Hyperspectral image classification based on structured sparse logistic regression and three-dimensional wavelet texture features," *Geoscience and Remote Sensing, IEEE Transactions on* **51**(4), 2276–2291 (2013).

[25] Benediktsson, J. A., Pesaresi, M., and Amason, K., "Classification and feature extraction for remote sensing images from urban areas based on morphological transformations," *Geoscience and Remote Sensing, IEEE Transactions on* **41**(9), 1940–1949 (2003).

[26] Benediktsson, J. A., Palmason, J. A., and Sveinsson, J. R., "Classification of hyperspectral data from urban areas based on extended morphological profiles," *Geoscience and Remote Sensing, IEEE Transactions on* **43**(3), 480–491 (2005).

[27] Dalla Mura, M., Benediktsson, J. A., Waske, B., and Bruzzone, L., "Morphological attribute profiles for the analysis of very high resolution images," *Geoscience and Remote Sensing, IEEE Transactions on* **48**(10), 3747–3762 (2010).

[28] Dalla Mura, M., Atli Benediktsson, J., Waske, B., and Bruzzone, L., "Extended profiles with morphological attribute filters for the analysis of hyperspectral data," *International Journal of Remote Sensing* **31**(22), 5975–5991 (2010).

[29] Wehr, A. and Lohr, U., "Airborne laser scanning?an introduction and overview," *ISPRS Journal of Photogrammetry and Remote Sensing* **54**(2), 68–82 (1999).

[30] Rottensteiner, F., Trinder, J., Clode, S., and Kubik, K., "Fusing airborne laser scanner data and aerial imagery for the automatic extraction of buildings in densely built-up areas," *International Archives of Photogrammetry and Remote Sensing* **35**(B3), 512–517 (2004).

[31] Kaartinen, H., Hyyppä, J., Gülch, E., Vosselman, G., Hyyppä, H., Matikainen, L., Hofmann, A., Mäder, U., Persson, Å., Söderman, U., et al., "Accuracy of 3d city models: Eurosdr comparison," *International archives of photogrammetry, remote sensing and spatial information sciences* **36**(3/W19), 227–232 (2005).

[32] Sampath, A. and Shan, J., "Building boundary tracing and regularization from airborne lidar point clouds," *Photogrammetric Engineering & Remote Sensing* **73**(7), 805–812 (2007).

[33] Zabuawala, S., Nguyen, H., Wei, H., and Yadegar, J., "Fusion of lidar and aerial imagery for accurate building footprint extraction," in [*IS&T/SPIE Electronic Imaging*], 72510Z–72510Z, International Society for Optics and Photonics (2009).

[34] Hu, X., Tao, C. V., and Hu, Y., "Automatic road extraction from dense urban area by integrated processing of high resolution imagery and lidar data," *International*

Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences. Istanbul, Turkey **35**, B3 (2004).

[35] Zhu, P., Lu, Z., Chen, X., Honda, K., and Eiumnoh, A., "Extraction of city roads through shadow path reconstruction using laser data," *Photogrammetric Engineering & Remote Sensing* **70**(12), 1433–1440 (2004).

[36] Pedergnana, M., Marpu, P. R., Mura, M. D., Benediktsson, J. A., and Bruzzone, L., "Classification of remote sensing optical and lidar data using extended attribute profiles," *Selected Topics in Signal Processing, IEEE Journal of* **6**(7), 856–865 (2012).

[37] Sugumaran, R. and Voss, M., "Object-oriented classification of lidar-fused hyperspectral imagery for tree species identification in an urban environment," in [*Urban Remote Sensing Joint Event, 2007*], 1–6, IEEE (2007).

[38] Debes, C., Merentitis, A., Heremans, R., Hahn, J., Frangiadakis, N., van Kasteren, T., Liao, W., Bellens, R., Pizurica, A., Gautama, S., et al., "Hyperspectral and lidar data fusion: Outcome of the 2013 grss data fusion contest," *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of* **7**(6), 2405–2418 (2014).

[39] Jianya, G., Haigang, S., Guorui, M., and Qiming, Z., "A review of multi-temporal remote sensing data change detection algorithms," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* **37**(B7), 757–762 (2008).

[40] Hemissi, S., Farah, I. R., Ettabaa, K. S., and Solaiman, B., "Multi-spectro-temporal analysis of hyperspectral imagery based on 3-d spectral modeling and multilinear algebra," *Geoscience and Remote Sensing, IEEE Transactions on* **51**(1), 199–216 (2013).

[41] Woodcock, C. E., Macomber, S. A., Pax-Lenney, M., and Cohen, W. B., "Monitoring large areas for forest change using landsat: Generalization across space, time and landsat sensors," *Remote Sensing of Environment* **78**(1), 194–203 (2001).

[42] Melgani, F., "Classification of multitemporal remote-sensing images by a fuzzy fusion of spectral and spatio-temporal contextual information," *International Journal of Pattern Recognition and Artificial Intelligence* **18**(02), 143–156 (2004).

[43] Wang, F., "A knowledge-based vision system for detecting land changes at urban fringes," *Geoscience and Remote Sensing, IEEE Transactions on* **31**(1), 136–145 (1993).

[44] Gopal, S. and Woodcock, C., "Remote sensing of forest change using artificial neural networks," *Geoscience and Remote Sensing, IEEE Transactions on* **34**(2), 398–404 (1996).

[45] Bruzzone, L., Prieto, D. F., and Serpico, S. B., "A neural-statistical approach to multitemporal and multisource remote-sensing image classification," *Geoscience and Remote Sensing, IEEE Transactions on* **37**(3), 1350–1359 (1999).

[46] Yang, H. L. and Crawford, M. M., "Manifold alignment for multitemporal hyperspectral image classification," in [*Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*], 4332–4335, IEEE (2011).

[47] Zhang, J., "Multi-source remote sensing data fusion: status and trends," *International Journal of Image and Data Fusion* **1**(1), 5–24 (2010).

[48] Carper, W. J., Lillesand, T. M., and Kiefer, P. W., "The use of Intensity-Hue-Saturation transformations for merging spot panchromatic and multispectral image data," (1990).

[49] Tu, T.-M., Huang, P. S., Hung, C.-L., and Chang, C.-P., "A fast intensity-hue-saturation fusion technique with spectral adjustment for ikonos imagery," *Geoscience and Remote Sensing Letters, IEEE* **1**(4), 309–312 (2004).

[50] Chavez, P., Sides, S. C., and Anderson, J. A., "Comparison of three different methods to merge multiresolution and multispectral data- landsat tm and spot panchromatic," *Photogrammetric Engineering and remote sensing* **57**(3), 295–303 (1991).

[51] Laben, C. A. and Brower, B. V., "Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening," (Jan. 4 2000). US Patent 6,011,875.

[52] Vrabel, J., "Multispectral imagery advanced band sharpening study," *Photogrammetric Engineering and Remote Sensing* **66**(1), 73–80 (2000).

[53] Aiazzi, B., Alparone, L., Baronti, S., and Garzelli, A., "Context-driven fusion of high spatial and spectral resolution images based on oversampled multiresolution

analysis," *Geoscience and Remote Sensing, IEEE Transactions on* **40**(10), 2300–2312 (2002).

[54] Nunez, J., Otazu, X., Fors, O., Prades, A., Pala, V., and Arbiol, R., "Multiresolution-based image fusion with additive wavelet decomposition," *Geoscience and Remote Sensing, IEEE Transactions on* **37**(3), 1204–1211 (1999).

[55] Qian, S.-E., Hollinger, A. B., Williams, D., and Manak, D., "Vector quantization using spectral index-based multiple subcodebooks for hyperspectral data compression," *Geoscience and Remote Sensing, IEEE Transactions on* **38**(3), 1183–1190 (2000).

[56] Du, C.-J. and Sun, D.-W., "Recent developments in the applications of image processing techniques for food quality evaluation," *Trends in Food Science & Technology* **15**(5), 230–249 (2004).

[57] Acito, N., Corsini, G., and Diani, M., "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*], **6**, 3745–3747 vol.6 (July 2003).

[58] Hay, G. J. and Castilla, G., "Geographic object-based image analysis (geobia): A new name for a new discipline," in [*Object-based image analysis*], 75–89, Springer (2008).

[59] Bondy, J. A. and Murty, U. S. R., [*Graph theory with applications*], vol. 290, Macmillan London (1976).

[60] West, D. B. et al., [*Introduction to graph theory*], vol. 2, Prentice hall Upper Saddle River (2001).

[61] Mohar, B., [*Some applications of Laplace eigenvalues of graphs*], Springer (1997).

[62] Basener, B., Ientilucci, E., and Messinger, D., "Anomaly detection using topology," in [*Proc. SPIE*], **6565**, 65650J (2007).

[63] Niyogi, X., "Locality preserving projections," in [*Neural information processing systems*], **16**, 153, MIT (2004).

[64] Bengio, Y., Paiement, J.-F., Vincent, P., Delalleau, O., Le Roux, N., and Ouimet, M., "Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering," *Advances in neural information processing systems* **16**, 177–184 (2004).

[65] Zhang, L., Zhang, L., Tao, D., and Huang, X., "Tensor discriminative locality alignment for hyperspectral image spectral–spatial feature extraction," *Geoscience and Remote Sensing, IEEE Transactions on* **51**(1), 242–256 (2013).

[66] Tenenbaum, J. B., De Silva, V., and Langford, J. C., "A global geometric framework for nonlinear dimensionality reduction," *Science* **290**(5500), 2319–2323 (2000).

[67] Belkin, M. and Niyogi, P., "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural computation* **15**(6), 1373–1396 (2003).

[68] Van der Maaten, L. and Hinton, G., "Visualizing data using t-sne," *Journal of Machine Learning Research* **9**(2579-2605), 85 (2008).

[69] Yan, S., Xu, D., Zhang, B., Zhang, H.-J., Yang, Q., and Lin, S., "Graph embedding and extensions: a general framework for dimensionality reduction," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **29**(1), 40–51 (2007).

[70] Crawford, M. M., Ma, L., and Kim, W., "Exploring nonlinear manifold learning for classification of hyperspectral data," in [*Optical Remote Sensing*], 207–234, Springer (2011).

[71] Shi, J. and Malik, J., "Normalized cuts and image segmentation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* **22**(8), 888–905 (2000).

[72] He, X., Cai, D., and Niyogi, P., "Laplacian score for feature selection," in [*Advances in neural information processing systems*], 507–514 (2005).

[73] Nie, F., Xiang, S., Jia, Y., Zhang, C., and Yan, S., "Trace ratio criterion for feature selection.," in [*AAAI*], **2**, 671–676 (2008).

[74] Hedjam, R. and Cheriet, M., "Hyperspectral band selection based on graph clustering," in [*Information Science, Signal Processing and their Applications (ISSPA), 2012 11th International Conference on*], 813–817, IEEE (2012).

[75] Keshava, N., "A survey of spectral unmixing algorithms," *Lincoln Laboratory Journal* **14**(1), 55–78 (2003).

[76] Winter, M. E., "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*], 266–275, International Society for Optics and Photonics (1999).

[77] Boardman, J. W., Kruse, F. A., and Green, R. O., "Mapping target signatures via partial unmixing of aviris data," in [*Proc. JPL airborne earth sci. workshop*], **1**, 23–26 (1995).

[78] Haralick, R. M., Shanmugam, K., and Dinstein, I. H., "Textural features for image classification," *Systems, Man and Cybernetics, IEEE Transactions on* (6), 610–621 (1973).

[79] Jain, A. K. and Farrokhnia, F., "Unsupervised texture segmentation using gabor filters," in [*Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on*], 14–19, IEEE (1990).

[80] Daugman, J. G., "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *JOSA A* **2**(7), 1160–1169 (1985).

[81] Cross, G. R. and Jain, A. K., "Markov random field texture models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (1), 25–39 (1983).

[82] Kashyap, R. L. and Chellappa, R., "Estimation and choice of neighbors in spatial-interaction models of images," *Information Theory, IEEE Transactions on* **29**(1), 60–72 (1983).

[83] Clausi, D. A., "Comparison and fusion of co-occurrence, gabor and mrf texture features for classification of sar sea-ice imagery," *Atmosphere-Ocean* **39**(3), 183–194 (2001).

[84] Clausi, D., Yue, B., et al., "Comparing cooccurrence probabilities and markov random fields for texture analysis of sar sea ice imagery," *Geoscience and Remote Sensing, IEEE Transactions on* **42**(1), 215–228 (2004).

[85] Soille, P., [*Morphological image analysis: principles and applications*], Springer-Verlag New York, Inc. (2003).

[86] Breen, E. J. and Jones, R., "Attribute openings, thinnings, and granulometries," *Computer Vision and Image Understanding* **64**(3), 377–389 (1996).

[87] Von Luxburg, U., "A tutorial on spectral clustering," *Statistics and computing* **17**(4), 395–416 (2007).

[88] Zelnik-Manor, L. and Perona, P., "Self-tuning spectral clustering," in [*Advances in neural information processing systems*], 1601–1608 (2004).

[89] Fan, L. and Messinger, D. W., "Graph based hyperspectral image segmentation with improved affinity matrix," in [*SPIE Defense+ Security*], 908802–908802, International Society for Optics and Photonics (2014).

[90] Steinbach, M., Karypis, G., Kumar, V., et al., "A comparison of document clustering techniques," in [*KDD workshop on text mining*], **400**(1), 525–526, Boston (2000).

[91] Ng, A. Y., Jordan, M. I., Weiss, Y., et al., "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems* **2**, 849–856 (2002).

[92] Fischer, I. and Poland, J., "Amplifying the block matrix structure for spectral clustering," in [*Proceedings of the 14th annual machine learning conference of Belgium and the Netherlands*], 21–28, Citeseer (2005).

[93] Giannandrea, A., Raqueno, N., Messinger, D. W., Faulring, J., Kerekes, J. P., van Aardt, J., Canham, K., Hagstrom, S., Ontiveros, E., Gerace, A., et al., "The share 2012 data campaign," in [*SPIE Defense, Security, and Sensing*], 87430F–87430F, International Society for Optics and Photonics (2013).

[94] Vasilescu, M. A. O. and Terzopoulos, D., "Multilinear analysis of image ensembles: Tensorfaces," in [*Computer Vision?ECCV 2002*], 447–460, Springer (2002).

[95] He, X., Cai, D., and Niyogi, P., "Tensor subspace analysis," in [*Advances in neural information processing systems*], 499–506 (2005).

[96] Renard, N., Bourennane, S., and Blanc-Talon, J., "Denoising and dimensionality reduction using multilinear tools for hyperspectral images," *Geoscience and Remote Sensing Letters, IEEE* **5**(2), 138–142 (2008).

[97] Kolda, T. G. and Bader, B. W., "Tensor decompositions and applications," *SIAM review* **51**(3), 455–500 (2009).

[98] Hitchcock, F. L., "Multiple invariants and generalized rank of a p-way matrix or tensor," *Journal of Mathematics and Physics* **7**(1), 39–79 (1928).

[99] Tucker, L. R., "Some mathematical notes on three-mode factor analysis," *Psychometrika* **31**(3), 279–311 (1966).

[100] Kiers, H. A. and Smilde, A. K., "Constrained three-mode factor analysis as a tool for parameter estimation with second-order instrumental data," *Journal of Chemometrics* **12**(2), 125–147 (1998).

[101] Carroll, J. D. and Chang, J.-J., "Analysis of individual differences in multidimensional scaling via an n-way generalization of ?eckart-young? decomposition," *Psychometrika* **35**(3), 283–319 (1970).

[102] Phan, A. H. and Cichocki, A., "Tensor decompositions for feature extraction and classification of high dimensional datasets," *Nonlinear theory and its applications, IEICE* **1**(1), 37–68 (2010).

[103] Lu, H., Plataniotis, K. N., and Venetsanopoulos, A. N., "A survey of multilinear subspace learning for tensor data," *Pattern Recognition* **44**(7), 1540–1551 (2011).

[104] Mitra, N. J. and Nguyen, A., "Estimating surface normals in noisy point cloud data," in [*Proceedings of the nineteenth annual symposium on Computational geometry*], 322–328, ACM (2003).

[105] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., and Stuetzle, W., [*Surface reconstruction from unorganized points*], vol. 26, ACM (1992).

[106] Mynatt, I., Bergbauer, S., and Pollard, D. D., "Using differential geometry to describe 3-d folds," *Journal of Structural Geology* **29**(7), 1256–1266 (2007).

[107] Ren, X. and Malik, J., "Learning a classification model for segmentation," in [*Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*], 10–17, IEEE (2003).

[108] Cichocki, A., Zdunek, R., Phan, A. H., and Amari, S.-i., [*Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation*], John Wiley & Sons (2009).

[109] Kim, Y.-D. and Choi, S., "Nonnegative tucker decomposition," in [*2007 IEEE Conference on Computer Vision and Pattern Recognition*], 1–8, IEEE (2007).

[110] Yang, J., Zhang, D., Frangi, A. F., and Yang, J.-y., "Two-dimensional pca: a new approach to appearance-based face representation and recognition," *IEEE transactions on pattern analysis and machine intelligence* **26**(1), 131–137 (2004).

[111] Ye, J., "Generalized low rank approximations of matrices," *Machine Learning* **61**(1-3), 167–191 (2005).

[112] Lu, H., Plataniotis, K. N., and Venetsanopoulos, A. N., "Mpca: Multilinear principal component analysis of tensor objects," *IEEE Transactions on Neural Networks* **19**(1), 18–39 (2008).

[113] "2013 ieee grss data fusion contest." `http://www.grss-ieee.org/community/technical-%20committees/data-fusion/`. Accessed: 2016-11-06.