Dimensionality Reduction using Parallel Independent Component Analysis in Hyperspectral Image Analysis

A Thesis

Presented for the

Master of Science

Department of Electrical and Computer Engineering

The University of Tennessee, Knoxville

Hongtao Du

May 2003

Abstract

Hyperspectral images, although providing abundant information of the object, also bring high computational burden to data processing. This thesis studies the challenging problem of dimensionality reduction in Hyperspectral Image (HSI) analysis. Currently, there are two methods to reduce the dimension: band selection and feature extraction. This thesis presents a band selection technique based on Independent Component Analysis (ICA), an unsupervised signal separation algorithm. Given only the observations of hyperspectral images, the ICA-based band selection picks the independent bands which contain most of the spectral information of the original images.

Due to the high volume of hyperspectral images, ICA-based band selection is a time-consuming process. This thesis develops a parallel ICA algorithm which divides the decorrelation process into internal decorrelation and external decorrelation such that computation burden can be distributed from single processor to multiple processors, and the ICA process can be run in a parallel mode.

Hardware implementation is always a faster and real-time solution to HSI analysis. Until now, there are few hardware designs for ICA-related processes. This thesis synthesizes the parallel ICA-based band selection on Field Programmable Gate Array (FPGA), which is the best choice for moderate designs and fast implementations. Compared to other design syntheses, the synthesis present in this thesis develops three ICA re-configurable components for the purpose of reusability. In addition, this thesis demonstrates the relationship between the design and the capacity utilization of a single FPGA, then discusses the features of High Performance Reconfigurable Computing (HPRC) to accomodate large capacity and design requirements.

Experiments are conducted on three data sets obtained from different sources. Ex-

perimental results show the effectiveness of the proposed ICA-based band selection, parallel ICA and its synthesis on FPGA.

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisor, Professor Hairong Qi, who has given me excellent guidance, beneficial advice and endless support during my study and research.

I would also like to thank Professor Gregory D. Peterson for his advice, suggestion and support in solving many practical problems.

Additionally I would like thank Professor Mongi A. Abidi for serving on my thesis committee.

I am deeply indebted to my beloved wife, Xiaoling Wang, my kind parents, Jixin Du and Xiuting Li, who always give me the strongest love, support and encouragement.

Finally, I express my appreciation to all my friends who have given me a lot of help during my study at UT.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Decades ago, people could only take black and white pictures, which recorded information from single spectral band. Nowadays, color pictures are everywhere. They normally contain information from 3 spectral bands. As technology advances, we can now capture hyperspectral images which compose of several hundred bands. In this chapter, the concepts of spectroscopy, multispectral sensor and hyperspectral sensor are reviewed. We then focus the discussion on some challenging problems presented in the hyperspectral image (HSI) analysis, as well as existing approaches to solving these problems.

1.1 Imaging Spectroscopy

As demonstrated in Fig. 1.1, when a beam of white light is dispersed by passing through a prism, a continuous range of color, the so-called color spectrum, is then formed. Color spectrum is only the visible region of the much wider electromagnetic spectrum (as shown in Fig. 1.2), which contains the entire wavelength range of electromagnetic radi-

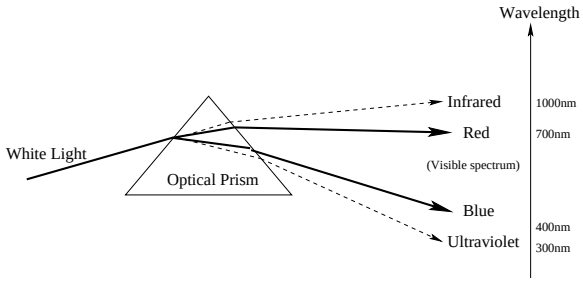


Figure 1.1: Color spectrum seen by passing white light through a prism.

ation extending from gamma rays to the longest radio waves [38]. All objects give off electromagnetic radiation or reflect from another source. Therefore, by detecting and analyzing the energy emitted or reflected from the object, we can obtain an enormous amount of information about the object.

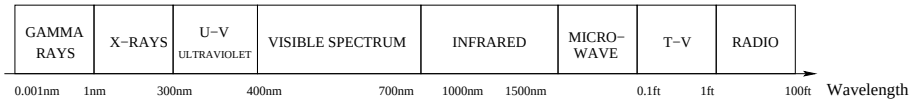


Figure 1.2: Electromagnetic spectrum.

1.1.1 Reflectance and Emission in Spectral Images

An image taken at certain wavelength is composed of reflectance and emission. Reflectance is the percentage of light hitting a material and then reflected by that material. Emission is electromagnetic waves radiated and discharged into the air by an object or substances, such as a smokestack or an automobile gasoline engine [38]. The Kirchhoff's Law gives the relationship between the reflectance and the emission: emissivity = 1- reflectance [32].

Either the reflectance or the emission is used in spectral analysis depending on the category of the object and the environment we observe. For example, the molecular

structure of gases or vapors are observed in emission, while reflectance is used to reveal details of the chemical composition of the surface of solids and liquids [56]. When we use hyperspectral images to analyze earth surface, *spectral reflectance* is the ground feature that we would like to measure using airborne or satellite hyperspectral sensors [51].

Reflectance varies along wavelength for most materials. At certain wavelength, some materials would reflect light and some other would absorb it. Moreover, different materials would have different reflectance percentages at the same wavelength. These reflectance percentages are presented as different gray scales at a single band spectral image, such as the one from AVIRIS sample data shown in Fig. 1.3. If we measure the

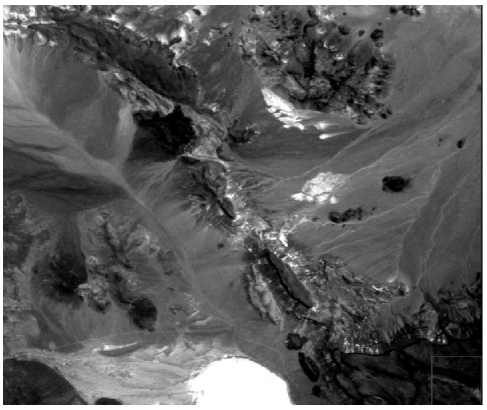
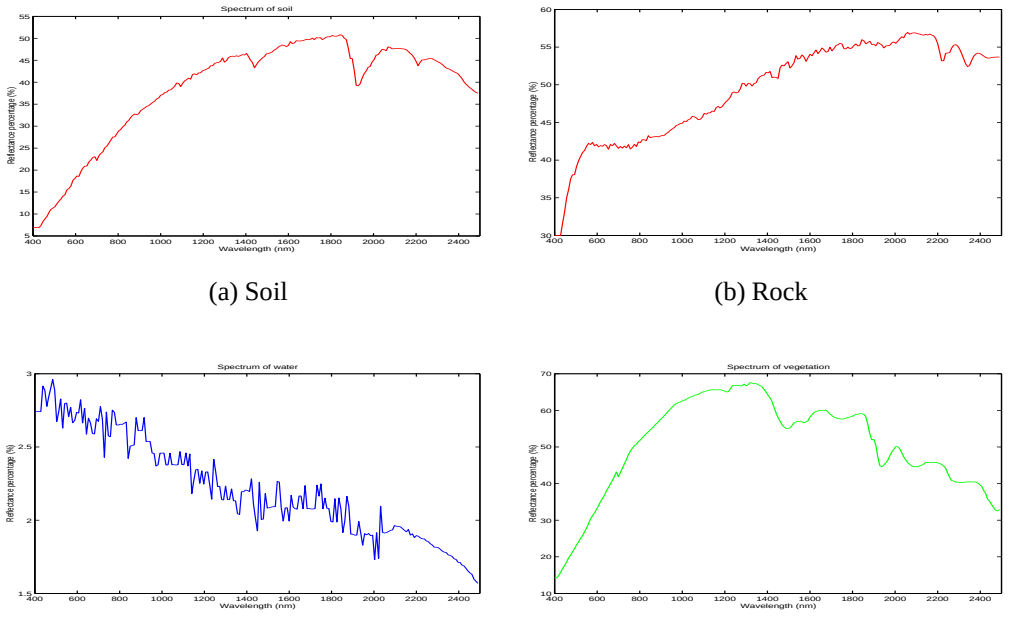


Figure 1.3: AVIRIS sample data: gray scale (single band) [40].

reflectance of a material across a range of wavelengths, we get the *reflectance spectrum* of this material. Figure 1.4 shows the reflectance spectra of four different materials. We can see that the light is absorbed selectively at individual bands by different materials. These bands are called the *absorption bands*. Since different materials have different set of absorption bands, we can use these reflectance variations, such as the shape of reflectance spectrum, the position and strength of absorption bands, to compare the

reflectance spectra for different materials and thus recognize the target made of certain material(s). These variations are also referred to as the *spectral signature*. With the spectral signatures shown in Fig. 1.4, water can be easily recognized from its spectral shape, while vegetation has higher reflectance percentage than soil and rock in the range between 800nm and 1200nm. Although the shape of the reflectance spectrum of soil resembles that of rock, it can be distinguished with the strength of absorption bands in visible range, which spreads from 400nm to 700nm.



(a) Soil (b) Rock

(c) Water (d) Vegetation

Figure 1.4: Reflectance spectra of different materials.

1.1.2 Multispectral Sensors

When we expand our observation from gray scale images to color images, we have been working on the simplest multispectral images. Although human can only perceive

reflectances in the visible range with wavelength from 400nm to 700nm, in a much wider range of the electromagnetic spectrum, multispectral remote sensors can generate images with up to tens of wavelength bands. These multiple reflectance bands are not necessarily contiguous and may overlap.

Multispectral sensor systems typically contain a digital color camera and some sensor filters. They can collect data in either simultaneous mode or sequential mode [55]. In simultaneous mode, images are taken at different spectral bands all at once. In the sequential mode, a spinning filter wheel or a tunable filter in front of a camera is typically used, and images are taken at different spectral bands successively. For example, the Landsat Thematic Mapper (TM) and SPOT XS collect data from four to seven spectral bands in the simultaneous mode. Some popularly used spaceborne and airborne multispectral sensors are listed in Table 1.1.

1.1.3 Hyperspectral Sensors

In the past few years, the airborne-based and satellite-based hyperspectral sensor systems have been developed for many different remote sensing applications. Compared to multispectral sensors, which collect data at a few wide but separated wavelength bands, hyperspectral sensors collect data from a series of narrow and contiguous wavelength bands with a band interval no more than 15nm. Moreover, hyperspectral sensors only collect data in the simultaneous manner. As shown in Table 1.2, most currently used hyperspectral sensors take hyperspectral images of more than one hundred bands.

Compared to multispectral images, hyperspectral images increase our abilities of exploring and identifying resource and targets on the earth surface. The whole hyperspectral image can be interpreted as a hyperspectral cube, which includes a two-

Table 1.1: Multispectral sensors [59].

Sensor	Full Name	Satellites	Organization	Pixel Size (m)	Number of Bands	Wavelength range (nm)
MSS	Multispectral Scanner	Landsat 1-5	EROS Data Center	79	4	500-1,100
TM, ETM	Thematic Mapper and Enhanced Thematic Mapper	Landsat 4,5,7	EROS Data Center	30	7	450-12,500
HRVIR	High Resolution Visible Infrared	SPOT 4	SPOT Image	20	4	500-1,750
HRG	High Resolution Geometric	SPOT 5	SPOT Image	10	4	500-1,750
AVHRR	Advanced Very High Resolution Radiometer	NOAA-9-17	EROS Data Center	1,100	5	580-12,400
QuickBird			DigitalGlobe	2.44	4	450-900
IKONOS			Space Imaging	4	4	450-900
MODIS	Moderate Resolution Imaging Spectrometer	EOS Terra and Aqua	NASA Earth Observing System Data Gateway	250-1,000	36	405-14,385
ASTER	Advanced Spaceborne Thermal Emission and Reflection Radiometer	EOS Terra	NASA Earth Observing System Data Gateway	12-90	14	520-11,650
MISR	Multiangle Imaging Spectro Radiometer	EOS Terra	NASA Earth Observing System Data Gateway	275	4	446-867
SeaWiFS	Sea-viewing Wide Field-of-view Sensor	OrbView-2	NASA GES DAAC or ORBIMAGE	1,100-4,500	8	402-885

dimensional image and a third index of band. For example, a hyperspectral image taken by AVIRIS on an ER-2 plane over Moffett Field, California [42] is displayed as a hyperspectral cube in Fig. 1.5. The top band of the cube is in the visible range of the spectrum and the bottom band of the cube is in the infrared range.

1.1.4 Applications of Hyperspectral Images

Hyperspectral imagery has been used to detect and identify a wide variety of materials with characteristic reflectance spectra, including classification of agriculture targets [53], underwater objects [54], buried land mines [30], military buildings [24], etc.

Table 1.2: Hyperspectral sensors [40, 55, 59].

Sensor	Full Name	Organization	Number of Bands	Wavelength range (μm)
AVIRIS	Airborne Visible/Infrared Imaging Spectrometer	NASA Jet Propulsion Lab, USA	224	0.4 - 2.5
KODAK		CIS Johns Hopkins University	31	0.4 - 0.7
AISA	Airborne Imaging Spectrometer for Applications	Spectral Imaging Ltd. Finland	288	0.43 - 1.0
CASI	Compact Airborne Spectrographic Imager	ITRES Research Limited Canada	288	0.4 - 1.0
CHRIS	Compact High Resolution Imaging Spectrometer	European Space Agency	N/A	0.45 - 1.05
DAIS2115	Digital Airborne Imaging Spectrometer	GER Corp. USA	211	0.43 - 12.0
FTHSI	Fourier-Transform Visible Hyperspectral Imager Satellite-based: MightySat II	Air Force Research Lab designed by Kestrel Corp.	256	0.35 - 1.05
HYMAP		Integrated Spectronics Australia	128	0.4 - 2.45
Hyperion	Satellite-based on EO-1	NASA Goddard Space Flight Center, USA	220	0.4 - 2.5
MIVIS	Multispectral Infrared and Visible Imaging Spectrometer	SenSyTech	102	0.4 - 2.5
PROBE-1		Earth Search Sciences Inc. USA	128	0.4 - 2.5
SFSI	Short Wavelength Infrared Full Spectrum Imager	Canadian Centre for Remote Sensing	120	1.2 - 2.4
TRWIS III	TRW Imaging Spectrometer	TRW Inc.	384	0.38 - 2.45

For earth resource research, hyperspectral images are used by geologists for material mapping to identify material category [16]. Given enough spectral range, spectral resolution, signal to noise, and spatial resolution, Clark et al. develop an analysis system called Tricorder to determine material categories such as mineral, vegetable and liquid. This system simultaneously maps minerals using multiple spectral features (e.g. absorption bands, spectra shape) [16]. Ben-Dor et al. use data acquired from the hyperspectral sensor DAIS-7915 over Israel Valley in northern Israel to detect soil properties such as soil field moisture, organic content, soil saturated moisture and soil salinity [6]. Their objective is to use the Visible and Near Infrared Analysis (VNIRA) approach to generate an empirical model which predicts the soil property from wet chemistry and

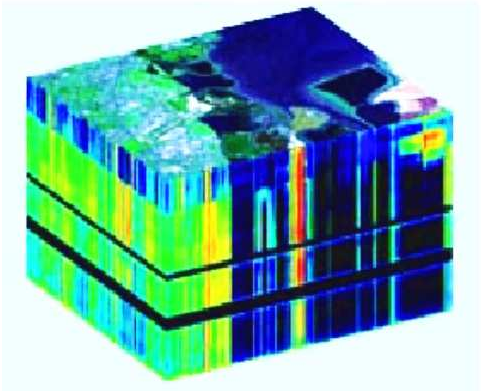


Figure 1.5: A hyperspectral image cube, taken by AVIRIS on an ER-2 plane over Moffett Field, CA [42].

spectral information of a representative sample set. With NASA's AVIRIS data sets, taken from the Harvard Forest in central Massachusetts and Blackhawk Island in south-central Wisconsin, Aber and Martin [1] identify spectral regions correlated with foliar chemistry at the canopy level in forests. To achieve real-time recognition of underwater objects in the hyperspectral data sets taken by remotely sensing and imaging ocean waters, Stein et al. [54] use the spectral matched filter (SMF) to select spectral bands that optimize the signal-to-noise ratio (SNR) losses.

For military applications, hyperspectral images have been used to detect different military targets in a variety of environments. The U.S. Army Hyperspectral Mine Detection Phenomenology (HMDP) program evaluates the advantage of the spectral discriminants in the land mines detection [30]. They first identify the spectral features from the significant mine signature data in the range between 350nm and 14,000nm. The detection metrics are then used and the detection performance is analyzed against different background types, age of buried mines, etc. Moreover, they intend to utilize Electro-Optics and Infrared (EO/IR) hyperspectral sensor to practically detect buried

land mines. In another example, Huertas et al. employ panchromatic (PAN) images to construct 3-D building models in the automated building detection and description system [24]. Using the thematic maps provided by the hyperspectral sensor HYDICE, they fulfill the accurate delineation and improve the efficiency and quality of the detection system.

1.2 Hyperspectral Image (HSI) Analysis

1.2.1 Existing Challenges

Hyperspectral images contain a large amount of valuable data, but interpreting them also presents unique challenges, which can be summarized as:

- the high dimensionality challenge,
- the classification accuracy challenge, and
- the pixel pureness challenge.

For target recognition in hyperspectral imaged, high dimensionality is both an advantage and a disadvantage. With more dimensions, we can obtain more feature information in more detail, therefore detect targets and classify materials with potentially higher accuracy. However, the high dimensionality inversely gives a big processing burden to further analysis. Taking the hyperspectral image collected from NASA AVIRIS as an example, the spatial size of a single band image is 614×512 in pixel, which accounts for a file size of 629Kb if each pixel uses 16 bits. However, for the total 224 bands, the file size explodes to 140.8Mb. Applying any classification algorithm to such

a big data set would consume significant computing time. Fortunately, in many cases, it is unnecessary to process all the spectral bands of a hyperspectral image. Most materials have specific characters only at certain bands, thus leaving all the rest reflectance readout somewhat redundant. Therefore, dimensionality reduction is an important preprocessing step in HSI analysis. However, dimensionality reduction itself is a time-consuming process.

On the other hand, with the dimensionality of hyperspectral image increasing, the number of training samples for each class must accordingly increase in order to achieve high classification accuracy [29]. The relationship between the recognition accuracy and the measurement complexity is demonstrated in Fig. 1.6, where the measurement complexity is related to the numbers of spectral bands, and the number of training samples (m). With the measurement complexity increasing, for a given number of training

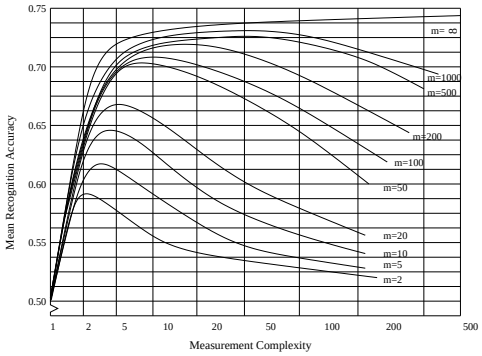


Figure 1.6: The relationship between the recognition accuracy and the measurement complexity [35].

samples, the recognition accuracy climbs to a maximum and then descends eventually. In another word, insufficient number of training samples hinders the recognition in high dimensionality. If the number of training samples grows larger, the recognition accuracy is accordingly increased [35]. However, the maximal recognition accuracy could

not approach to a high rate without proper consideration of pixel pureness.

Taken with airborne or satellite hyperspectral sensors, hyperspectral images possess resolutions at normally tens meters level (AVIRIS, Hyperion, FTHSI, etc) [41, 59], where multiple different materials could co-exist. Therefore, it is common that one pixel in a hyperspectral image presents a mixture of the reflectance spectra of several objects or materials. How to extract the spectrum of interest from the mixture or how to achieve sub-pixel recognition accuracy remains a challenging problem.

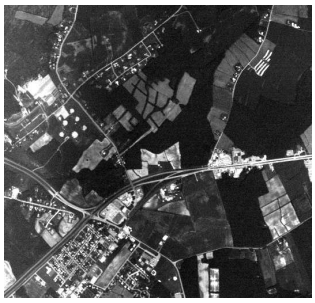
1.2.2 HSI Analysis Space

To explore solutions to the existing challenges, the hyperspectral image analysis is conducted in three spaces: image space, spectral space and feature space [35, 37], as shown in Fig. 1.7.

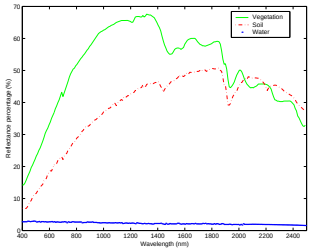
In the image space, pixels are demonstrated in geometric relationship to one another according to individual wavelength. At a given band, the corresponding reflectance value of each pixel is represented in gray scale, such that different materials can be distinctively displayed in an image.

In the spectral space, for each pixel, the reflectance values at individual observation bands are represented as a function of wavelength. Therefore, we can analyze the absorption bands and the spectral shapes to recognize specific materials.

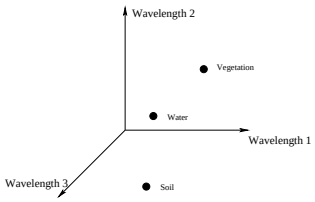
In the feature space, all observation bands or extracted features first construct a multidimensional space with each individual band/feature as one axis, pixels and training samples are then displayed as points in this space. Consequently, we can classify materials or recognize objects of interest by measuring distance.



(a) Image Space



(b) Spectral Space



(c) Feature Space

Figure 1.7: HSI analysis space [35].

1.3 Previous Work in HSI Analysis

1.3.1 Dimensionality Reduction

To solve the high dimensionality problem and thus improve classification accuracy, many dimensionality reduction methods have been developed for different data sets and application backgrounds. These methods can be divided into two categories, band selection and feature extraction.

As demonstrated in Fig. 1.8, the intention of feature extraction is to extract features from the hyperspectral data set and form a feature space with considerably low dimension. In essence, feature extraction is a projection process where the high dimensional hyperspectral data set is projected onto a lower-dimensional feature space. The pro-

jection, although changes the physical meaning of each spectral band, should preserve most information that the original set contains. For example, Principal Component

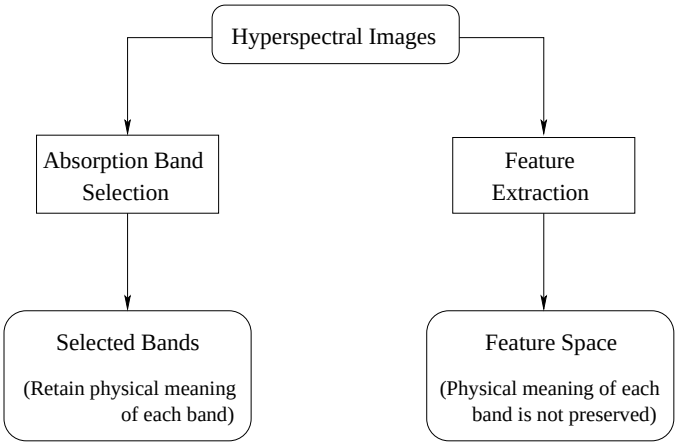


Figure 1.8: Dimensionality reduction in HSI analysis.

Analysis (PCA) is one of the commonly used schemes in feature extraction. The objective of PCA is to project the original data to a feature space which can be represented as the combination of basis vectors. These basis vectors are derived from the eigenvectors corresponding to the maximum eigenvalues of the covariance matrix. The eventual feature space is optimal for signals with Gaussian noise. As another example, Jimenez and Landgrebe [37] use Projection Pursuit (PP) to extract a lower-dimensional space. PP is a technique that automatically picks a maximally effective, lower-dimensional projection from high-dimensional data by optimizing a function called the projection index. Given the initial data set $\mathbf{X}$, dimensionality reduced data set $\mathbf{Y}$, and the parametric orthonormal matrix $\mathbf{A}$, where $\mathbf{Y} = \mathbf{A}^T\mathbf{X}$, PP computes A by optimizing the projection index $I(\mathbf{A}^T\mathbf{X})$. Lennon et al. [36] use Independent Component Analysis (ICA), a multivariate data analysis process, to represent hyperspectral images. Although closely related to PP, ICA simultaneously looks for the components and find the

directions where all the projected components are the most independent in the sense of negentropy, which we will discuss in detail in Chapter 2.

Compared to feature extraction, band selection reduces the high dimension by capturing the absorption bands, a subset of the original set of spectral bands that characterize most features of the spectral profile without changing its physical property. For instance, Vélez-Reyes et al. [58] first use singular value to rank the QR matrix factorizations, where Q is a matrix of orthonormal columns and R is an upper triangular matrix. Then they use these matrix factorizations to select the most independent columns, which correspond to the most independent bands containing the maximum spatial information. As another example, in the spectral space, Keshava [31] quantifies the distance between the spectra of two materials at corresponding spectral bands, and then analyzes the separability of these two spectra using the Spectral Angle Mapper (SAM) metric. The spectral bands are selected in order to maximize the SAM metric, and thus maximize the angle between the two spectra.

1.3.2 Classification

Besides the high dimensionality problem, the classification accuracy challenge also spurs researchers applying novel classification techniques to hyperspectral images. For example, Gualtieri et al. [23] use Support Vector Machine (SVM) to classify high-dimensional data without reducing dimensionality. The intention of SVM is to find the support vectors through training data, then form the separating surfaces with the support vectors, where the separating surfaces are the boundaries of classes. SVM first maps data to a high dimensional space with a non-linear transformation, then try to find a linear separating surface between two classes. Therefore, the non-linear separating

surfaces in the original data space can be found as linear separating surfaces in higher dimensional space. As another example, Robila et al. [48] use ICA to detect small man-made targets. They prepossess the hyperspectral data through whitening and sphering, which eliminates the first and second order statistics respectively. Then they apply ICA and sort the resulting bands according to kurtosis, which is the normalized fourth order moment of the sample distribution. Finally, they identify the targets from the high kurtosis valued bands which indicate the presence of small man-made targets.

For different classification algorithms in HSI analysis, Landgrebe has summarized the general procedure in [33, 34].

1.3.3 Pixel Pureness

To settle the pixel pureness challenge in HSI analysis, many techniques for separating the complex combinations found in the mixed pixels have been presented. Most of these approaches belong to one of the two categories: projection and linear spectral unmixing.

The projection techniques project the mixed pixels onto another space and then extract desired information. For example, Chiang et al. [14] assume the image background can be modeled by a Gaussian distribution, then employ skewness and kurtosis to design a projection index and detect small target in an unknown image scene. Harsanyi and Chang [13] first project the pixel vectors onto a subspace which is orthogonal to the undesired signatures, then null the interfering signatures and project the residual onto a signature of interest. Hence, the signal-to-noise ratio is maximized and the results thus best represent the interested target.

Compared to the projection, the linear spectral unmixing utilizes a linear mixture

model to estimate the fractions of the signatures within a mixed pixel [12]. For example, Bowles et al. [10] use the filter vector, the estimation of material concentration based on the reflectance variations, to demix complex mixture patterns. This process achieves fast computing speed on well-known materials, therefore making real time implementation possible. Theo et al. [50] use the fuzzy classification to determine the presence and abundance of the basic spectra in a measured spectrum, thus obtain the fractions of the materials presenting in each pixel. As another example, Bayliss et al. [5] assume that the spectral information of different materials are close to statistically independent, and apply ICA to find pure materials or some known mixtures, such as Halloysite + Kaolinite.

1.4 Thesis Contributions

This thesis focuses on the problem of band selection in HSI analysis. In the thesis work, we use ICA to estimate a weight matrix concerning the independent components and the spectral bands. We then compare the weights of individual spectral bands and select the most independent bands which contain the maximum information. This procedure is called the ICA-based band selection. As a significant benefit, ICA-based band selection retains the most of features of the spectral profile given only the observations of hyperspectral images.

ICA algorithm divides the decorrelation process into internal decorrelation and external decorrelation Although powerful, ICA is a time-consuming process. To speed up the ICA-based band selection, we develop a parallel ICA algorithm which divides the decorrelation process into internal decorrelation and external decorrelation, which

decorrelate independent components from the same processor and those from other processors respectively, such that computation burden can be distributed from single processor to multiple processors, and the ICA process can be run in a parallel mode.

Hardware implementation is a faster and real-time solution to HSI analysis. Until now, there are few hardware designs for ICA-related processes. In this thesis, we synthesize the parallel ICA-based band selection on Field Programmable Gate Arrays (FPGA), which will be described in detail in Chapter 4. Compared to some other design syntheses, in our synthesis procedure, we develop three ICA re-configurable components for the purpose of reusability. In addition, we demonstrate the relationship between the design and the capacity utilization of a single FPGA, then discuss the features of High Performance Reconfigurable Computing (HPRC) to accomodate large capacity and design requirements.

1.5 Thesis Outline

The thesis outline is illustrated in Fig. 1.9.

Chapter 2 describes the Independent Component Analysis (ICA), then presents the ICA-based band selection and parallel ICA algorithm.

Chapter 3 reviews both unsupervised classification (k-means, winner-take-all, Kohonen maps) and supervised classification (kNN), then discusses how to identify mixture pixels.

Chapter 4 first illustrates features of Field Programmable Gate Arrays (FPGAs) and High Performance Reconfigurable Computing (HPRC). It then describes the synthesis procedure on FPGA using re-configurable components in VHDL.

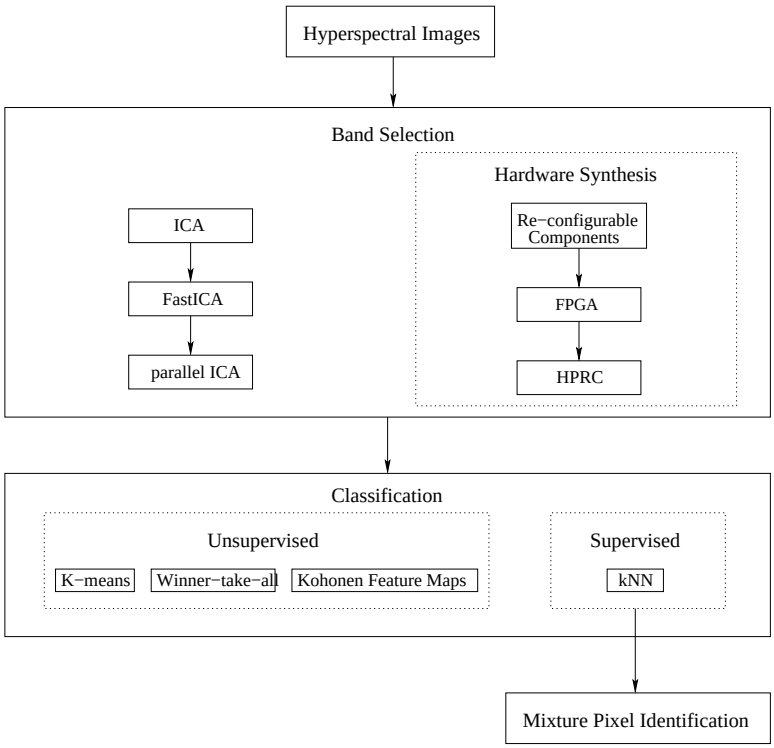


Figure 1.9: Thesis outline.

Chapter 5 describes data sets of one multispectral image set and two hyperspectral image sets used in this thesis. It then presents the spectral bands picked by the ICA-based band selection, and results of unsupervised classifications and supervised classification. Finally, this chapter shows the synthesis simulations on an FPGA, Xilinx V1000EHQ240-6.

Chapter 6 summarizes the algorithms and methods developed in this thesis, and discusses future work in HSI analysis and FPGA / HPRC implementation.

Chapter 2

Independent Component Analysis based Band Selection

As pointed out in chapter 1, one of the biggest challenges of hyperspectral image analysis is its high dimensionality. Although informative, many of the spectral bands provide redundant information. In order to save computation time and achieve high classification accuracy, the dimension should be reduced before any further processing. There are two methods to reduce the dimension: by selecting the absorption bands and by extracting features. In this chapter, we apply a linear separation technique, independent component analysis (ICA), to hyperspectral image band selection. In addition, we develop a parallel ICA algorithm to distribute the computation burden caused by the high dimensionality in hyperspectral data set.

2.1 Independent Component Analysis (ICA)

ICA is a technique that searches for a linear transformation which minimizes the statistical dependence between components [17]. ICA is first proposed by Pierre Comon [17] in 1994, and has been used in a variety of applications, including blind source separation [25], feature extraction [36], recognition [4], etc.

2.1.1 Independent Component Analysis

Suppose m is the number of source signals and n is the number of observed signals, where the observed signals are linear mixtures of the source signals. Then, the observed signal $\mathbf{X}$, where $\mathbf{X} = (\mathbf{x}_1, \cdots, \mathbf{x}_n)^T$, and the source signal $\mathbf{S}$, where $\mathbf{S} = (\mathbf{s}_1, \cdots, \mathbf{s}_m)^T$, are represented in the following ICA model,

$$\mathbf{X} = \mathbf{AS} \tag{2.1}$$

where $\mathbf{A} = [\mathbf{a}_1, \cdots, \mathbf{a}_m]$ is a mixing matrix, and $\mathbf{a}_i = [a_{1i}, \cdots, a_{ni}]^T$, $i = 1, \cdots, m$.

If the mixing matrix is estimated, the unmixing matrix, which is the inverse of the mixing matrix, can then be used to separate the independent components from the observations, as shown in Eq. 2.2

$$\mathbf{S} = \mathbf{WX} \tag{2.2}$$

where $\mathbf{W} = \mathbf{A}^{-1}$.

To estimate the mixing matrix in the ICA model, the components $\mathbf{s}_i$ are assumed to be statistically independent and not Gaussian distributed. The procedure of estimating

independent components (ICs) $\mathbf{s}_i$ is actually a process of decorrelation. If information on $\mathbf{s}_i$ does not give any information on the other components, then $\mathbf{s}_i$ is considered independent of these components. According to the assumption of nongaussian distribution, the desired independent components $\mathbf{s}_i$ contains the least Gaussian components.

A measure of nongaussianity is the key for estimating the unmixing matrix and therefore the independent components. The classical measure of nongaussianity is kurtosis, which is the fourth order statistics and has zero value for Gaussian distribution, as shown in Eq. 2.3.

$$kurt(y) = E\{y^4\} - 3(E\{y^2\})^2 \tag{2.3}$$

The random variables with negative kurtosis are called subgaussian and those with positive kurtosis are called supergaussian [27].

However, kurtosis is sensitive to outliers. Because a Gaussian variable has the largest entropy among all random variables of equal variance [19], negentropy can be used as a measure of nongaussianity. The negentropy is defined as Eq. 2.4,

$$J(\mathbf{X}) = H(p_{\mathbf{X}gauss}) - H(p_{\mathbf{X}}) \tag{2.4}$$

where $H(p_{\mathbf{X}gauss})$ is the entropy of a Gaussian random variable with the same covariance matrix as $\mathbf{X}$, $H(p_{\mathbf{X}})$ is the differential entropy and is defined as Eq. 2.5,

$$H(p_{\mathbf{X}}) = -\int p_{\mathbf{X}}(u) \log p_{\mathbf{X}}(u) du \tag{2.5}$$

where $p_{\mathbf{X}}(u)$ is the probability density function of $\mathbf{X}$.

Since the negentropy is difficult to compute, an approximation is used instead [26].

$$J(\mathbf{X}) \approx \{E[G(\mathbf{X})] - E[G(\mathbf{X}_{gauss})]\}^2 \tag{2.6}$$

where $G(\mathbf{X})$ is a non-quadratic function. If the $G(\mathbf{X})$ is chosen not to grow too fast, the estimation would be robust. It has been proved that the following two forms of $G(\mathbf{X})$ are very useful.

$$G(\mathbf{X}) = \frac{1}{a_1} \log \cosh a_1 \mathbf{X} \tag{2.7}$$

where $1 \leq a_1 \leq 2$, or

$$G(\mathbf{X}) = -exp(-\frac{\mathbf{X}^2}{2}) \tag{2.8}$$

2.1.2 FastICA Algorithm

For practical implementation, Hyvärunen [26] has developed the so-called FastICA algorithm, which is claimed to be the fastest practical ICA method so far, to maximize the objective function shown in Eq. 2.6. The FastICA algorithm consists of two processes: the one unit process and the decorrelation process. As illustrated in Fig. 2.1, the one unit process includes:

1. Initialize the weight vector $\mathbf{w}_i$.

2. Update $\mathbf{w}_i$ by

$$\mathbf{w}_i^+ = E\{\mathbf{X} g(\mathbf{w}_i^T \mathbf{X})\} - E\{g'(\mathbf{w}_i^T \mathbf{X})\}\mathbf{w}_i \tag{2.9}$$

   where $g(.) = \tanh(.)$.

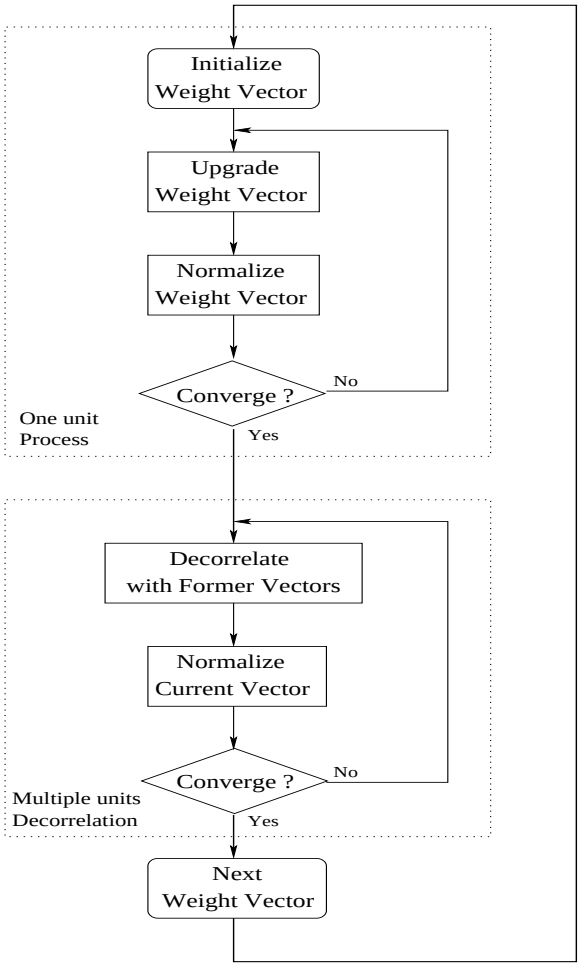


Figure 2.1: Flow of FastICA algorithm.

3. Normalize $\mathbf{w}_i$,

$$\mathbf{w}_i = \mathbf{w}_i^+ / ||\mathbf{w}_i^+|| \tag{2.10}$$

4. Goto 2, until $\Delta\mathbf{w}_i = (\mathbf{w}_i - \mathbf{w}_i^+)^2$ is less than a small number.

The one unit process is only used to estimate one IC. To estimate several ICs, the decorrelation process is used to avoid different weight vectors converging to the same maximal. Given the $(p+1)^{th}$ weight vector is estimated by one unit process, its decor-

relation process includes:

1. Update the weight vector $\mathbf{w}_{p+1}$ by,

$$\mathbf{w}_{p+1}^{+} = \mathbf{w}_{p+1} - \sum_{i=1}^{p} \mathbf{w}_{p+1}^{T} \mathbf{w}_i \mathbf{w}_i \tag{2.11}$$

   where $\mathbf{w}_1, \cdots, \mathbf{w}_p$ are the already decorrelated weight vectors.

2. Normalize $\mathbf{w}_{p+1}$,

$$\mathbf{w}_{p+1} = \mathbf{w}_{p+1}^{+} / ||\mathbf{w}_{p+1}^{+}|| \tag{2.12}$$

3. Goto 1, until $\Delta \mathbf{w}_{p+1} = (\mathbf{w}_{p+1} - \mathbf{w}_{p+1}^{+})^2$ is less than a small number.

2.2 ICA based Band Selection

We have discussed in Chapter 1 that ICA has been used to reduce high dimensionality by projecting the original hyperspectral image to a lower dimensional space. In this projection, the number of observed signals n is the original dimensionality and the number of source signals m is the low dimensional space corresponding to the number of materials existing in the hyperspectral image. Then the spectral profile of all pixels in the hyperspectral images are treated as the observed signal $\mathbf{X}$ and used to estimate the weight matrix $\mathbf{W}$. Subsequently, the resulting weight matrix is used in ICA projection $\mathbf{S} = \mathbf{W}\mathbf{X}$, where $\mathbf{S}$ is the source signal with lower dimensionality, and each independent component $\mathbf{s}_i$ is distinctive for one material. Using this method, Lennon et al. [36] reduce the dimensionality of the CASI data from 14 to 3, then detect wood from different types of soil occupation and roads. Chiang et al. [15] reduce the dimen-

sionality of the AVIRIS data from 224 to 158, and distinguish cinders, rhyolite, playa, vegetation and shade in one hyperspectral image.

However, in most cases, we do not know the number of materials existing in a hyperspectral image. In this section, we evaluate the weight matrix $\mathbf{W}$ to observe how each original individual band contributing to the transformation described above. Therefore, we obtain the independence of all the original bands, and select the most independent bands. According to the selected bands, we generate a new spectral image, which is a subset of the original hyperspectral image, and therefore achieve the purpose of dimensionality reduction.

Given the condition that we do not know the materials categories in an n-band hyperspectral image, we assume the number of materials as m and obtain the corresponding weight matrix $\mathbf{W}_{m\times n}$ through ICA. When we detect the source $\mathbf{S}$ (pure materials) from the observation $\mathbf{X}$ (pixels in hyperspectral image) with the weight matrix $\mathbf{W}$, the k^{th} element s_{ik} in the independent component $\mathbf{s}_i$ is obtained by

$$s_{ik} = \sum_{j=1}^{n} w_{ij} x_{jk} \tag{2.13}$$

where $i = 1 \cdots m$, and w_{ij} denotes the weight of the j^{th} band regarding to the component $\mathbf{s}_i$. In another word, w_{ij} shows how much information is included in the j^{th} band in regard to the i^{th} material. Following the same way, we can estimate the importance of each individual band corresponding different materials.

Considering our given condition, we calculate the average absolute weight coeffi-

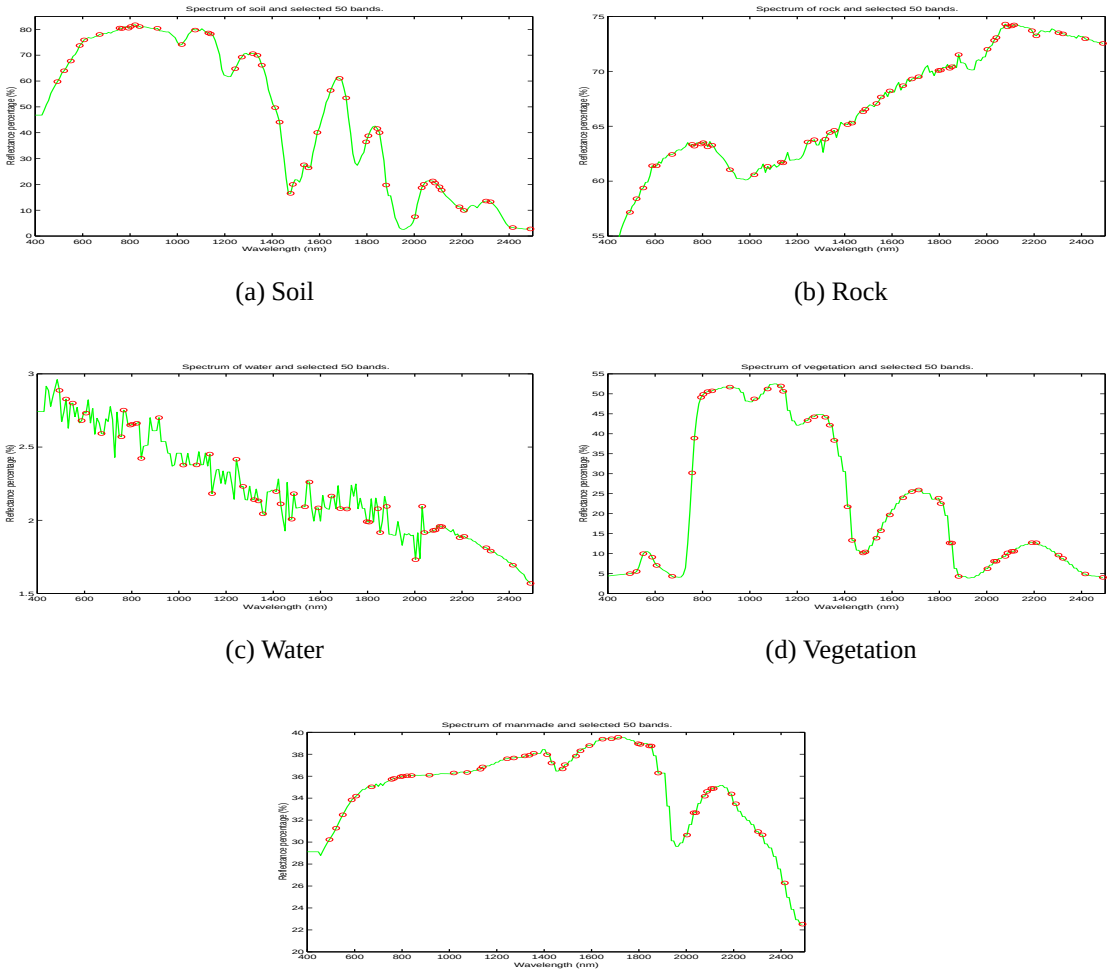
cient of each individual band for all materials, as shown in Eq. 2.14,

$$\bar{w_j} = \frac{1}{m}\sum_{i=1}^{m}|w_{ij}| \tag{2.14}$$

where $j = 1 \cdots n$. We then obtain a band independence sequence by sorting the average absolute weight coefficients for all bands. In this sequence, the bands with higher average absolute weight coefficients contribute more to the ICA transformation than other bands do. In another word, these bands contain more spectral information than other bands. From the band independence sequence, we then select those bands with highest average absolute weight coefficients. As typical multispectral analysis methods treating each spectral band as an independent variable, we call these selected bands as *independent bands* which are no longer contiguous. The independent bands contains most information of the original hyperspectral image.

For example, we apply the ICA-based band selection to a 224-band synthetic hyperspectral image which will be described in detail in chapter 5. This hyperspectral image contains spectral profiles of both pure materials and mixtures. We select 50 independent bands and illustrate them respectively on the spectra of 5 pure materials: soil (white gypsum dune sand), rock (aplite), water (tap water), vegetation (conifer) and manmade material (construction concrete), in Fig. 2.2. It demonstrates that the selected independent bands on the spectra curves contain the most important information including maximum, minimum and inflection points, thus retain most spectral information such as absorption bands and spectra shapes.

In addition, when we make the assumption about number of different materials m, if the assumed m is larger than the ground truth, certain material would be separated



(a) Soil

(b) Rock

(c) Water

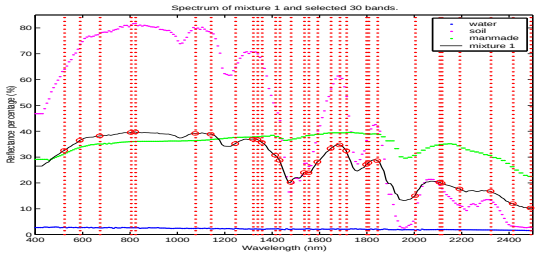
(d) Vegetation

(e) Manmade

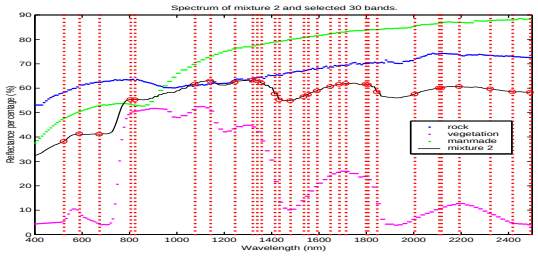
Figure 2.2: 50 independent bands on the spectra of 5 pure materials from synthetic hyperspectral image.

into two kinds or more, whereas if the assumed m is less than the ground truth, some components would contain mixtures of different materials. Because we intend to select independent bands for all materials, the ICA-based band selection calculates the average absolute weight coefficient of each individual band for all materials, pure materials or mixtures.

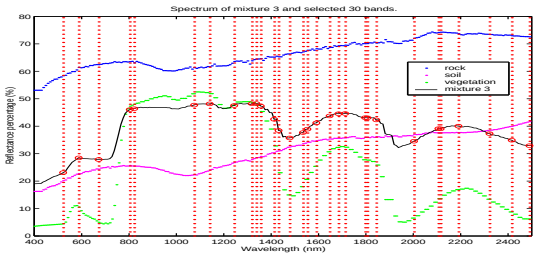
For the 224-band synthetic hyperspectral image, the selected 30 independent bands are illustrated respectively on the spectra of 3 mixtures in Fig. 2.3, where the dark solid curves are the spectra of the mixtures and the color dotted curves denote the composing materials of the corresponding mixtures. It shows that the selected independent bands for the mixtures consist of the independent bands of composing materials of the mixtures.



(a) Mixture 1: water, soil, manmade



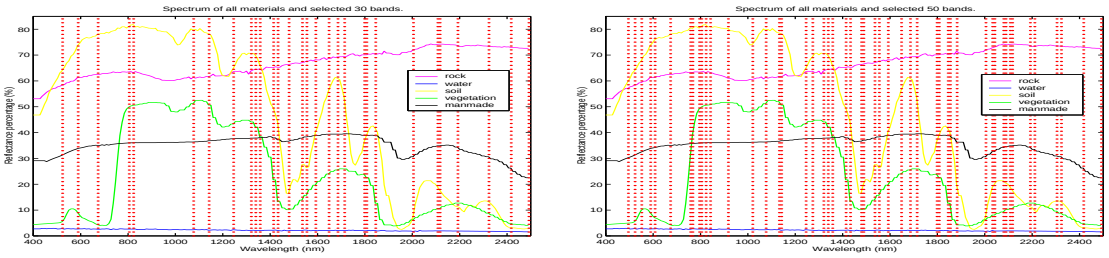
(b) Mixture 2: rock, vegetation, manmade



(c) Mixture 3: rock, soil, vegetation

Figure 2.3: 30 independent bands on the spectra of 3 mixtures from synthetic hyperspectral image.

Using the ICA-based band selection, the dimensionality can be reduced to an specified lower number. However, if the number of independent bands we choose is too small, we might lose some information of certain material. For the 224-band synthetic hyperspectral image as an example, we respectively select 30 and 50 independent bands and show them as vertical dashed lines in Fig. 2.4 on the 5 pure materials: soil, rock, water, vegetation and manmade material. It illustrates that the set of 30 independent bands loses some information about soil, rock and vegetation between 800nm and 1200nm, where the lost information is expressed as minimum and inflection points on the spectra.



(a) 30 independent bands

(b) 50 independent bands

Figure 2.4: Effects of different numbers of independent bands on 5 pure materials from synthetic hyperspectral image.

2.3 Parallel ICA

Due to the high volume of the hyperspectral data sets, the process of ICA-based band selection is a time-consuming procedure. To calculate the weight matrix more efficiently, in this section we design a parallel ICA algorithm based on FastICA. The basic idea of parallel ICA is to estimate the weight vectors using the one unit process in

parallel mode, and perform the decorrelation at two levels: internal decorrelation and external decorrelation.

In parallel ICA, the weight matrix $\mathbf{W}$ is first divided into several sub-matrices with different numbers of weight vectors,

$$\mathbf{W} = (\mathbf{W}_1, \cdots, \mathbf{W}_i, \cdots, \mathbf{W}_k)^T$$

where $\mathbf{W}_i = (\mathbf{w}_{i1}, \cdots, \mathbf{w}_{ij}, \cdots, \mathbf{w}_{in_i})^T$, $\mathbf{w}_{ij}$ denotes the j^{th} weight vector in the i^{th} sub-matrix, n_i is the number of weight vectors in sub-matrix $\mathbf{W}_i$, and the total number of weight vectors $n = n_1 + \cdots + n_i + \cdots + n_k$.

The sub-matrices $\mathbf{W}_i$'s can be estimated in a parallel manner. Eqs. 2.15 and 2.16 are used to decorrelate the weight vectors within each sub-matrix, which we call the internal decorrelation process.

$$\mathbf{w}^+_{i(p+1)} = \mathbf{w}_{i(p+1)} - \sum_{j=1}^{p, p \leq n_i} \mathbf{w}^T_{i(p+1)} \mathbf{w}_{ij} \mathbf{w}_{ij} \tag{2.15}$$

$$\mathbf{w}_{i(p+1)} = \mathbf{w}^+_{i(p+1)} / ||\mathbf{w}^+_{i(p+1)}|| \tag{2.16}$$

Furthermore, the weight vectors generated from different sub-matrices in parallel need additional decorrelation, which is called the external decorrelation process. This process is carried out as follows,

1. Update each weight vector in different sub-matrices by,

$$\mathbf{w}^+_{(q+1)i} = \mathbf{w}_{(q+1)i} - \sum_{j=1}^{q, q \leq (n-n_i)} \mathbf{w}^T_{(q+1)i} \mathbf{w}_j \mathbf{w}_j \tag{2.17}$$

where $\mathbf{w}_{(q+1)i}$ is a weight vector of the sub-matrix $\mathbf{W}_i$ and $\mathbf{w}_j$ is a weight vector of other sub-matrices.

2. Normalize the weight vector,

$$\mathbf{w}_{(q+1)i} = \mathbf{w}^+_{(q+1)i} / ||\mathbf{w}^+_{(q+1)i}|| \tag{2.18}$$

With the internal decorrelation and the external decorrelation, we can decorrelate all weight vectors in all sub-matrices. See Fig. 2.5 for a structural illustration of parallel ICA. The decorrelation of two sub-matrices is proved as follows.

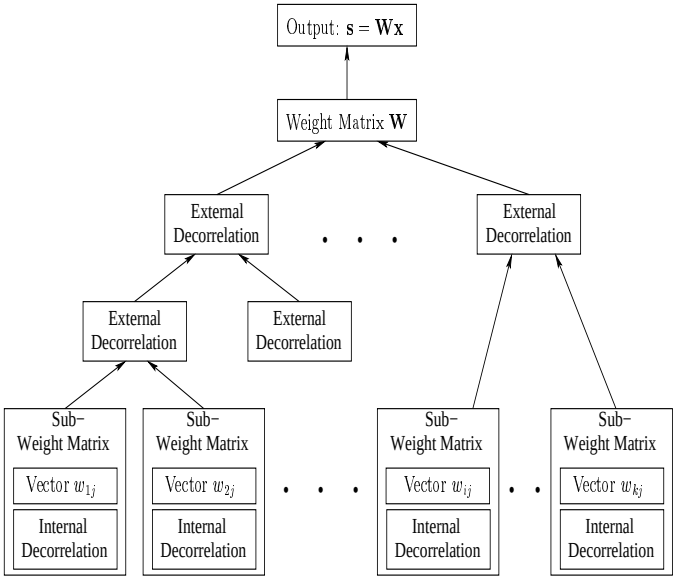


Figure 2.5: Structure of parallel ICA.

Let us first consider the external decorrelation. Given two sub-matrices $\mathbf{W}_i$ and $\mathbf{W}_j$, where $\mathbf{W}_i = (\mathbf{w}_{i1}, \cdots, \mathbf{w}_{in_i})^T$ and $\mathbf{W}_j = (\mathbf{w}_{j1}, \cdots, \mathbf{w}_{jn_j})^T$. Without loss of generality, we assume $\mathbf{W}_j$ is prior to $\mathbf{W}_i$. Then we have the already decorrelated

weight vectors in matrix $\mathbf{W}_i$,

$$\mathbf{w}_{iu}^{+} = \mathbf{w}_{iu} - \sum_{v=1}^{n_j} \mathbf{w}_{iu}^{T} \mathbf{w}_{jv} \mathbf{w}_{jv} \tag{2.19}$$

where $u = 1, \cdots, n_i$.

Without loss of generality, we assume the weight vector $\mathbf{w}_{i(p+1)}$ has the internal decorrelation defined as Eq. 2.15 and rewritten as,

$$\mathbf{w}_{i(p+1)}^{+} = \mathbf{w}_{i(p+1)} - \sum_{v=1}^{p} \mathbf{w}_{iu}{}^{T} \mathbf{w}_{iv} \mathbf{w}_{iv} \tag{2.20}$$

where $p = 1, \cdots, n_i$.

Substituting the external decorrelation (Eq. 2.19) into the internal decorrelation (Eq. 2.20), we get

$$\mathbf{w}_{i(p+1)} = \mathbf{w}_{i(p+1)} - \sum_{v=1}^{p} \mathbf{w}_{i(p+1)}^{T} \mathbf{w}_{iv} \mathbf{w}_{iv} - \sum_{r=1}^{n_j} \mathbf{w}_{i(p+1)}^{T} \mathbf{w}_{jr} \mathbf{w}_{jr} \tag{2.21}$$

where the second component comes from the internal decorrelation and the third component from the external decorrelation. That is,

$$\mathbf{w}_{i(p+1)} = \mathbf{w}_{i(p+1)} - \sum_{v=1}^{p+n_j} \mathbf{w}_{i(p+1)}^{T} \mathbf{w}_{v} \mathbf{w}_{v} \tag{2.22}$$

Comparing Eq. 2.22 with Eq. 2.11, we draw the conclusion that with the external decorrelation each weight vector in the two sub-matrices can be decorrelated as if it is computed within one matrix. For decorrelation of more than two sub-matrices, the proof is similar.

Generally speaking, in the parallel ICA algorithm, not only the sub-matrices can be estimated in parallel mode, the external decorrelation processes can also be carried out in a distributed fashion, as shown in Fig. 2.5.

2.4 Summary

In this chapter, we first reviewed the principles and properties of ICA algorithm, as well as the procedure of FastICA. ICA estimates source signals given only the observations which are linear mixtures of the source signals. Taking this advantage, we then present the ICA-based band selection, which chooses independent bands containing the most information of the original hyperspectral images. To speed up the ICA based band selection, we developed the parallel ICA algorithm which estimates and decorrelates independent components in a parallel mode. The results of the parallel ICA-based band selection serve as the input to the classifiers described in the next chapter.

Chapter 3

Classifiers Design

Many classification techniques have been applied to hyperspectral images. Most of these techniques belong to one of the two categories: the unsupervised classification and the supervised classification. In this chapter, we use three unsupervised classifiers: k-means, winner-take-all (WTA) and Kohonen maps, and one supervised classifier, k-Nearest-Neighbor (kNN), to classify pixels in hyperspectral images. In addition, we compare the pixel classification error rate to identify mixture pixels.

3.1 Unsupervised Classifiers

Unsupervised classification refers to the process that tries to find the intrinsic structure of unlabeled data by organizing it into groups or clusters [20]. In hyperspectral image analysis, the unsupervised classifiers can classify each pixel into different clusters, which denote different materials, without using any training set. The cluster centers can be used to represent the pixels within their corresponding clusters.

3.1.1 k-means Classifier

As an unsupervised classifier, k-means, also known as c-means, minimizes a performance metric which is defined as the sum of the squared distances between pixels and the corresponding cluster centers [20]. k stands for the number of cluster centers, $\mu_1, \cdots, \mu_k$.

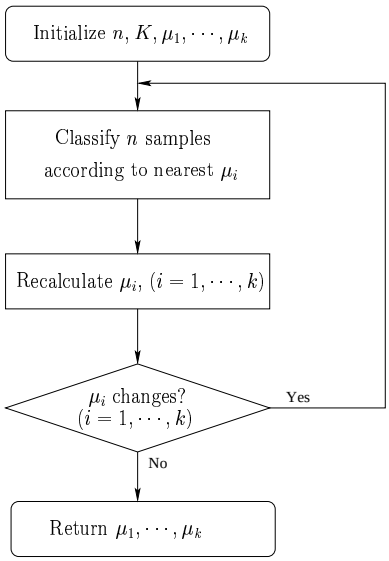


Figure 3.1: The diagram of k-means algorithm.

As shown in Fig. 3.1, k-means algorithm assumes there are k clusters, i.e., the hyperspectral image includes k kinds of materials. It first arbitrarily divide pixels into k clusters by randomly initializing cluster centers $\mu_1, \cdots, \mu_k$ and assigning pixels to the nearest cluster by measuring the Euclidean distance. The cluster centers can then be iteratively updated by the mean value of the cluster members, shown in Eq. 3.1.

$$\mu_j = \frac{1}{N_j} \sum_{i=1}^{N_j} x_i, \quad x_i \in S_j \tag{3.1}$$

where μ_j is the updated cluster center of the j^{th} cluster S_j, N_j is the number of cluster members in S_j which could vary at different iterations. With updated cluster centers, cluster members are accordingly reassigned to the nearest renewed cluster by Eq. 3.2,

$$x \in S_j, \quad \text{if } \|x - \mu_j\| \leq \|x - \mu_i\| \text{ for all } i = 1, \cdots, k \tag{3.2}$$

This loop lasts until no cluster member changes its membership, that is, all cluster centers remain the same. The cluster centers can then be used to represent all the members in each corresponding cluster.

3.1.2 Winner-take-all (WTA) Classifier

As a generalized approach to the k-means, the winner-take-all (WTA) algorithm, also called the competitive learning [52], begins again with k arbitrary cluster centers ω_i, where $i = 1, \cdots, k$. For each input pixel x, the closest cluster center ω_α is obtained the same way as in Eq. 3.2, with a change of notation, as shown in Eq. 3.3, where we use ω_α to denote the winning cluster center [20].

$$x \in S_\alpha, \quad \text{if } \|x - \omega_\alpha\| \leq \|x - \omega_i\| \text{ for all } i = 1, \cdots, k \tag{3.3}$$

The difference between WTA and k-means is that the winning cluster center is also justified by each pixel it claims membership of:

$$\omega_\alpha = \omega_\alpha + \varepsilon(x - \omega_\alpha) \tag{3.4}$$

where ε is a learning parameter and has a typical value in the range of $0.01 \sim 0.1$ [45]. The equation says that besides identifying a winner, each pixel also "pulls" the winner towards itself a little bit which can be interpreted as a privilege of the pixel over its winner. The pulling process of each pixel on its winner is illustrated in Fig. 3.2.

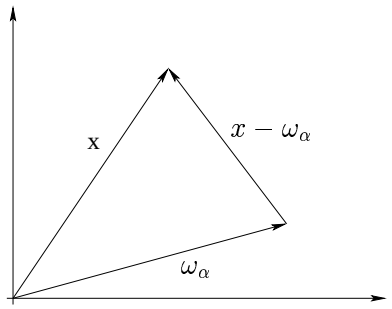


Figure 3.2: Winner-take-all: update of winner.

Similarly, the algorithm converges when all the cluster centers remain the same.

3.1.3 Kohonen Maps Classifier

Compared to k-means and WTA, Kohonen maps, also called self-organizing feature maps, is an unsupervised neural network which represents pixels with one layer of neurons and assumes a problem-dependent topological distance between the neurons [45].

Following the principle of k-means and WTA, Kohonen maps also defines the neurons (cluster centers) with weight vectors and randomly initializes them. For each input pixel, the closest neuron, called the Best Matching Unit (BMU) [39], is activated and adjusted to closely resembles the input pixel. The difference is that besides the BMU, the surrounding neurons of BMU are simultaneously adjusted according to their topological distance to the BMU, which is illustrated in Fig. 3.3.

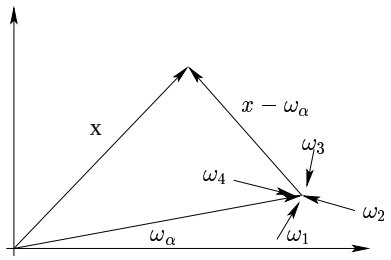


Figure 3.3: Kohonen feature maps: updating BMU and its surrounding neurons.

As shown in Fig. 3.3, the BMU ω_α and its surrounding neurons, ω_1, ω_2, ω_3, ω_4, are modified by Eq. 3.5.

$$\omega_i = \omega_i + \varepsilon\Phi(x - \omega_i) \tag{3.5}$$

where $i = 1, \cdots, \alpha, \cdots, N$ and N is the number of neurons.

ε is the learning rate and given as below.

$$\varepsilon = \varepsilon_{max}\left(\frac{\varepsilon_{min}}{\varepsilon_{max}}\right)^{\frac{n}{n_{max}}} \tag{3.6}$$

where ε_{min} and ε_{max} are the minimum and maximum learning rate, $0 \leq \varepsilon_{min} \leq \varepsilon_{max} \leq 1$, n_{max} is the maximum iteration number, and n is the current iteration number. ε_{min}, ε_{max} and n_{max} are set at the beginning of the process. As n increases, ε decreases which makes the learning process slower and the neural network more stable. To simplify the computing complexity, ε is typically set to 0.01.

Φ in Eq. 3.5 is defined as

$$\Phi = \exp\left(-\frac{\|\omega_\alpha - \omega_i\|^2}{2\sigma^2}\right) \tag{3.7}$$

which is a Gaussian function with respect to the distance between the neuron and the

BMU. The closer the neuron to the BMU, the more it is affected. σ is the variance defining the neighborhood factor,

$$\sigma = \sigma_{max} \left(\frac{\sigma_{min}}{\sigma_{max}} \right)^{\frac{n}{n_{max}}} \tag{3.8}$$

where σ_{min} is set at initialization and $\sigma_{max} = \frac{\sqrt{N}}{2}$.

Similar to WTA, Kohonen maps iteratively employs input pixels until all neurons remain the same or $n > n_{max}$. The neurons then represent pixels surrounding them.

An important benefit of Kohonen maps algorithm is that it can not only classify the pixels, it is also able to cluster the neurons resembling the pixels. For example, if the assumed number of neurons is larger than the actual number of material types, some neurons may represent the most typical ones while other surrounding neurons may only represent the less typical pixels. Thus, these less representative neurons can be clustered with their adjacent primary neurons and be neglected in the following iterations.

3.2 Supervised Classifiers

By possessing a training set of labeled samples, we can employ the supervised classifiers. Depending on whether we know the form of the sample density function, the supervised classifiers can be divided into two categories: parametric and non-parametric [20]. For example, assume the density function of the data is known and has the shape of Gaussian function, the parametric classifiers include the maximum *a-posteriori* (MAP) classifier and the linear discriminant functions. On the other hand, without assuming the density function to be known *a-priori*, the non-parametric classifiers include the k-nearest-neighbor classifier and the neural network. In this work, we

implement the k-Nearest-Neighbor classifier in hyperspectral image analysis.

3.2.1 k-Nearest-Neighbor Classifier

k-Nearest-Neighbor (kNN) classifier is a non-parametric supervised classification technique [22] in that there is no assumption that the forms of the densities are known. kNN inputs a pixel x to the training set and expands a region centered at the input pixel until this region contains k training samples, as demonstrated in Fig. 3.4.

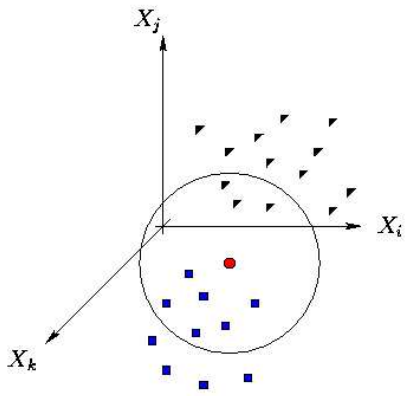


Figure 3.4: kNN classification. The circle indicates the region centering at the input pixel and containing k training samples. The circle represents the input pixel. Squares represent the training samples in class 1. Triangles represent the training samples in class 2.

In general, k is a function of the number of samples in the training set, n, and typically set to $k = \sqrt{n}$ [45]. The category of the input pixel is then estimated by comparing the number of training samples from different categories within the k nearest neighbors. Suppose k_R of the k nearest neighbors belong to category R, then the

probability of x belonging to category R is

$$p(\omega_R \mid x) = \frac{k_R}{k} \tag{3.9}$$

If $p(\omega_R \mid x)$ is larger than the probabilities of all other categories, kNN assigns the input pixel to category R.

Besides classifying input pixels to the known categories, kNN can also help to simultaneously identify mixture pixels in hyperspectral images for further processing.

3.2.2 Mixture Pixels Identification

As described in Chapter 1, we cannot achieve a high classification accuracy without proper consideration of pixel pureness. In another word, there always exists a distance between an input pixel and the training sample nearest to it. We call this distance, caused by noise or pixel unpureness, as pixel classification error. The noise in hyperspectral image comes mainly from the air, environment, or equipment condition, and has a less effect on the classification results than on the pixel unpureness. On the other hand, the pixel unpureness comes from mixing of different materials in one pixel, which can sometimes dramatically affect the classification result. Therefore, given the condition that we do not know the distribution of the noise existing in a hyperspectral image, we can find the mixture pixels by comparing the classification error rates for all pixels in this hyperspectral image, where the classification error rate is given in Eq. 3.10.

$$e = \frac{d - D_{min}}{D_{max} - D_{min}} \tag{3.10}$$

where $d = |x - c|$ is the distance between the input pixel x and the training sample c nearest to it. $D_{max} = \max_{\forall x} d$, and $D_{min} = \min_{\forall x} d$.

Then, we assume an acceptable error rate as threshold, and compare it with the classification error rate of each pixel. If the error rate of a pixel is less than the acceptable error rate, we consider it as a pure pixel. However, if the error rate of a pixel is greater than the acceptable error rate, it will be marked as a mixture pixel for further processing.

3.3 Summary

This chapter reviews the procedures of k-means, winner-take-all, Kohonen maps and kNN classifiers. k-means, winner-take-all and Kohonen maps are unsupervised classifiers which cluster pixels and represent them with cluster centers. kNN is a non-parameter supervised classification technique based on its nearest neighbors. Applying kNN and obtaining classification results, we evaluate the pixel classification error rates corresponding to pure pixels, and then identify mixture pixels in hyperspectral images. The identification results will be evaluated in Chapter 5.

Chapter 4

Synthesis of Parallel ICA on FPGA

Integrated Circuit (IC) is a microelectronic semiconductor device consisting of many interconnected transistors and other components, such as inverters, AND gates, OR gates, etc. IC is fabricated on a die, which is a small rectangle cut from a silicon wafer. From the Small Scale Integration (SSI) containing tens of transistors in early years, IC technology has developed through Medium Scale Integration (MSI), which includes hundreds of transistors, to Large Scale Integration (LSI) with capacity of thousands of transistors. Today, Very Large-Scale Integrated Circuit (VLSI) allows user to implement large complex design on millions of transistors [21].

To synthesize a design, microelectronic system designers first specify the architectural requirements using Hardware Description Language (HDL). Then, the synthesis is verified through a series of simulations considering both accuracy and efficiency. Finally, the design is implemented on Application-Specific Integrated Circuits (ASICs).

Field Programmable Gate Arrays (FPGAs), a member in the ASIC family, is the best selection for fast design implementation. It features low prototyping costs, re-

configurability and less production time. However, for some complex designs, single FPGA is usually insufficient in providing the required capacity and speed. A solution to this is High Performance Reconfigurable Computing (HPRC), a growing research field which provides performance gain by connecting FPGA-based computing nodes together [44].

In this chapter, the members in ASIC family are first introduced and compared. Furthermore, the features of FPGA and the architecture of HPRC are presented in detail. We then briefly review the (Very High Speed Integrated Circuit) Hardware Description Language (VHDL), the re-configurable components in VHDL and the synthesis procedure. Finally, we focus on the synthesis of parallel ICA on FPGA using re-configurable components.

4.1 FPGA and HPRC

4.1.1 ASICs

ASICs can be divided into two categories: semi-custom group and full-custom group. In full-custom group, designers develop their ICs using analog and digital mixed technologies where the silicon can be used in the most efficient manner. Designers always replace up to 10,000,000 gates or more for different implementations. However, the high cost of the workstation-based development system ($150,000) and the slow turnaround time (usually 8 weeks) make it unsuitable for small and moderate size designs and fast implementations.

The semi-custom group includes user programmable ICs and non-programmable ICs. The non-programmable ICs consist of Mask Gate Arrays (MGAs) and Standard

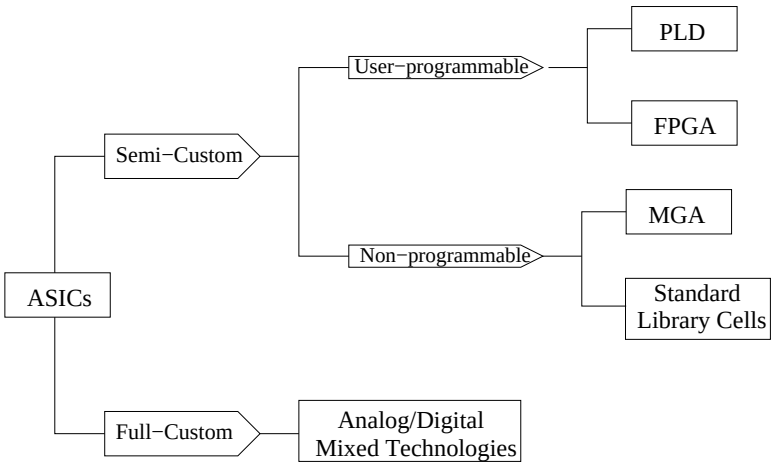


Figure 4.1: ASIC family.

Library Cells. With non-programmable devices, designers can implement their designs by specifying interconnections on ICs once for all.

MGA, containing 10,000,000 gates or more, is the best for moderate size designs. Before implementation, MGA vendors prefabricate rows of gates on wafers, where designers can specify and interconnect two layers to implement logic functions. To achieve higher performance, Random Access Memory (RAM) may be embedded inside MGAs. Normally, a workstation-based development system of MGA would cost $\$50{,}000$ and it takes $3\sim5$ weeks to implement a prototype.

Compared to MGAs, Standard Library Cells are the best selections for large amount of production applications with significant functions such as multipliers and CPU. For Standard Library Cells, vendors develop library disk files of significant functions, while users select cells, which is a logic level component with constant height on chip, according to their designs and specify two layers of interconnections to fulfill the whole scheme. Even though Standard Library Cells make the implementation of complicated designs possible, there are two disadvantages hindering their widely usage: the cost of

the workstation-based development system goes as high as $100,000 and the turnaround time for prototypes is around 8 weeks.

The user programmable ICs include Programmable Logic Devices (PLDs) and FPGAs, which are developed based on re-configurable technology [43]. Using programmable devices, designers can implement and modify their designs multiple times.

PLD, which includes 300 to 8,000 gates within a single package of 20$\sim$ 84 pins, is the best choice for simple design with many inputs and outputs. PLD vendors fabricate multiple sets of gates with programmable connections, with which users can specify to implement logic functions. Although the capacity of PLD limits the complexity of designs that are targeted to it, the lowest development expense and the shortest turnaround time make PLD very flexible to handle simple designs. A PC-based PLD development system costs only $3,000$\sim$$5,000, and designers can electrically program and erase their designs once within minutes.

4.1.2 Features of FPGA

FPGA, which is larger and more complicated than PLD, but cheaper and faster in implementation than MGA, is the best choice for small amount of production applications. FPGA covers high-end PLD applications and low-end MGA applications at the same time [7]. Comparing to PLD, FPGA is suitable for more complicated designs since FPGA contains more gates and uses architectures which support a balance between logic resources and routing resources [9]. On the other hand, compared to MGA, FPGA reduces the time-to-market and results in profitability increasing, since designers can program the interconnections in a few hours instead of waiting several weeks for the final metalization of the MGAs.

FPGAs consist of a two-dimensional array of logic blocks. These logic blocks, also called configurable logic blocks (CLBs) [49], are programmable and can be used to implement logic functions. FPGA vendors prefabricate rows of gates and programmable connections, and designers specify the programmable connections, which are programmed to connect the CLBs, to implement complex logic functions.

From the internal structure point of view, currently, there are four main categories of FPGAs: symmetrical array, row-based, hierarchical PLD, and sea of gates [18], as shown in Fig. 4.2.

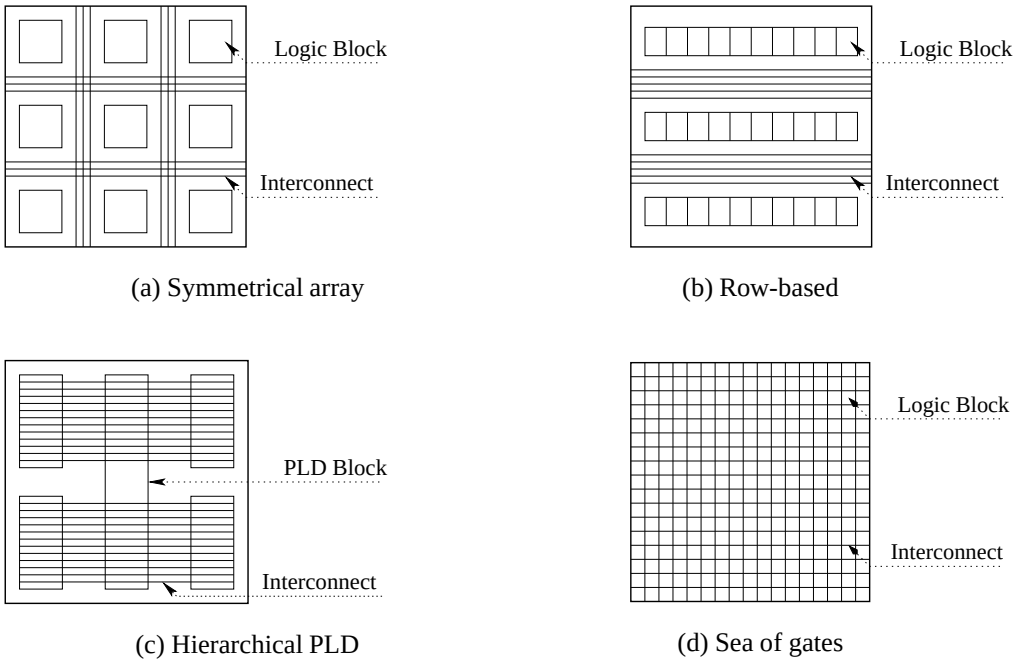
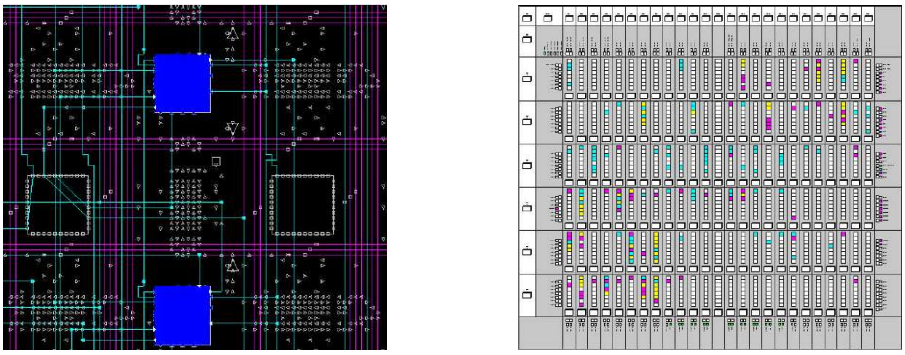


Figure 4.2: Four types of FPGAs [18].

In current commercial FPGAs, products of Altera employ the hierarchical-PLD structure, Xilinx utilizes the symmetrical array structure, while Actel FPGAs use row-based structure. Figure 4.3 shows the physical layouts of Xilinx Virtex and Altera

FLEX-10K.



(a) Xilinx (b) Altera FLEX-10K

Figure 4.3: Layout of FPGAs [7]. (a) is a Xilinx FPGA using the symmetrical array structure. (b) is an Altera FPGA using the hierarchical-PLD structure.

As re-programmable ICs, FPGAs feature rapid prototyping, low testing cost and risk, standard product advantages, and reusable life cycle. Moreover, FPGAs have advantages on development flexibility and expense. A PC-based development system costs only \$5,000$\sim$\$10,000.

On the other hand, FPGAs also have disadvantages of lower speed of circuits and lower gate density. From the view of ICs instead of development systems, an FPGA, which typically contains 2,000 to 2,000,000 gates [8], may be 2 to 10 times slower on processing and 2 to 10 times more expensive than an MGA with equivalent number of gates.

Additionally, capacities of most single FPGAs are not adequate to the application requirements of some complicated designs, where system capacity indicates the physical resource available on a single FPGA while application requirement indicates the resource needed for specific design. HPRC provides solutions for such capacity prob-

lems by connecting FPGA-based computing nodes in a network [44].

4.1.3 HPRC

Based on FPGA, HPRC exploits the parallel processing benefits of High Performance Computing (HPC), which seeks extreme computing power with supercomputers and parallel computing, in conjunction with adaptive hardware acceleration associated with Reconfigurable Computing (RC) [11], which employs the re-programmable feature of FPGAs by modifying hardware at runtime [2]. As shown in Fig. 4.4, the HPRC platform consists of a number of computing nodes and RC boards, which are reconfigurable computing elements associated with the computing nodes. For the purpose of data exchanging and synchronization, all computing nodes are connected by the interconnection network (ICN) or reconfigurable ICN.

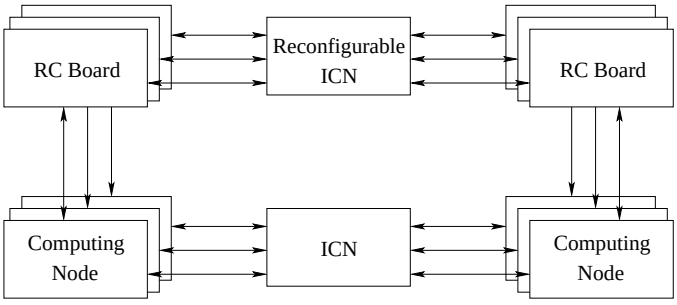


Figure 4.4: Architecture of HPRC.

4.2 Synthesis Procedure

To implement a design on FPGA or HPRC, first of all, the designers should have a thorough understanding of the application requirements, followed by the interpretation

of desired functionality, timing and architectural specifications with HDL. In the end, the function and structure representations are synthesized on the structural level for a specified FPGA.

4.2.1 VHDL

As one of the HDLs, VHDL was originally developed by the US Department of Defense in 1981 for the purpose of maintenance and redesign. Its standard was approved by IEEE in 1987 [3]. Compared to other software languages, VHDL is designed to describe digital electronic systems.

VHDL possesses many features to fulfill the needs in design procedure. First of all, VHDL allows the description of the structure of a system. Thus, designers can decompose the design of a system into multiple subsystems and interconnect them. In each detailed subsystem design, VHDL allows concurrent processes, such that the designers can module their function components in parallel mode, visualize the designed processes and manipulate them efficiently. Additionally, VHDL allows a complicated design to be synthesized from a high-level specification, such as reconfigurable components. As a result, designers can concentrate more on strategic design instead of low-level implementation. Furthermore, by implementing the design of a system using VHDL, the designers can compare different design options and validate the design with a series of simulations in a short time.

When we represent a design using VHDL, we first develop an initial architectural specifications, followed by decomposing the design into subsystems, each of which consists of several processes. Consequently, we draw a block diagram of the top level, develop the structural VHDL for all levels to show the interconnections of subsystems,

and then develop the VHDL codes for each subsystem. Accordingly, each subsystem and the top level block are compiled and tested individually for functionality validation. Finally, the tested subsystems are integrated and the whole system is tested as well.

4.2.2 Re-configurable Components

In the procedure of representing a system with VHDL, re-configurable components (RC) make system design and synthesis faster, easier and more efficient. The best benefit of using the RCs is the reduced time-to-market which directly leads to higher revenues. It has been shown that, compared to the effort of one-time use, it takes about 50% more to prepare code for reuse and re-configuration [7]. Although reuse still requires a little amount of time to integrate and verify the RCs in the specific environment, the designers would benefit for 70% or more design time reduction from the RCs developed by predecessors [7]. As another advantage, the well-tested RCs not only shrink the design time, they also reduce the risk of mistakes by avoiding the development and verification of available RCs, and thus sparing more effort on system structure and other specific processes.

When a system is designed using the RCs, designers should thoroughly understand the system architecture and the design requirements as well. If a RC is needed and is available in the library, designers can fetch and configure it, as demonstrated in Fig. 4.5.

To integrate and interconnect all necessary components, one top level block, as shown in Fig. 4.6, is designed, which can arrange the RCs in distributed or parallel modes on multiple layers.

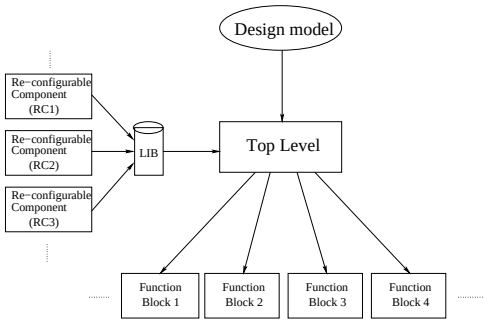


Figure 4.5: Design flow using re-configurable components.

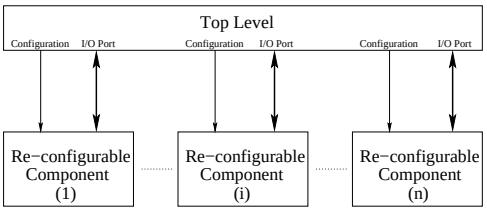


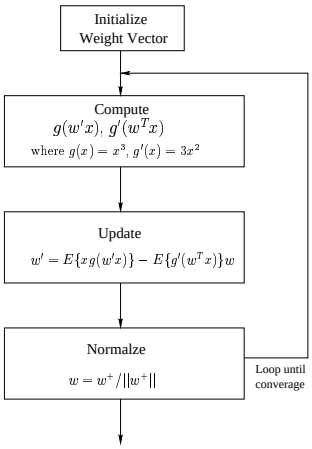
Figure 4.6: Structure of the re-configurable components specification.

4.2.3 Re-configurable Components in Parallel ICA Synthesis

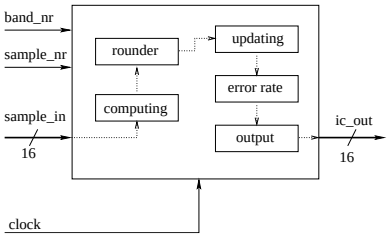
In the parallel ICA synthesis, we develop re-configurable components of the one unit process, decorrelation process and comparison process for the ICA based band selection, as shown in Fig. 4.7.

The one unit component consists of 5 processes: computing, rounder, updating, error rate computing and output. Compared to the other 4 basic functional processes, the rounder process is necessary for avoiding overflow, since the 16-bit vector instead of floating point number is used as data format in the computing.

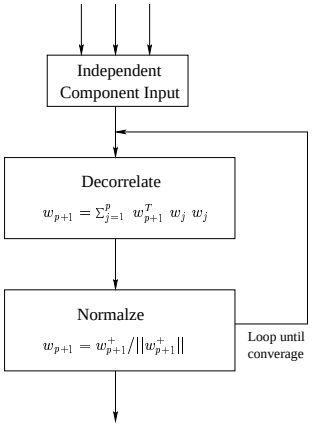
The decorrelation component contains processes of decorrelating, updating, error rate computing and output. For different number of input weight vectors, the designer needs to modify the number of input ports for the decorrelation component. The comparison component compares and sorts all decorrelated weight vectors, and selects the



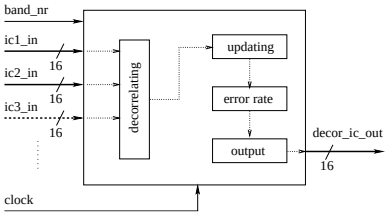
(a) One unit process



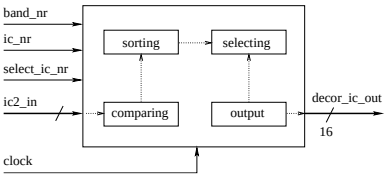
(b) One unit component



(c) Decorrelation process



(d) Decorrelation component



(e) Comparison component

Figure 4.7: The re-configurable components in synthesis.

specified number of the most independent bands.

4.2.4 Synthesis on FPGA

Synthesis focuses on high level design and relies on synthesis software and programmable logic to produce prototypes in short time. Therefore, the prototypes can be easily modified if the requirement changes.

The synthesis procedure begins from synthesizing the structural description, which consists of a set of function blocks and interconnections, using VHDL and re-configurable components from library. The synthesis procedure includes:

- Translating VHDL into Boolean mathematical representations.
- Optimizing the representations based on criteria such as size, delay and testability.
- Mapping the optimized mathematical representations to a technology-specific library of components.

From the simulation point of view, the synthesis procedure can be summarized into four steps as illustrated in Fig. 4.8: the prelayout simulation, the gate level simulation, the placement and routing, and the postlayout simulation.

The prelayout simulation, also called behavioral simulation, is the first level test and verification in the synthesis procedure. At this stage, any error in the design can be corrected easily and fast. The prelayout simulation focuses on the functionality test and ignores the timing and unit delay simulation related to the target technology by setting the delay to a fixed value. In another word, the prelayout simulation is a technology independent simulation.

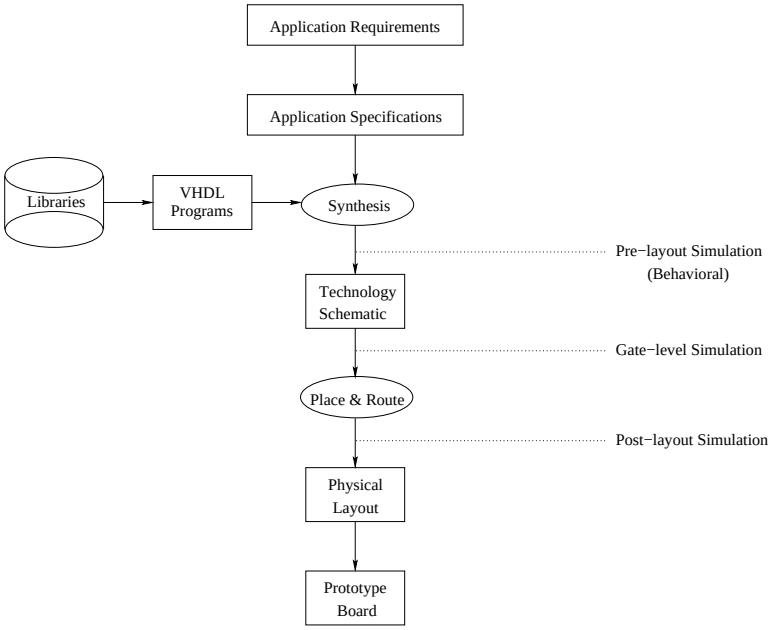


Figure 4.8: Synthesis on FPGA.

Following the functionality test in the prelayout simulation, the gate level simulation synthesizes the high-level design descriptions written in VHDL into optimized gate-level design. In this step, synthesis software, such as FPGA Compiler II (FC2) provided by Synopsys, uses the target library to optimize the design. As a result, the frequency and slices utilization of the design are achieved and evaluated. Additionally, in the gate-level simulation, synthesis software can use user-specified standard cell and gate array libraries to translate the design from one technology to another.

After the gate level simulation, the design has been mapped with the target library and the technology has been specified. The placement and routing process can then synthesize function blocks on FPGA through programming interface and test bench. The process generates the physical layout of the design as well as the standard delay format such as the logic-cell delays and the interconnection delays.

As the final test and verification step, the postlayout simulation simulates the function blocks on FPGA. Compared to prelayout simulation, postlayout simulation is technology-dependent and relies on the target library. The delays corresponding to different technology would vary. Different from the gate level simulation, the postlayout simulation estimates the signal outputs delay of the design. If the postlayout simulation result is satisfied, the whole synthesis procedure is completed and the system is ready to be downloaded to a prototype board for demonstration.

Generally speaking, synthesis reduces the design time required to achieve and verify a given functionality, since multiple candidate solutions can be constructed quickly and accurately. Moreover, prototyping with FPGAs also speeds up verification and reduces design risk.

4.3 Synthesis of Parallel ICA

Taking the advantages of synthesis on FPGA, the parallel ICA process containing 4 independent components is synthesized according to the architectural specification demonstrated in Fig. 4.9.

Inside the system, a top level block is designed to configure and interconnect the 4 one unit components, 3 decorrelation components and 1 comparison component. In the mean while, the top level block also transfers input data, controls internal signals among different components and outputs final results.

As the first step, each hyperspectral pixel is extracted from the hyperspectral image by representing its reflectance percentage on each band with a 16-bit binary. The hyperspectral pixels are then input to the parallel ICA system and distributed by the top

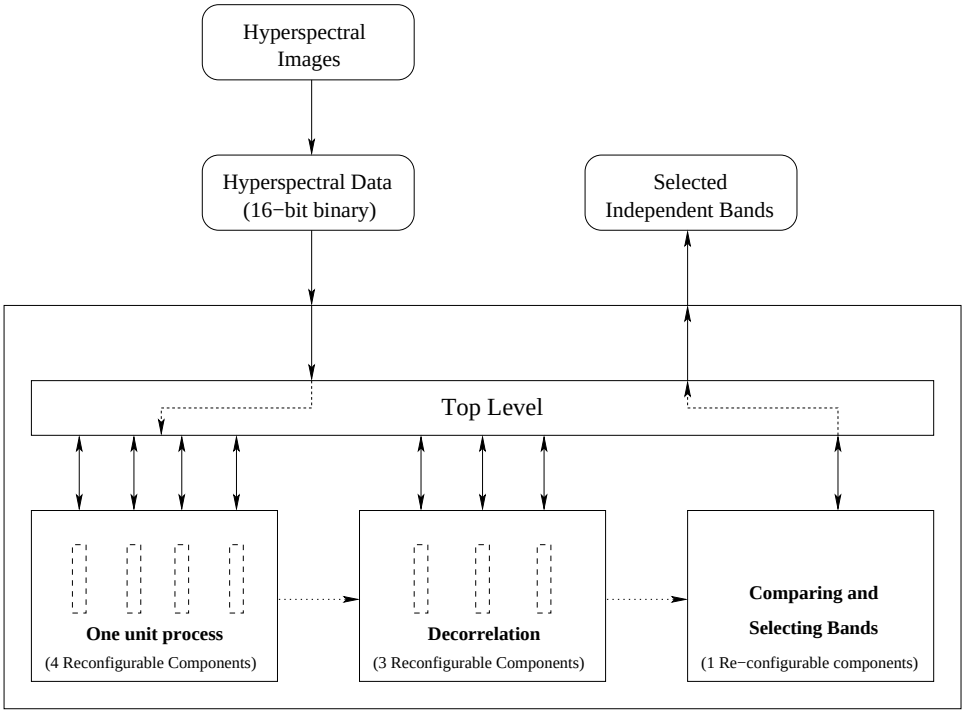


Figure 4.9: Architectural specification of parallel ICA. (Solid lines denote data exchange and configuration. Dotted lines indicate the virtual processing flow.)

level block to the 4 one unit re-configurable components. Each one unit component processes samples and feeds a weight vector back to the top level block, which in turn sends the weight vectors to corresponding decorrelation components. Consequently, the decorrelation components decorrelate weight vectors and generate the final weight matrix. Finally, the comparison component receives the weight matrix through the top level block, compares and selects the corresponding independent bands. The selected bands are the outputs from the top level block.

4.4 Summary

In this chapter, we compared the advantages and disadvantages of members in the ASIC family. We reviewed the features and architectures of FPGA and HPRC. FPGA is the best selection for moderate designs and fast implementations, while HPRC interconnects FPGA-based computing nodes and takes the re-programmable feature of FPGA. In addition, we presented the synthesis procedure of the parallel ICA based band selection on an FPGA, Xilinx V1000E. In the synthesis, we developed three re-configurable components, which can be reused and reconfigured for other designs. Simulations and layouts will be presented in Chapter 5.

Chapter 5

Experiments and Performance Evaluation

In this chapter, we present the experimental results for the algorithms and methods proposed in previous chapters. As demonstrated in Fig. 5.1, this chapter first describes the multispectral and hyperspectral image data sets used in the experiments. The parallel ICA is then applied to these data sets to select the most independent spectral bands. Both unsupervised and supervised classifiers are applied to the original and the band reduced spectral images. The classification results are evaluated from several aspects. Mixture pixels are identified by comparing pixel classification error rate. Finally, the parallel ICA based band selection is synthesized on FPGA, and the relationship between the number of independent components and capacity utilization of FPGA is analyzed.

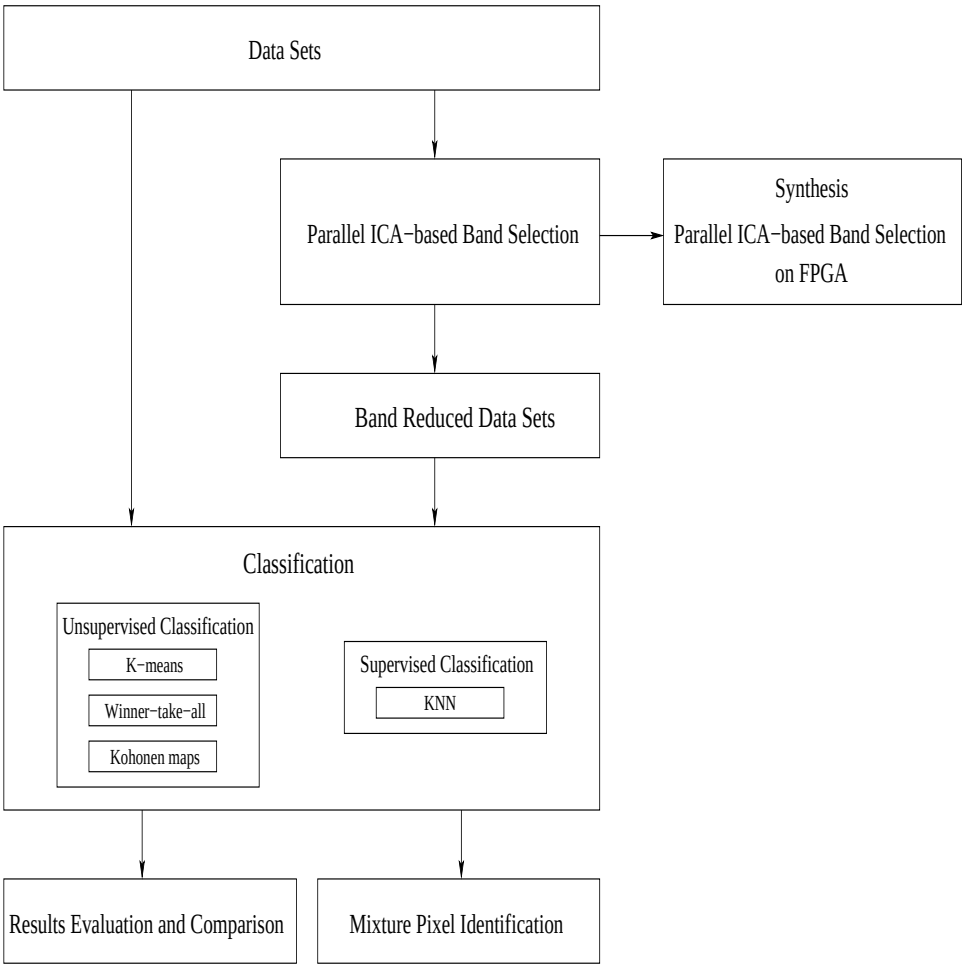


Figure 5.1: Experimental flow.

5.1 Experimental Data Sets

The data sets used in the designed experiments are obtained from three sources: multispectral images taken at North Carolina State University, hyperspectral images taken by NASA AVIRIS and synthetic hyperspectral images generated based on the JPL and USGS spectral library. These data sets are used to evaluate the band selection algorithm we proposed.

5.1.1 Multispectral Image Set

The multispectral image data set is provided by researchers of the Image Analysis Laboratory at NCSU [46]. They use a custom-made Pulnix digital color camera and an infrared camera sensitive in the 3$\sim$5μm range combined with filters obtained from ThermoOriel, as shown in Fig. 5.2. The system provides us multispectral images

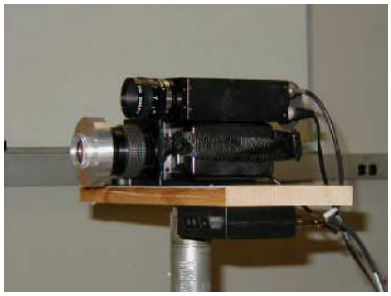


Figure 5.2: IR cameras with filters setting up on them [46].

with 6 bands: red, green, blue, and three infrared bands of 3$\sim$3.2μm, 3.2$\sim$4.2μm and 4.2$\sim$5μm [46].

The red band of one multispectral image used in our experiments is shown in Fig. 5.3. The scene is composed of three targets: a hot object, a toy truck and a cold object [47]. The hot object (the leftmost target) is a steel vessel containing warm water,

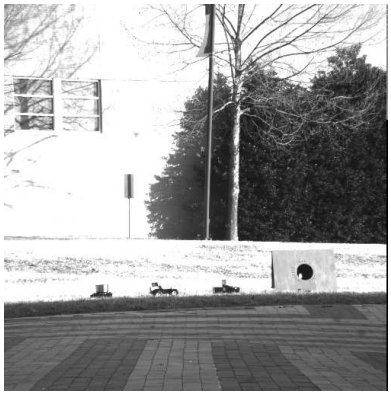


Figure 5.3: The red band of one multispectral image used in experiments.

the toy truck (middle target) has rubber tires and a body made of plastic and metal, while the cold object (right to the truck) is a steel vessel containing slush of ice and water. Along with these targets on grass, a rectangular plastic box with a circle cut-out in its center is put on rightmost. A sidewalk is in front on these targets, while a traffic sign and some trees are behind them. A building is located behind the traffic sign and those trees. Additionally, the background of this scene includes sky and a window on the building. In this scene, we define 14 categories of materials which are listed in Table 5.1.

Table 5.1: Categories in training set.

Category	1	2	3	4	5
Material	box	grass	brick	tree	sky
Note	rightmost rectangular		wall & sidewalk		
Category	6	7	8	9	10
Material	window	metal	target mixtures	target1 top	target2 bottom
Note	glass & frame	traffi c sign	three targets	leftmost, hot object	leftmost, hot object
Category	11	12	13	14	
Material	target2 top	target2 bottom	target3 top	target3 bottom	
Note	middle, truck cargo	middle, truck tires	right to the truck, cold object	right to the truck, cold object	

The multispectral image set used in our experiments consists of 41 multispectral

images in a multispectral close-in sequence, which is collected by taking a sequence of multispectral images as the camera is moving closer to the targets [47]. The main objective of this work is to emulate a "smart" missile, equipped with a multispectral camera, closing on a target. The multispectral image set is taken to emulate several seconds of flight.

5.1.2 AVIRIS Hyperspectral Images

The hyperspectral image set is obtained from NASA AVIRIS. Each hyperspectral image in the AVIRIS data set contains 224 spectral bands ranging from 369.85 nm to 2506.81 nm with a 9.8 nm interval between adjacent bands. Figure 5.4 shows a hyperspectral

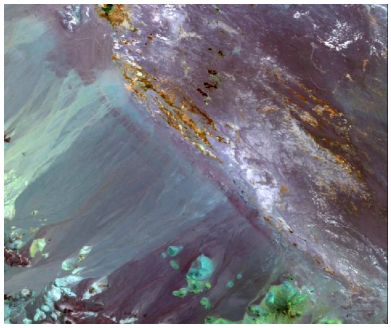


Figure 5.4: A hyperspectral image taken from AVIRIS (RGB).

image taken over Lunar Lake in NV on October 14, 1997 by AVIRIS operated by NASA JPL. AVIRIS collects data at a rate of 2 km^2/sec with a resolution of 17 m^2. Each obtained hyperspectral image has the spatial size of 614×512 in pixel and covers a 10.5 km^2 area. AVIRIS saves image data using interleave-by-pixel with dimensions of (224, 614, 512) [channel, sample, line]. Each reflectance is a binary IEEE 16-bit signed integer multiplied by 10000.

5.1.3 Synthetic Hyperspectral Image

The third set of testing images is synthetic, generated from the JPL and USGS spectral library [28, 57] for the availability of the ground truth. The JPL ASTER spectral library includes about 2000 spectra of natural and man made materials, and the USGS spectral library contains about 400 spectra of minerals and a few plants.

Since the ground truth of the AVIRIS hyperspectral image is unavailable, it is not appropriate in algorithm evaluation. Therefore, we develop a spectral library whose wavelength range, number of bands and interval are the same as those of the AVIRIS images. As listed in Table 5.2, this spectral library includes 418 samples of 6 categories of materials: man-made, rocks, soils, vegetation, water and minerals.

Table 5.2: Categories in spectral library.

Category	man-made	rocks	soils	vegetation	water	minerals
Number of Samples	45	296	41	4	6	26

Using samples in this spectral library, we then generate a 224-band synthetic hyperspectral image whose spatial size is 90×90 (row$\times$column) in pixel. As shown in Fig. 5.5, this synthetic hyperspectral image consists of 9 blocks, with each block con-

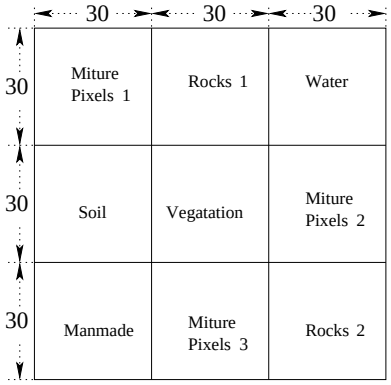


Figure 5.5: Composition of synthetic hyperspectral image.

taining either pure or mixture materials. Each pure material block includes 4 kinds of samples from the corresponding category, while each mixture block is composed of multiple materials according to Eqs. 5.1 $\sim$ 5.3.

$$Mixture\_1(SWM) = soil \times 30\% + water \times 30\% + manmade \times 40\% \quad (5.1)$$

$$Mixture\_2(RVM) = rocks \times 30\% + vegatation \times 30\% + manmade \times 40\% \quad (5.2)$$

$$Mixture\_3(SRV) = soil \times 30\% + rocks \times 30\% + vegatation \times 40\% \quad (5.3)$$

Figure 5.6 shows a label image of the corresponding synthetic hyperspectral image, with the color red, black, yellow, green, blue and white pixels representing man-made, rocks, soils, vegetation, water materials and mixture pixels, respectively.

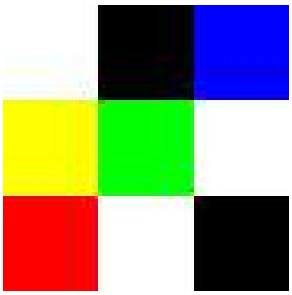


Figure 5.6: A label image of the corresponding synthetic hyperspectral image.

In order to simulate the effect of noise, we also add different noises into the original synthetic hyperspectral image.

5.2 Parallel ICA based Band Selection

The first experiment we conduct is to apply the parallel ICA developed in Chapter 2 to all the three data sets in order to select the most independent bands.

5.2.1 Independent Bands

We first apply the parallel ICA to the multispectral images from NCSU. For the multispectral image shown in Fig. 5.3, we assume 4 independent bands, instead of the original 6 bands, are enough to represent the information. The sorted weight according to Eq. 2.14 shows descending order of red, IR3 ($4.2 \sim 5\mu$m), blue, IR2 ($3.2 \sim 4.2\mu$m), IR1 ($3 \sim 3.2\mu$m) and green. In another word, ICA considers the band of red, blue and the IR bands of $4.2 \sim 5\mu$m and $3.2 \sim 4.2\mu$m containing the most information of all materials.

We then apply the parallel ICA to the AVIRIS data set shown in Fig. 5.4. Table 5.3 lists the 150 most independent bands out of the total 224 bands. In order to illustrate

Table 5.3: Bands for 150-channel weight matrix.

Band	44	116	141	151	109	46	111	101	91	172
	134	155	124	103	122	216	183	21	128	150
	80	14	224	138	30	206	99	192	73	184
	175	17	176	156	79	39	181	43	67	180
	56	204	194	11	159	40	94	48	23	117
	18	70	153	5	24	74	108	222	200	104
	178	135	10	66	185	186	87	92	118	201
	27	211	50	41	142	86	190	220	62	88
	161	106	71	171	28	152	78	85	83	25
	36	34	133	208	154	68	213	89	61	75
	52	203	82	158	174	130	45	136	95	16
	215	207	69	214	125	219	166	221	90	55
	107	93	123	218	58	49	191	115	42	105
	35	148	47	12	22	13	145	20	98	96
	38	26	162	157	187	217	110	3	29	139

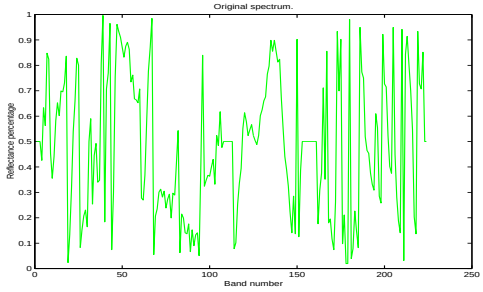
the importance of the independent bands within the spectrum, we randomly pick a pixel

from Fig. 5.4 and plot its spectral profile in Fig. 5.7. We highlight the first 150, 100, 50, 30 and 10 most independent bands and mark them on the profile respectively. As displayed in Fig. 5.7, the independent bands on the spectral profile contain the most important information to characterize the curve, including the maximum, minimum and inflection points. In another word, we can represent the original spectral profile with these independent bands, as plotted in Fig. 5.8. Generally speaking, spectral profiles represented by the selected independent bands reserve most characteristics of the curve. The more spectral bands selected, the closer it is to the original profile, but the less redundancy has been reduced. According to the spectra discrepancies between the objective materials and their surroundings, we need to assume enough numbers of independent bands considering specific cases.

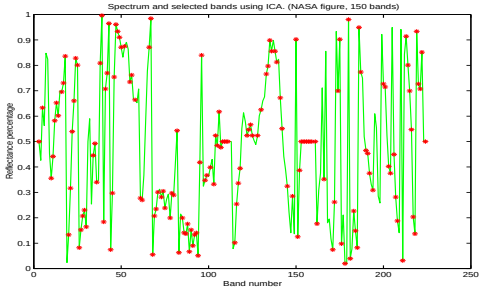
We repeat the same experiment on the synthetic hyperspectral image without noise, with uniform noise $(0 \sim 0.1)$, and different degrees of Gaussian noises. The results are shown in Figs. 5.9$\sim$ 5.12. Obviously, for this specific pixel, 50 independent bands instead of the original 224 bands are enough to well represent its spectral signature. The spectra and the order of the independent bands vary a little for the reason of different noises.

5.2.2 Computation Time of Parallel ICA

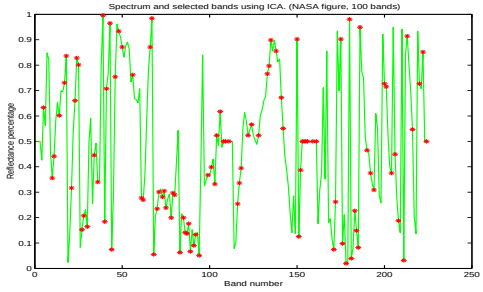
To evaluate the performance of parallel ICA, we manually distribute the weight vector estimation process to 5 computers and obtain 5 sub-matrices from internal decorrelation. We then execute the external decorrelation process for every 2 sub-matrices and decorrelate all sub-matrices with 4 external decorrelations. The computation time of the parallel ICA is estimated by adding the computing time of the slowest sub-matrix



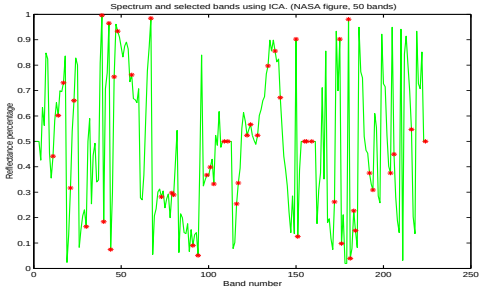
(a) 224-channel spectrum curve



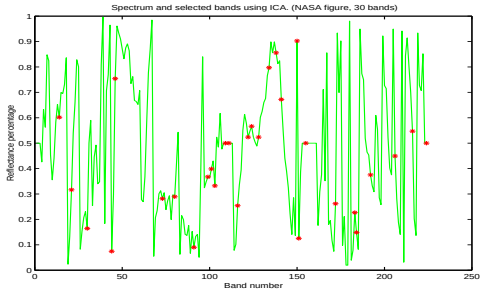
(b) 150 independent bands.



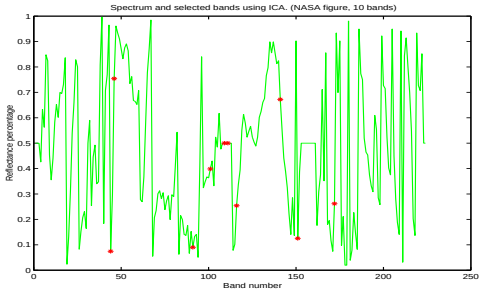
(c) 100 independent bands.



(d) 50 independent bands.

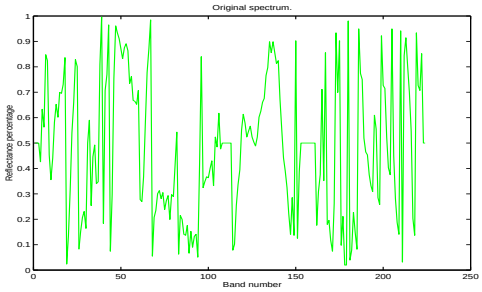


(e) 30 independent bands.

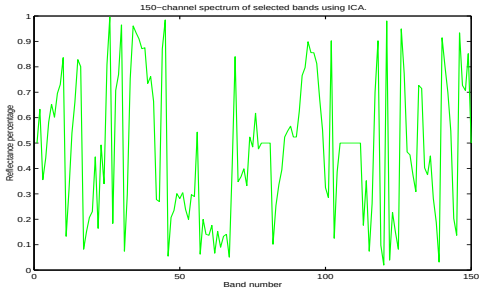


(f) 10 independent bands

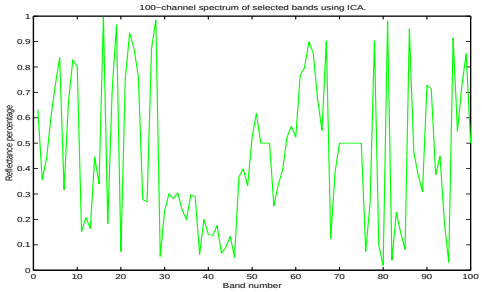
Figure 5.7: The ICA selected independent bands for the AVIRIS hyperspectral image.



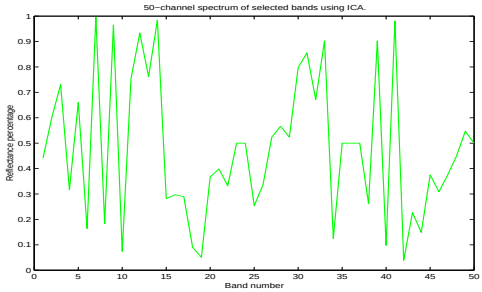
(a) Original 224-channel spectrum curve



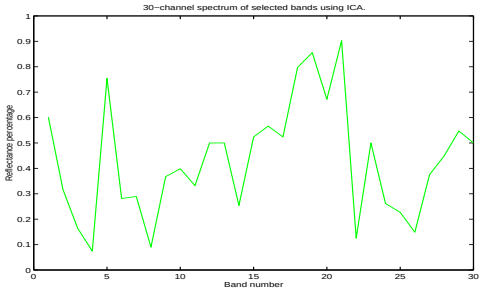
(b) 150-channel spectrum curve



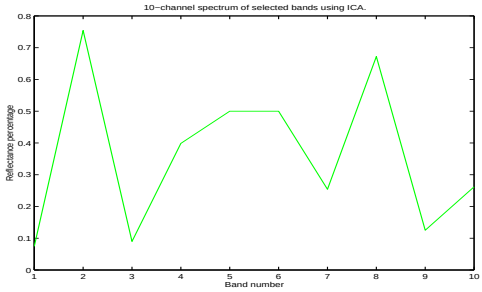
(c) 100-channel spectrum curve



(d) 50-channel spectrum curve

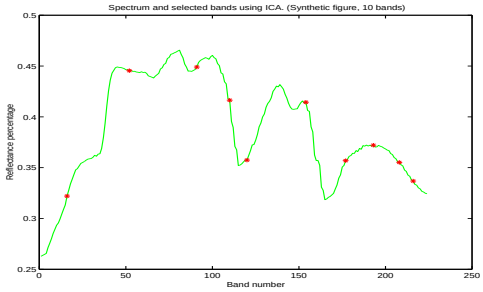


(e) 30-channel spectrum curve

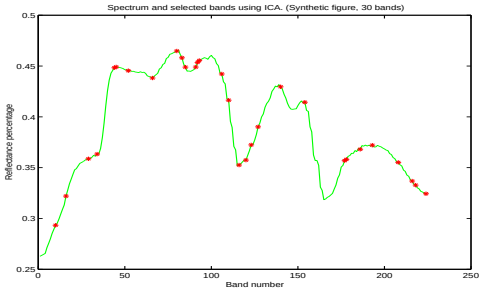


(f) 10-channel spectrum curve

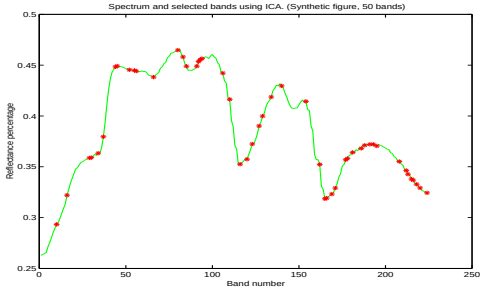
Figure 5.8: Spectral profiles from the original image and the band-selected image from Figure 5.7.



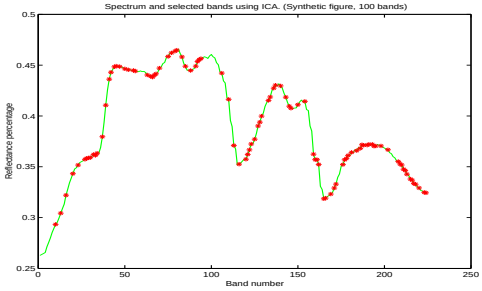
(a) 10 independent bands.



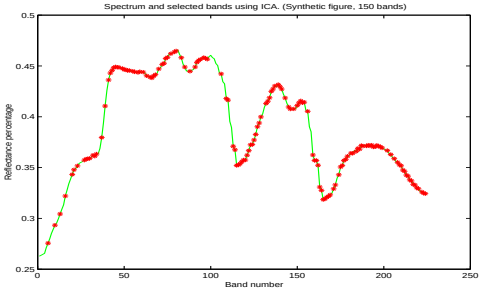
(b) 30 independent bands.



(c) 50 independent bands.

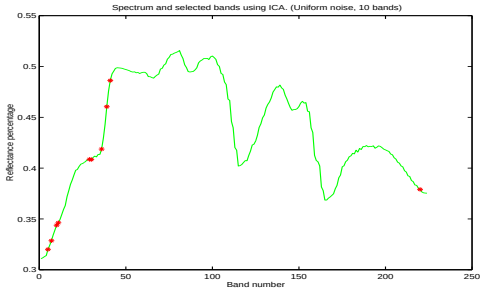


(d) 100 independent bands.

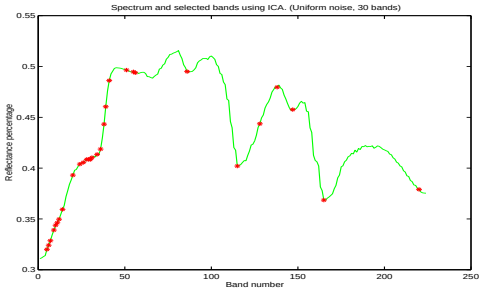


(e) 150 independent bands.

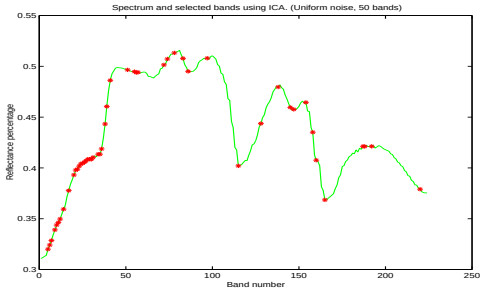
Figure 5.9: The ICA selected independent bands for the synthetic hyperspectral image without noise.



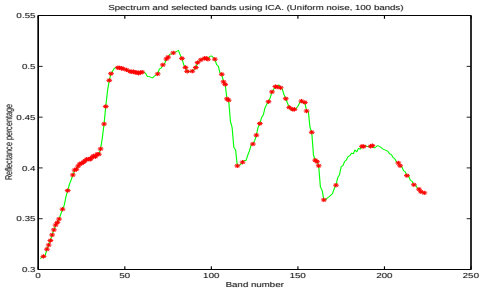
(a) 10 independent bands



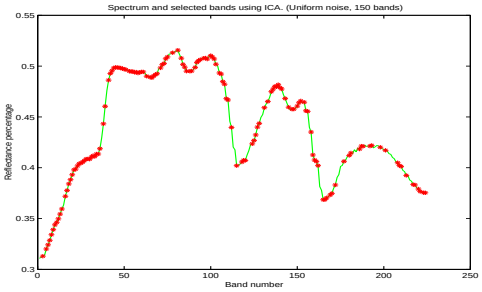
(b) 30 independent bands.



(c) 50 independent bands.

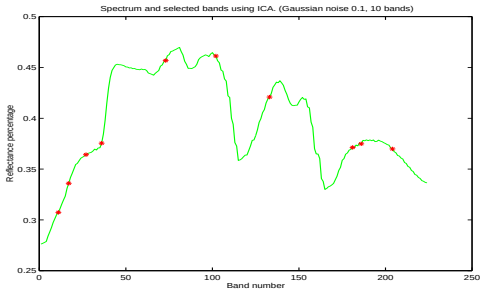


(d) 100 independent bands.

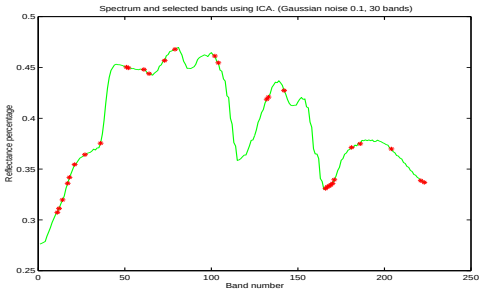


(e) 150 independent bands.

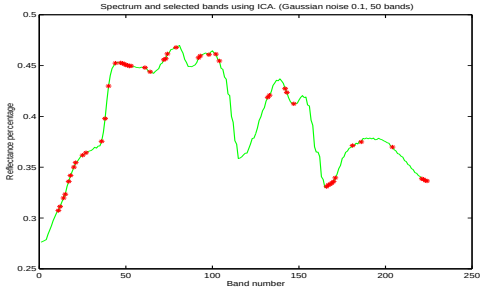
Figure 5.10: The ICA selected independent bands for the synthetic hyperspectral image with uniform noise $(0 \sim 0.1)$.



(a) 10 independent bands

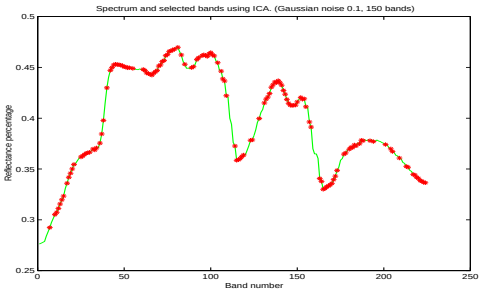


(b) 30 independent bands.



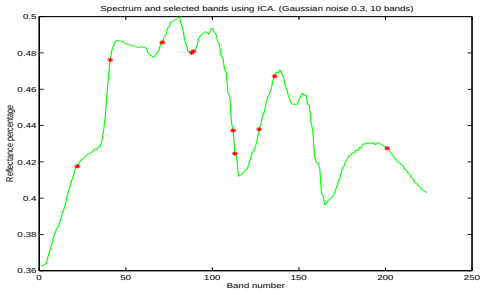
(c) 50 independent bands.

(d) 100 independent bands.

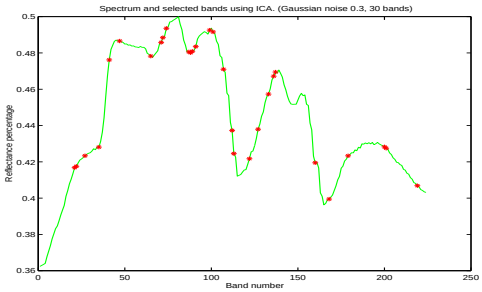


(e) 150 independent bands.

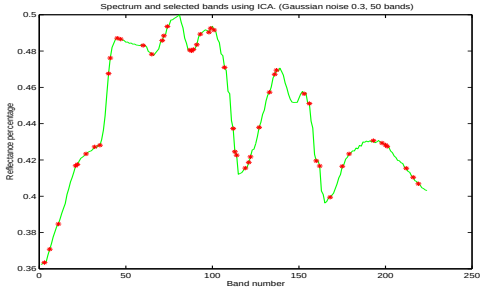
Figure 5.11: The ICA selected independent bands for the synthetic hyperspectral image with Gaussian noise ($\sigma = 0.1$).



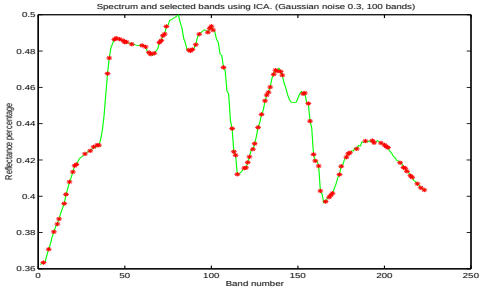
(a) 10 independent bands



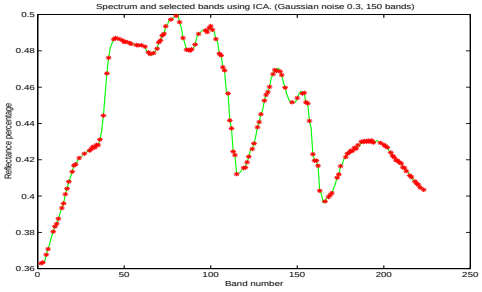
(b) 30 independent bands.



(c) 50 independent bands.



(d) 100 independent bands.



(e) 150 independent bands.

Figure 5.12: The ICA selected independent bands for the synthetic hyperspectral image with Gaussian noise ($\sigma = 0.3$).

estimation on one computer, to the computing time of the slowest external decorrelation process on one computer.

Figure 5.13 demonstrates the computation time of the FastICA and the parallel

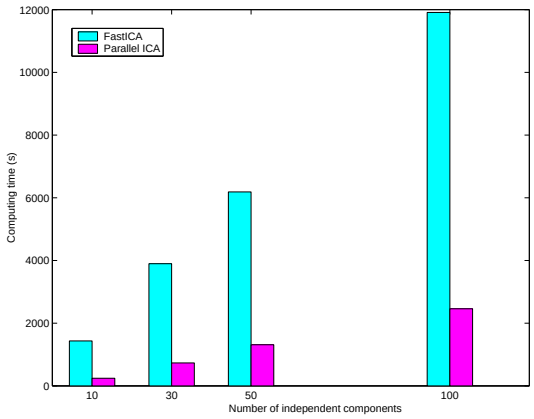


Figure 5.13: Computation time comparison between FastICA and parallel ICA.

ICA. For the processes of estimating 10, 30, 50 and 100 independent components, parallel ICA only consumes about one fifth computation time required by FastICA.

5.3 Effect of Band Selection Using Unsupervised Classifiers

The second experiment conducted in this thesis is to use unsupervised classification to evaluate the effect of using the band-selected image set vs. original image set. We only use the multispectral data set in this experiment.

For the 6-band NCSU multispectral image set, we set the cluster number to 10 according to the material categories we describe before. The classification results of the three unsupervised classifiers (k-means, winner-take-all, and Kohonen maps) are re-

spectively shown using the label images in Fig. 5.14. These label images are generated

(a) k-means (b) Winner-take-all. (c) Kohonen maps.

Figure 5.14: Unsupervised classifications on the NCSU multispectral image.

by representing different clusters with different gray scales.

k-means classifies most pixels of the truck (target 2) to the category of surrounding grass, while winner-take-all has a better classification result on the truck (target 2) but confuses the background. Kohonen maps best represents the original multispectral images.

As we have presented before, for the multispectral image, ICA selects the band of red, blue and the two IR bands of $4.2 \sim 5\mu$m and $3.2 \sim 4.2\mu$m from the original 6 bands. To see if the band selection process has any effect on the classification, we also apply the three unsupervised classifiers to the 4-band multispectral image and show the corresponding label images in Fig. 5.15. Obviously, the band selection least affects the classification result from k-means. For the winner-take-all, although the band selection does not take much effect on the classification of the background, it distinguishes the target from the surrounding grass a little bit better. The Kohonen maps is seriously affected by the band selection. In Kohonen maps, each input pixel modifies both the BMU and its surrounding neurons, thus 4-band pixels are less efficient to separate neu-

(a) k-means (b) Winner-take-all. (c) Kohonen maps.

Figure 5.15: Unsupervised classifications on the 4-band NCSU multispectral image.

rons than the 6-band pixels do. As a result, most pixels of the target are classified to the category of surrounding grass.

5.4 Effect of Band Selection Using Supervised Classifier

The third experiment we conduct in this thesis is to use supervised classification to evaluate the effect of using the band-selected image set vs. original image set. We use the multispectral data set and the synthetic hyperspectral data sets in this experiment.

5.4.1 Experiments on NCSU Multispectral Image

Training Sets and Testing Sets

To employ the supervised classification, we first extract a training set from the multispectral image data set. In this work, 1,000 samples are extracted for each of the 14 categories of materials. Consequently, we randomly divide the total 14,000 samples into 10 sets, each of which containing 1,400 samples. Alternately, one of these 10 sets

is utilized as testing set in the following classification, while other 9 sets are combined to form the training set.

Classification on 6-band Multispectral Image and Result Evaluation

On these generated training sets and testing sets, we employ kNN classifier and set $k = 20$. We use Euclidean distance for distance comparison.

After classifying all samples, we obtain an overall accuracy rate of 0.744571 for all 14 categories. The accuracy rate of each category is compared in Fig. 5.16. Comparing all accuracy rates, we find that categories of sky, brick and metal possess the highest accuracy rates, as they are least mixed with any other material. For the categories of the object mixtures and the target bottom, samples in which would highly mix with other materials, the accuracy rates are as low as 0.393899 and 0.416817, respectively.

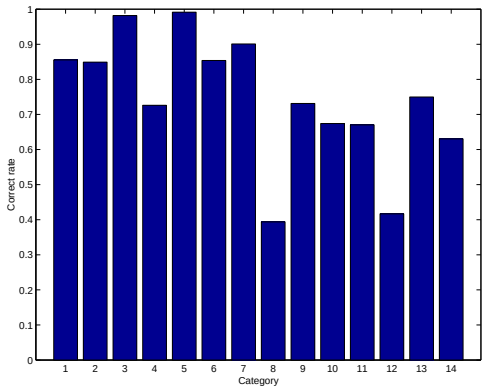


Figure 5.16: Accuracy rate for each category.

For more detailed analysis, the classification results are demonstrated in the confusion matrix, shown in Table 5.4.

As displayed in Fig. 5.17, in the classification results of the target2 top (truck cargo, category 11), the category of target3 top (cold object, category 13) includes the most

Table 5.4: Confusion matrix of kNN classification.

	Classified to category	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Correct rate
From category																
1		855	19	6	14	0	2	16	5	13	32	10	14	0	14	0.855382
2		44	849	5	4	0	4	16	8	3	0	17	0	21	29	0.848984
3		8	5	982	0	0	0	4	0	1	0	0	0	0	0	0.981801
4		32	34	14	728	0	1	58	21	14	14	47	17	2	18	0.726069
5		0	0	0	0	991	0	0	0	0	0	2	3	3	1	0.991369
6		58	30	29	4	0	853	8	7	5	0	0	5	0	1	0.853840
7		10	25	8	7	0	0	901	3	25	9	0	7	0	5	0.900926
8		55	89	80	62	0	40	90	393	23	53	24	43	15	33	0.393899
9		28	11	16	5	33	61	35	0	731	0	29	2	33	16	0.731428
10		14	15	12	12	0	12	39	50	78	674	24	21	2	47	0.674020
11		4	75	0	22	0	0	4	7	33	13	673	10	109	50	0.670207
12		21	72	4	31	0	25	62	50	49	47	120	415	63	41	0.416817
13		0	103	0	1	1	3	0	1	11	0	105	5	748	22	0.749748
14		5	45	0	14	0	10	35	17	63	75	43	24	38	631	0.630413

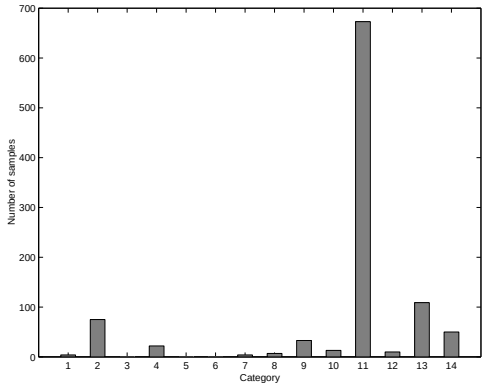
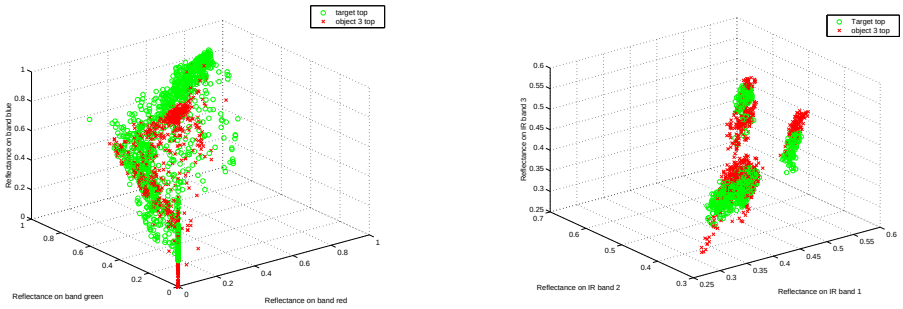


Figure 5.17: Classification results of target top.

misclassified samples. Analyzing the classification results of object3 top, we find that the category of the target top just includes the most misclassified samples. The reason is that samples of these two categories overlap seriously, as shown in Fig. 5.18.

Surrounded by grass, the target top have many boundary pixels overlapping (or mixing) with grass. Therefore, the category of grass contains 75 misclassified samples from the target top. The distributions of the categories of target top and grass are displayed in Fig. 5.19. Even though these two categories highly overlap in the RGB bands, they are a little further away from each other in the IR bands.

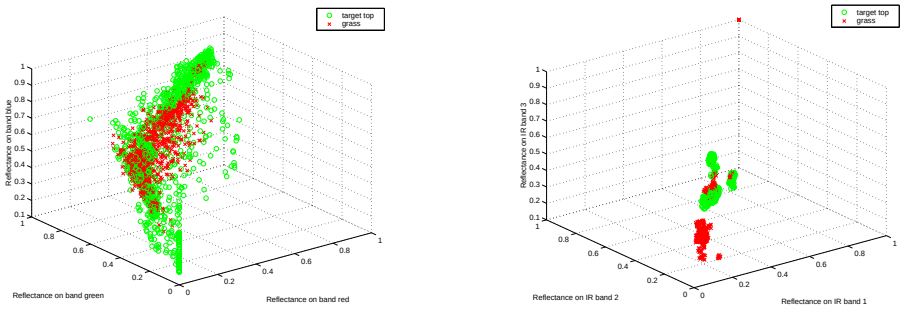
The category of the target bottom has a classification accuracy rate of 0.416817.



(a) Samples in RGB bands.

(b) Samples in IR bands.

Figure 5.18: Sample distributions of categories of the target top and the object3 top.



(a) Samples in RGB bands.

(b) Samples in IR bands.

Figure 5.19: Sample distributions of categories of the target top and grass.

As shown in Fig. 5.20, in the classification results of the target bottom, the category of the target top contains the most misclassified samples. As displayed in Fig. 5.21, samples of these two categories overlap in some regions. The reason is that these two categories have same material in common. The yellow-color cargo of the toy truck is made of steel, while the wheel of the truck contains one steel axis in yellow color as well. Obviously, there are more yellow steel samples in the category of the truck top than those in the category of truck bottom. Therefore, even though some samples of

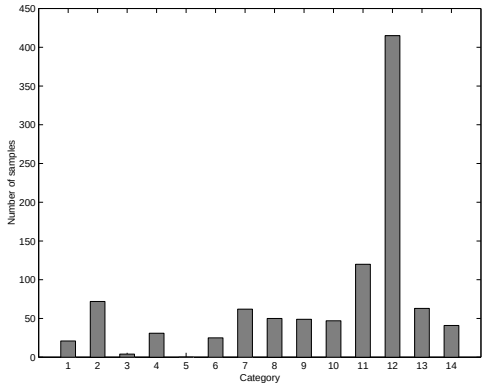
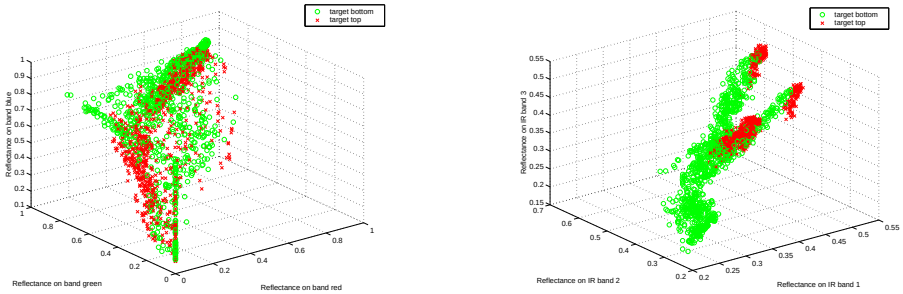


Figure 5.20: Classification results of target bottom.



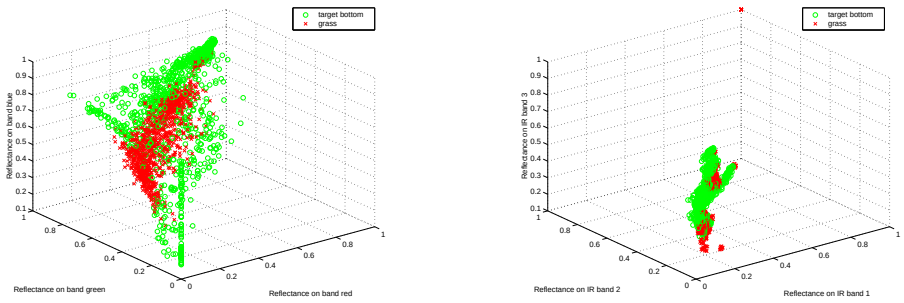
(a) Samples in RGB bands.

(b) Samples in IR bands.

Figure 5.21: Sample distributions of categories of the target bottom and the target top.

these two categories overlap, few samples of the truck top are misclassified to the truck bottom.

Almost covered by grass, the target bottom has the same problem as the target top. Therefore, the category of grass contains 72 misclassified samples from the target bottom. The distributions of these two categories are shown in Fig. 5.22.



(a) Samples in RGB bands. (b) Samples in IR bands.

Figure 5.22: Sample distributions of categories of the target bottom and the grass.

Classification on 4-band Multispectral Image and Results Comparisons

To evaluate the effect of band selection on the supervised classification, we apply the kNN classifier to the 4-band multispectral image as well, and obtain an overall accuracy rate of 0.719357. As a benefit, the time spent in classification of the 4-band data set is only 77.2074% of that of the original 6-band data set.

The detailed classification results are demonstrated in a confusion matrix in Table 5.5. In Table 5.6, we compare the classification results of the 6-band data set and the

Table 5.5: Confusion matrix of kNN classification.

	Classify to category	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Accuracy rate
From category																
1		844	21	3	14	0	24	12	7	18	26	12	8	1	10	0.844023
2		80	787	4	4	0	11	16	9	3	0	43	2	12	29	0.787260
3		2	1	990	0	0	0	4	1	2	0	0	0	0	0	0.989927
4		43	25	12	740	0	2	48	22	7	10	58	15	0	18	0.738284
5		0	0	0	0	989	0	0	0	3	0	0	0	6	2	0.989147
6		51	47	24	4	0	834	13	15	0	0	0	6	3	3	0.834930
7		14	32	29	17	0	4	857	8	15	14	0	5	0	5	0.857358
8		64	57	76	68	0	38	100	408	25	48	35	41	12	28	0.408052
9		47	6	15	5	29	54	30	5	702	8	31	29	18	21	0.701830
10		16	16	12	17	0	13	52	51	88	638	35	11	1	50	0.638345
11		3	73	0	33	0	0	3	8	45	18	634	10	118	55	0.631583
12		58	84	4	35	0	34	57	52	90	82	100	291	64	49	0.291548
13		0	96	0	2	0	4	0	1	18	0	125	14	721	19	0.722348
14		3	43	2	21	0	5	35	10	46	101	57	14	27	636	0.635153

4-band data set by subtracting the 6-band confusion matrix from the 4-band confusion

matrix. In the total 14 categories, 4 of them gain a slightly better accuracy rates, while 10 categories share a slightly worse accuracy rate.

Table 5.6: Difference between two kNN classification results.

Changes	Classify to category	1	2	3	4	5	6	7	8	9	10	11	12	13	14	Accuracy rate changes
From category																
1		-11	2	-3	0	0	22	-4	2	5	-6	2	-6	1	-4	-0.011359
2		36	-62	-1	0	0	7	0	1	0	0	26	2	-9	0	-0.061724
3		-6	-4	8	0	0	0	0	1	1	0	0	0	0	0	0.008126
4		11	-9	-2	12	0	1	-10	1	-7	-4	11	-2	-2	0	0.012215
5		0	0	0	0	-2	0	0	0	3	0	-2	-3	3	1	-0.002222
6		-7	17	-5	0	0	-19	5	8	-5	0	0	1	3	2	-0.018910
7		4	7	21	10	0	4	-44	5	-10	5	0	-2	0	0	-0.043568
8		9	-32	-4	6	0	-2	10	15	2	-5	11	-2	-3	-5	0.014153
9		19	-5	-1	0	-4	-7	-5	5	-29	8	2	27	-15	5	-0.029598
10		2	1	0	5	0	1	13	1	10	-36	11	-10	-1	3	-0.035675
11		-1	-2	0	11	0	0	-1	1	12	5	-39	0	9	5	-0.038624
12		37	12	0	4	0	9	-5	2	41	35	-20	-124	1	8	-0.125269
13		0	-7	0	1	-1	1	0	0	7	0	20	9	-27	-3	-0.027400
14		-2	-2	2	7	0	-5	0	-7	-17	26	14	-10	-11	5	0.004740

For illustration purpose, the accuracy rates of the original 6-band data set and the 4-band data sets are also compared in Fig. 5.23 in bar chart. For most categories, the

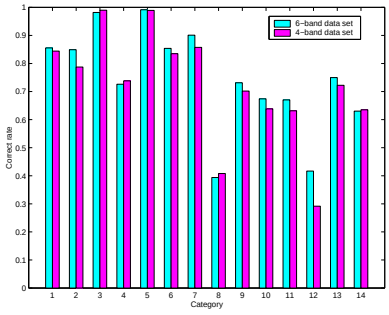


Figure 5.23: Comparison of correct rates.

difference between the classification results on the two data sets is very small, meaning that the 4-band data has kept most of the information. However, the most critical advantage of the ICA based band selection is its computation efficiency even with a little compromise on classification accuracy.

5.4.2 Experiments on Synthetic Hyperspectral Images

We apply the kNN classifier to the original 224-band and a series of band reduced synthetic hyperspectral images which are generated based on the spectral library we develop. In this spectral library, the number of samples for different categories vary greatly. For example, the vegetation category contains only 4 samples, while the rock category includes 296 sample. Therefore, we set $k = 4$ instead of $k = \sqrt{n}$ to avoid biased classification results. Based on the classification results, we then identify mixture pixels by comparing the pixel classification error rates, given in Eq. 5.4.

$$e = \frac{d - D_{min}}{D_{max} - D_{min}} \tag{5.4}$$

where $d = |x - c|$ is the deviation between the input pixel x and the nearest training sample c. $D_{max} = \max_{\forall x} d$, and $D_{min} = \min_{\forall x} d$. When we identify mixture pixels, we assume an acceptable error rate as described in Chapter 3. The acceptable error rate corresponds to the minimum acceptable distance between the testing pixel and the nearest training sample.

For the synthetic hyperspectral image without noise, we set the acceptable error rate to 1%. The classification results of 224 bands and 150, 100, 50, 30, 10 independent bands spectral images are shown respectively in Fig. 5.24. The classification results for all spectral images are 100% correct.

For the synthetic hyperspectral image with uniform noise $(0 \sim 0.1)$, we set the acceptable error rate to 35%, 30% and 20%, respectively. The corresponding classification results of 224 bands and 150, 100, 50, 30, 10 independent bands spectral images are shown respectively in Fig. 5.25, Fig. 5.26 and Fig. 5.27. When we set the accept-

(a) Original 224 bands.

(b) 150 independent bands.

(c) 100 independent bands.

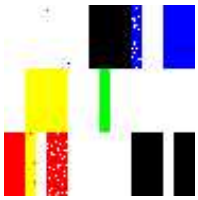
(d) 50 independent bands.

(e) 30 independent bands.

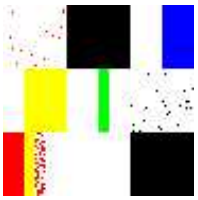
(f) 10 independent bands

Figure 5.24: Supervised classification and mixture pixels identification with 1% acceptable error rate for synthetic hyperspectral image without noise.

(a) 224 bands.



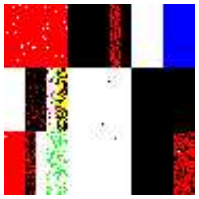
(b) 150 independent bands



(c) 100 independent bands.

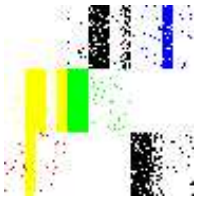
(d) 50 independent bands.

(e) 30 independent bands.

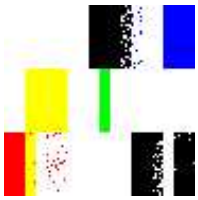


(f) 10 independent bands.

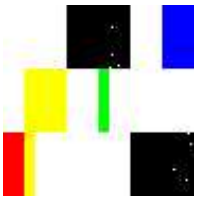
Figure 5.25: Supervised classification and mixture pixels identification with 35% acceptable error rate for synthetic hyperspectral image with uniform noise $(0 \sim 0.1)$.



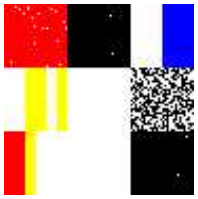
(a) 224 bands.



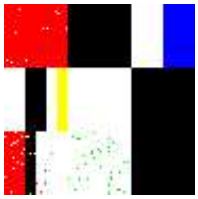
(b) 150 independent bands



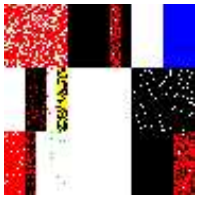
(c) 100 independent bands.



(d) 50 independent bands.



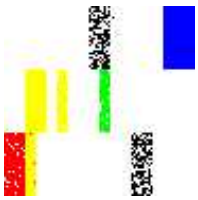
(e) 30 independent bands.



(f) 10 independent bands.

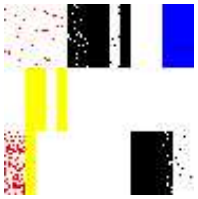
Figure 5.26: Supervised classification and mixture pixels identification with 30% acceptable error rate for synthetic hyperspectral image with uniform noise $(0 \sim 0.1)$.

(a) 224 bands.

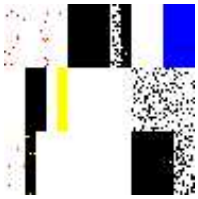


(b) 150 independent bands

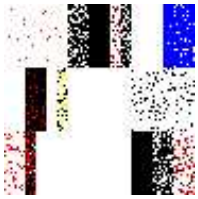
(c) 100 independent bands.



(d) 50 independent bands.



(e) 30 independent bands.



(f) 10 independent bands.

Figure 5.27: Supervised classification and mixture pixels identification with 20% acceptable error rate for synthetic hyperspectral image with uniform noise $(0 \sim 0.1)$.

able error rate as high as 35%, most pixels with 224 bands are classified to the correct categories. With the number of bands decreasing, more mixture pixels are classified to certain pure material category, while more pure pixels are treated as mixture pixels. However, if we decrease the acceptable error rate to 30% and 25%, the 224 band hyperspectral images generate a worse classification accuracy rate, while band reduced spectral images can generate a higher accuracy rate.

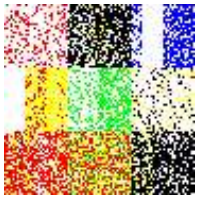
For the synthetic hyperspectral image with Gaussian noise ($\sigma = 0.1$), we set the acceptable error rate to 30%, 20% and 15%, respectively. The corresponding classification results of 224 bands and 150, 100, 50, 30, 10 independent bands spectral images are shown respectively in Fig. 5.28, Fig. 5.29 and Fig. 5.30.

For the synthetic hyperspectral image with Gaussian noise ($\sigma = 0.3$), we set the acceptable error rate to 20%, 15% and 10%, respectively. The corresponding classification results of 224 bands and 150, 100, 50, 30, 10 independent bands spectral images are shown respectively in Fig. 5.31, Fig. 5.32 and Fig. 5.33.

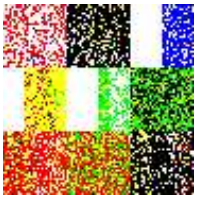
5.5 Synthesis on FPGA

To speed up the band selection process, the parallel ICA is synthesized on Xilinx V1000EHQ240-6 using re-configurable components presented in Chapter 4. For the parallel ICA procedure, 3 re-configurable components, one-unit process, decorrelation process and comparison process are employed. These re-configurable components are configured and connected by a top level block. Figure 5.34 shows the coverage of the reconfigurable components and the top level. The coverage parameter indicates the efficiency of the corresponding design. From Fig. 5.34, we observe that the coverage of

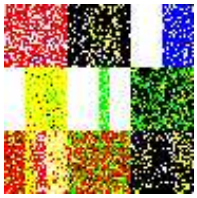
(a) 224 bands.



(b) 150 independent bands

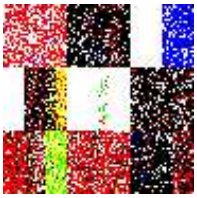


(c) 100 independent bands.



(d) 50 independent bands.

(e) 30 independent bands.



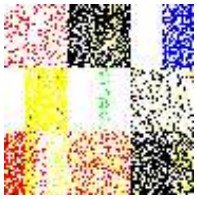
(f) 10 independent bands.

Figure 5.28: Supervised classification and mixture pixels identification with 30% acceptable error rate for synthetic hyperspectral image with Gaussian noise (STD 0.1).

(a) 224 bands.

(b) 150 independent bands

(c) 100 independent bands.



(d) 50 independent bands.

(e) 30 independent bands.

(f) 10 independent bands.

Figure 5.29: Supervised classification and mixture pixels identification with 20% acceptable error rate for synthetic hyperspectral image with Gaussian noise (STD 0.1).

(a) 224 bands.

(b) 150 independent bands

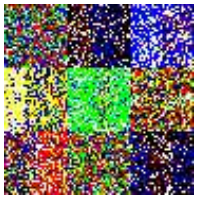
(c) 100 independent bands.

(d) 50 independent bands.

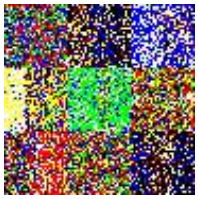
(e) 30 independent bands.

(f) 10 independent bands.

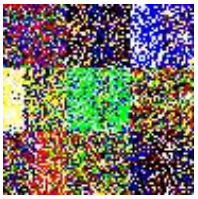
Figure 5.30: Supervised classification and mixture pixels identification with 15% acceptable error rate for synthetic hyperspectral image with Gaussian noise (STD 0.1).



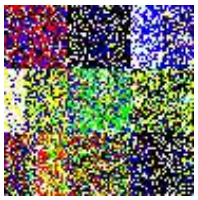
(a) 224 bands.



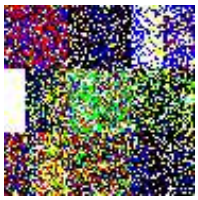
(b) 150 independent bands



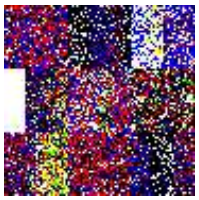
(c) 100 independent bands.



(d) 50 independent bands.

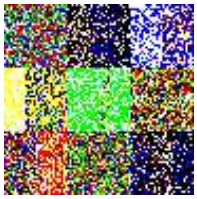


(e) 30 independent bands.

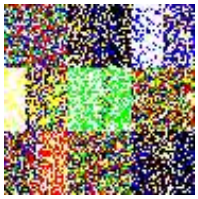


(f) 10 independent bands.

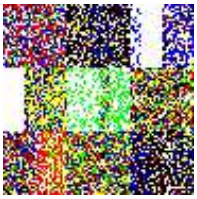
Figure 5.31: Supervised classification and mixture pixels identification with 20% acceptable error rate for synthetic hyperspectral image with Gaussian noise (STD 0.3).



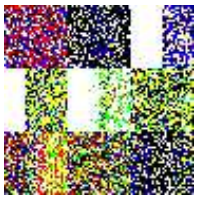
(a) 224 bands.



(b) 150 independent bands

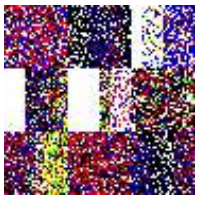


(c) 100 independent bands.



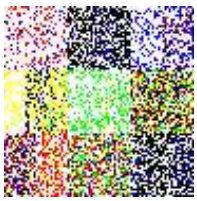
(d) 50 independent bands.

(e) 30 independent bands.



(f) 10 independent bands.

Figure 5.32: Supervised classification and mixture pixels identification with 15% acceptable error rate for synthetic hyperspectral image with Gaussian noise (STD 0.3).



(a) 224 bands.

(b) 150 independent bands

(c) 100 independent bands.

(d) 50 independent bands.

(e) 30 independent bands.

(f) 10 independent bands.

Figure 5.33: Supervised classification and mixture pixels identification with 10% acceptable error rate for synthetic hyperspectral image with Gaussian noise (STD 0.3).

top level and most re-configurable components reach 100%.

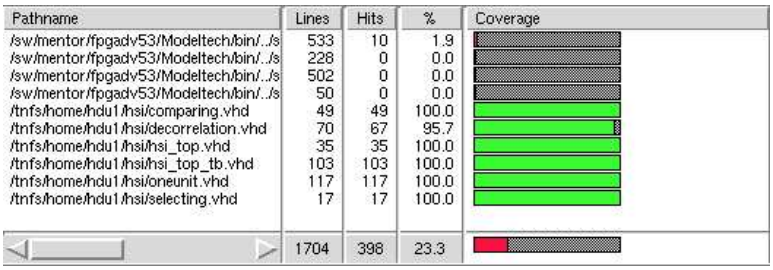


Figure 5.34: Coverage of the top level and re-configurable components.

To test the functionality of our design, we first execute the prelayout simulations on the three re-configurable components respectively and finally on the whole design. As demonstrated in Fig. 5.35, the design of parallel ICA outputs the band 1, 6, 3 and

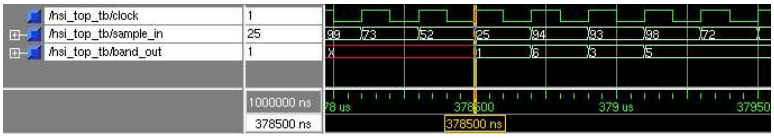


Figure 5.35: Prelayout simulation of the parallel ICA based band selection (I/O).

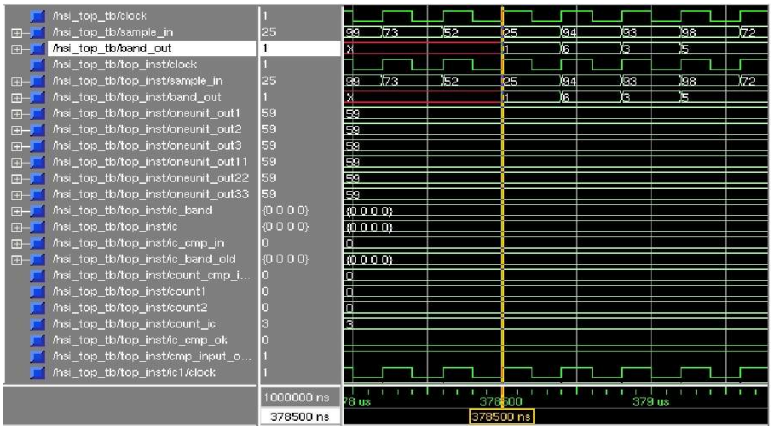


Figure 5.36: Prelayout simulation of the ICA based band selection (Internal signals).

5 which correspond to the band of red, the IR band of $3.2 \sim 4.2\mu$m, blue, and the IR

bands of $4.2 \sim 5\mu$m respectively, which align with the result obtained from software implementation. Figure 5.36 shows the internal signals used in our design. From this figure, we can check the efficiency of the internal process and analyze which process causing the most delay.

After the prelayout simulation, we used the Synopsys FC2 to synthesize the design of parallel ICA on the Xilinx VIRTEXE V1000EHQ240-6, and set clock frequency to 20 MHz. The design and device utilization are listed in Table 5.7. The parallel ICA uses 92% slices of the V1000EHQ240-6. In the consequent placement and routing

Item	Usage	Usage Percentage
Number of Slices: Number of Slices containing unrelated logic	11,318 out of 12,288 0 out of 4,774	92% 0%
Number of Slice Flip Flops	6,061 out of 24,576	24%
Total Number 4 input LUTs Number used as LUTs Number used as a route-thru	19,114 out of 24,576 18,259 855	77%
Number of bonded IOBs	32 out of 158	20%
Number of GCLKs	1 out of 4	25%
Number of GCLKIOBs	1 out of 4	25%
Total equivalent gate count for design	229,500	
Additional JTAG gate count for IOBs	1,584	

Table 5.7: Design and device utilization report.

process, our design achieves 100% coverage, with 129,753,145,344 paths, 26,884 nets, and 73,169 connections. A brief summary of our synthesis is listed below.

- Minimum period: 49.600ns (Maximum frequency: 20.161MHz)
- Maximum net delay: 13.119ns
- Number of Slices: 11,318 out of 12,288 92% usage

Finally, our design of the parallel ICA is synthesized on Xilinx V1000E. The physical layout is shown in Fig. 5.37.

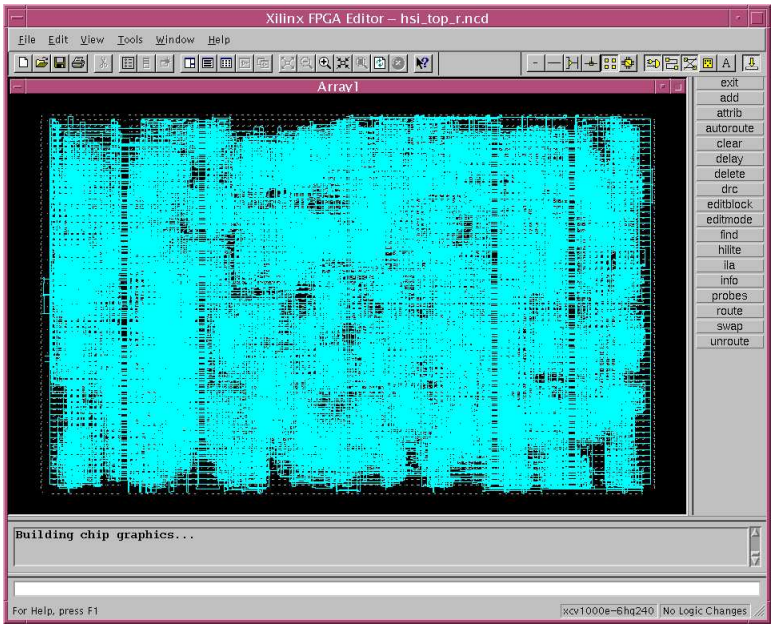


Figure 5.37: Layout of Xilinx V1000E.

Based on the synthesis aiming at the Xilinx V1000E, we then perform the postlayout simulation to verify the functionality on circuit level. Figure 5.38 illustrates the inputs and outputs of the parallel ICA synthesized on Xilinx V1000E.

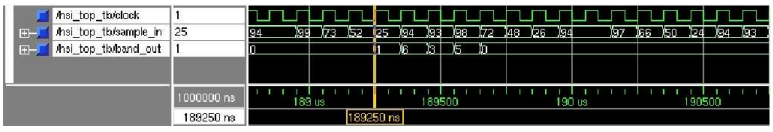
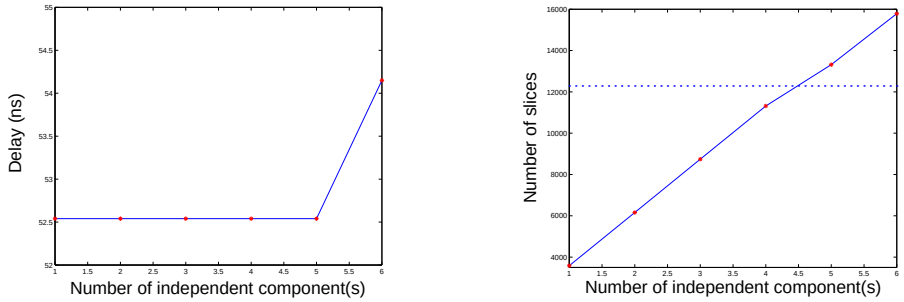


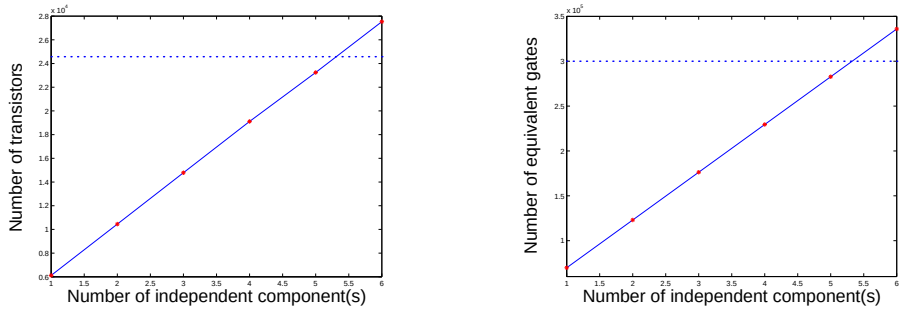
Figure 5.38: Postlayout simulation for Xilinx V1000E (I/O).

During the processes of simulation, placement and routing, we notice the capacity of single FPGA is very limited for some complex designs such as parallel ICA. Based on the synthesis of parallel ICA, the relationship between the number of the independent components and the capacity utilization of the FPGA Xilinx V1000E, evaluated

with delay, slice, transistor and equivalent gate, is exhibited in Fig. 5.39. The dotted lines denote the maximum capacity of the FPGA. It clearly shows that at most 4



(a) Delay (b) Slice



(c) Transistor (d) Equivalent gate

Figure 5.39: Number of independent components and capacity utilization of a single FPGA.

ICs estimations can be contained in a single FPGA. Even for one independent component estimation, the design utilizes 29% capacity of one Xilinx V1000E, as shown in Fig. 5.40 Obviously, if a more complex parallel ICA procedure needs to be synthesized on FPGA in the future, we would pursue re-configurable computing and HPRC techniques for better solutions, as described in Chapter 4.

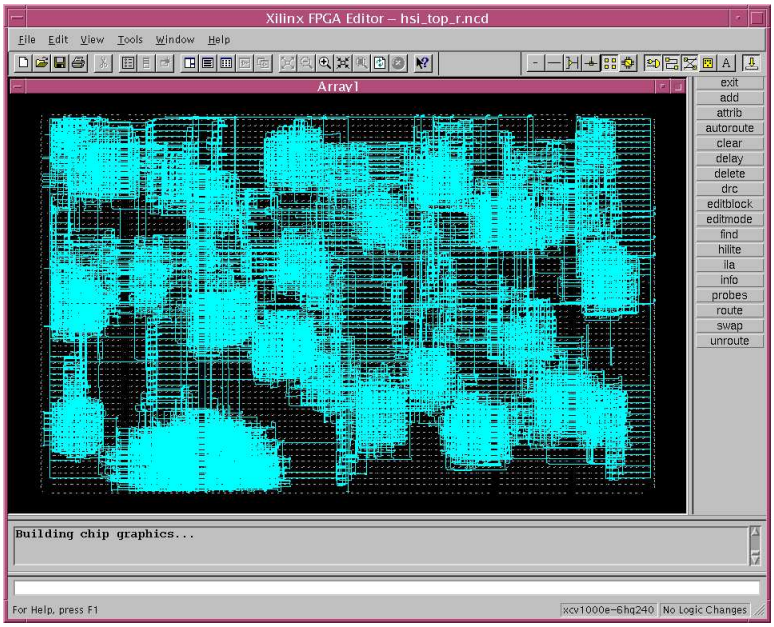


Figure 5.40: Layout of Xilinx V1000E with one independent component estimation.

5.6 Summary

In this chapter, we evaluated the performance of the parallel ICA based band selection. We performed the algorithm evaluation by comparing the unsupervised and supervised classification results on both the original and the band reduced multispectral and hyperspectral images. Using kNN, we also demonstrated the performance of mixture pixel identification based on the classification error rate. We depicted the FPGA synthesis procedure and simulations of the parallel ICA based band selection. We also presented the relationship between the number of independent components and capacity utilization of a single FPGA.

Chapter 6

Conclusions and Future Work

Hyperspectral images (HSI) analysis has been widely used in resource classification and target detection. One of the biggest challenges in HSI analysis is its high dimensionality. Independent Component Analysis (ICA) is an unsupervised signal separation algorithm. It estimates source signals given only their linear mixtures observations. This thesis concentrates on the study of dimensionality reduction using ICA-based band selection.

The contributions of this thesis include:

1. **ICA-based band selection.** This technique employs ICA to estimate the independence of individual bands, then selects the most independent bands to represent the original hyperspectral images.

2. **Parallel ICA algorithm.** This algorithm divides the unmixing matrix in ICA model into multiple sub-matrices, then estimates and decorrelates weight vectors in parallel mode. This algorithm distributes the computation burden caused by the high dimensionality to multiple CPUs.

3. **Mixture pixels Identification.** This method identifies mixture pixels in hyperspectral images by comparing the supervised classification error rates of individual pixels.

4. **Synthesis of parallel ICA based band selection on FPGA.** This procedure utilizes re-configurable components and sets up a distribute structure to implement the design. This synthesis provides hardware solution for parallel ICA and other ICA related designs.

The following conclusions are drawn from the experiments and simulations:

1. ICA based band selection plays an effective role in dimensionality reduction. It selects the independent bands containing the most important information that delicately represent the original spectra.

2. The parallel ICA significantly reduces the computation time of the ICA process.

3. For kNN classifier, the parallel ICA based band selection improves the computation efficiency with a little compromise on classification accuracy, decreasing the computation time by 22.8% but losing classification accuracy by 3.79%

4. The re-configurable component makes the design structure clearer and easier to modify. Employing re-configurable components, 70% more process in our design are re-usable.

5. A single FPGA Xilinx V1000EHQ240-6 is sufficient for a parallel ICA with at most 4 independent components estimations, which takes 92% slices of this FPGA.

Future work is need in following aspects:

1. An objective function is needed to estimate the minimum number of independent bands for specific target materials. In this thesis, the number of independent bands is assumed by prior knowledge only. If the number of independent bands is inadequate, some important information of target materials would be lost. Additional analysis of the objective function would be helpful in accurate independent band estimation.

2. Besides mixtuer pixel identification, unmixing is another challenge.

3. More re-configurable components need to be developed for ICA synthesis. This thesis only develops three re-configurable components based on parallel ICA process. More re-configurable components for other ICA algorithms would be helpful for complex designs.

4. HPRC implementation. Considering the capacity limits of a single FPGA, this thesis discusses the features and architecture of HPRC for future implementations. This implementation requires more considerations and analysis on the re-programmable feature of FPGA.

Bibliography

[1] John D. Aber and Mary E. Martin. High spectral resolution remote sensing of canopy chemistry. In *Summaries of the Fifth JPL Airborne Earth Science Workshop*, volume 1, pages 1–4. JPL Publication, January 1995.

[2] Michael Alexander. *Reconfigurable Computing*. Washington State University, `http://www.eecs.wsu.edu/~reconfig/`.

[3] Peter J. Ashenden. *The designer's guide to VHDL*. Morgan Kaufmann Publushers, 1996.

[4] M. Stewart Bartlett and T. J. Sejnowski. *Viewpoint invariant face recognition using independent component analysis and attractor networks*, chapter Neural Information Processing Systems-Natural and Synthetic 9, pages 817–823. MIT Press, Cambridge, MA, 1997.

[5] J. D. Bayliss, J. A. Gualtieri, and R. F. Cromp. Analyzing hyperspectral data with independent component analysis. In *Proc. SPIE Applied Image and Pattern Recognition Workshop*, October 1997.

[6] E. Ben-Dor, K. Patkin, R. Richter, A. Müller, and H. Kaufmann. Mapping of several soil properties using dais-7915 data. In *Proc. 20th EARSeL/ESA-Symp*, Dresden Germany, 2001.

[7] Don Bouldin. Developments in design reuse. Technical report, University of Tennessee, 2001.

[8] Don Bouldin. ECE 551: Designing application-specific integrated circuits, Fall 2001.

[9] Don Bouldin. *Design of Systems on a Chip*, chapter Synthesis of FPGAs and Testable ASICs. Kluwer Academic Press, 2003.

[10] J. Bowles, P. Palmadesso, J. Antoniades, and M. Baumback. Use of filter vectors in hyperspectral data analysis. In *SPIE: Infrared Spaceborne Remote Sensing III*, volume 2553, pages 148–157, San Diego, CA, 12-14 July 1995.

[11] Smith M. C., Drager, and etc. High performance reconfigurable computing systems. In *Circuits and Systems, 2001. MWSCAS 2001. Proceedings of the 44th IEEE 2001*, volume 1, pages 462 –465, 2001.

[12] Chein-I Chang and Hsuan Ren. An experiment-based quantitative and comparative analysis of target detection and image calssification algorithms for hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 38(2):1044–1063, March 2000.

[13] Joseph C.Harsanyi and Chein-I Chang. Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach. *IEEE Transactions on Geoscience and Remote Sensing*, 32(4):779–785, July 1994.

[14] Shao-Shan Chiang and Chein-I Chang. Subpixel detection form hyperspectral images using projection pursuit. In *EUROPTO Conference on Image and Signal Processing for Remote Sensing V*, volume 3871, pages 107–115, Florence, Italy, September 1999. SPIE.

[15] Shao-Shan Chiang, Chein-I Chang, and Irving W. Ginsberg. Unsupervised hyperspectral image analysis using independent component analysis. In *Geoscience and Remote Sensing Symposium, 2000. Proceedings. IGARSS 2000*, volume 4, pages 3136 – 3138, Honolulu, HI, USA, 24-28 July 2000. IEEE.

[16] R.N. Clark and G.A Swayze. Mapping minerals, amorphous materials, environmental materials, vegetation, water, ice, snow and other materials: The usgs tricorder algorithm. In *Summaries of the Fifth Annual JPL Airborne Earth Science Workshop*, volume 1, pages 39–40. JPL Publication, 1 1995.

[17] Pierre Common. Independent component analysis, a new concept? *Signal Processing*, 36(3):287–314, April 1994. Special Issue on High-order Statistics.

[18] Virtual Computer Corporation. Field programmable gate arrays (fpgas): An enabling technology. Technical report, Virtual Computer Corporation, 2000.

[19] Thomas M. Cover and Joy A. Thmoas. *Element of Information Theory*. John Wiley & Sons, 1991.

[20] Richard O. Duda, Peter E Hart, and David G. Stork. *Pattern Classification*. Wiley-Interscience Publication, second edition, 2000.

[21] Deniel D. Gajdki and Loganath Ramachandran. Introduction to high-level synthesis. *IEEE Design and Test of Computers*, pages 44–54, 1994.

[22] PCI Geomatics. K-nearest neighbour supervised classifier. Technical report, PCI Geomatics, http://www.pcigeomatics.com/cgi-bin/pcihlp/KNN, 2000.

[23] J. Gualtieri and R. Cromp. Support vector machines for hyperspectral remote sensing classification. In R. J. Merisko, editor, *Proceeding of the SPIE*, volume 3584, pages 221–232, citeseer.nj.nec.com/gualtieri98support.html, 1998. 27'th AIPR Workshop, Advances in Computer Assisted Recognition.

[24] A. Huertas, R. Nevatia, and D. Landgrebe. Use of hyperspectral data with intensity images for automatic building modeling. In *Second International Conference on Information Fusion*, Sunnyvale, CA, USA, July 6-8 1999.

[25] Aapo Hyvärinen. Blind source separation by nonstationarity of variance: A cumulate-based approach. *IEEE Transactions on neural networks*, 12(6):1471–1474, November 2001.

[26] Aapo Hyvärinen and Erkki Oja. A fast fixed-point algorithm for independent component analysis. *Neural Computation*, 9:1483–1492, 1997.

[27] Aapo Hyvärunen and Erkki Oja. Independent component analysis: Algorithms and applications. *Neural Networks*, 13(4-5):411–430, 2000.

[28] Jet Propulsion Laboratory, California Institute of Technology, http://speclib.jpl.nasa.gov. *JPL Spectral Library*. Reproduced from the ASTER Spectral Library through the courtesy of the Jet Propulsion Laboratory,

California Institute of Technology, Pasadena, California. Copyright ©1999, California Institute of Technology. ALL RIGHTS RESERVED.

[29] Xiuping Jia, John A. Richards, and D.E. Ricken. *Remote Sensing Digital Image Analysis: An Introduction*. Springer, third edition, 1999.

[30] Arthur C. Kenton, Craig R. Schwartz, Robert Horvath, John N. Cederquist, Linnea S. Nooden, David R. Twede, James A. Numez, James A. Wright, John W Salisbury, and Kurt Montavon. Detection of land mines with hyperspectral data. In *SPIE Conference on Detection and Remediation Technologies for Mines and Minelike Targets IV*, volume 3710, Orlando, Florida, April 1999. SPIE.

[31] Nirmal Keshava. Best bands selection for detection in hyperspectral processing. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages 3149 –3152, 2001.

[32] Laurel Kirkland. Thermal infrared hyperspectral studies of materials in the field. Technical report, Lunar and Planetary Institute, `http://www.lpi.usra.edu/science/kirkland/Mesa/text.html`, 2001.

[33] David Landgrebe. Hyperspectral image data analysis. *IEEE Signal Processing Magazine*, 19(1):17–28, January 2002.

[34] David Landgrebe. Hyperspectral image data analysis as a high dimensional signal processing problem. *Special Issue of the IEEE Signal Processing Magazine*, 19(1):17–28, January 2002.

[35] David Landgrede. Some fundamentals and methods for hyperspectral image data analysis. In *SPIE Conference on Clinical Diagnostic Systems and Technologies*, San Jose, California, January 1999.

[36] M. Lennon, G. Mercier, M. C. Mouchot, and L. Hubert-Moy. Independent component analysis as a tool for the dimensionality reduction and the representation of hyperspectral images. In *SPIE Remote Sensing*, volume 4541, pages 2893–2895, Toulouse, France, Sept 19-21 2001.

[37] Jimenez Luis and David Landgrebe. Supervised classification in high dimensional space: Geometrical, statistical, and asymptotical properties of multivariate data. *IEEE Transactions on System, Man, and Cybernetics*, 28, Part C(1):39–54, February 1998.

[38] Inc. Merriam-Webster. *Merriam-Webster Dictionary*. Merriam-Webster, `http://www.m-w.com`, 2003.

[39] Louis Nagtegaal. Concept-formation with kohonen feature maps. Technical report, University of Amsterdam, `http://cf.hum.uva.nl/computerlinguistiek/louis/research1/nagtegaal.html%`, 1997.

[40] NASA, Jet Propulsion Laboratory, California Institute of Technology, `http://popo.jpl.nasa.gov/html/aviris.freedata.html`. *AVIRIS Free Standard Data Products*, 1997.

[41] NASA, Jet Propulsion Laboratory, California Institute of Technology, `http://popo.jpl.nasa.gov/html/aviris.concept.html`. *AVIRIS concept*, 2001.

[42] NASA, Jet Propulsion Laboratory, California Institute of Technology, `http://popo.jpl.nasa.gov/html/aviris.cube.html`. *AVIRIS Moffett Field Image Cube*, 2001.

[43] Stuart Newton and David Vornholt. Reconfigurable computing for 3-d medical imaging. Technical report, Medical Electronics Manufacturing, 2001.

[44] Gregory D. Peterson and Melissa C. Smith. Programming high performance reconfigurable computers. In *International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, SSGRR 2001*, L'Aquila, Italy, Aug 06-12 2001.

[45] Hairong Qi. *Handout of ECE571: Pattern Recognition*. Dept. of Electrical and Computer Engineering, Univ. of Tennessee, Knoxville, Spring 2002.

[46] Hairong Qi and Wesley E. Snyder. Quarterly report: Smart automated target recognition using weighted spectral and geometric information. Other research members: Rajeev Ramanath, Cheolha Pedro Lee and Hongtao Du, July 2002.

[47] Hairong Qi and Wesley E. Snyder. Quarterly report: Smart automated target recognition using weighted spectral and geometric information. Other research members: Rajeev Ramanath, Cheolha Pedro Lee and Hongtao Du, December 2002.

[48] S. A. Robila and P. K. Varshney. Target detection in hyperspectral images based on independent component analysis. In *SPIE AeroSense*, Orlando, FL, April 2002.

[49] Habib Youssef Sadiq M. Sait. *VLSI Physical Design Automation, Theory and Practice*. ISBN: 9810238835. World Scientific Publishing Company, June 1999.

[50] T. Schouten and M. Gebbinck. *In Neural networks: Best practice in Europe*, chapter Fuzzy classification of spectra, pages 198–201. World Scientific, 1997.

[51] Randall B. Smith. Introduction to hyperspectral imaging. Technical report, TNTmips, August 2000.

[52] Wesley E. Snyder. Ece759 statistical pattern recognition: Unsupervised learning. Reading assignment, ECE Department, North Carolina State University, `http://courses.ncsu.edu/classes/ece763001/ece759/`, 2000.

[53] K. Staenz. Classification of a hyperspectral agricultural data set using band moments for reduction of the spectral dimensionality. *Canadian Journal of Remote Sensing*, 22(3):248–257, September 1996.

[54] David Stein, Steve Stewart, Gray Gilbert, and Jon Schoonmaker. Band selection for viewing underwater objects using hyperspectral sensors. In *SPIE Conference on Airborne and In-Water Underwater Imaging*, volume 3761, pages 50–61, Denver, Colorado, July 1999.

[55] STI Government Systems. Spectral imaging. Technical report, STI Government Systems, `http://www.sti-government.com/Spectral_Imaging.html`, 2001.

[56] Albert Tebo. Imaging spectrometry for the mission to planet earth. Technical report, SPIE, http://www.spie.org/web/oer/january/jan97/tebo.html, 1997.

[57] U.S. Geological Survey, U.S. Department of the Interior, http://speclab.cr.usgs.gov/spectral-lib.html. *USGS Spectral Library*.

[58] Miguel Velez-Reyes, Luis O. Jimenez, Daphnia M. Linares, and Hector T. Velazquez. Comparison of matrix factorization algorithms for band selection in hyperspectral imagery. In *SPIE 14th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls*, volume 4049(2000), pages 288–297, Orlando, Florida, April 2000.

[59] GEO World. Hyperspectral vs. multispectral new earth imagery choices offer a spectrum of possibilities. Technical report, Geo Resources, http://www.geoplace.com/gw/2002/0210/0210hyp1.asp, 2002.

Vita

Hongtao Du was born in Beijing, P. R. China in 1974. His major interests and research areas are parallel signal processing and microelectronic systems design. In 1993, he attended Northeastern University, P. R. China, where he received a Bachelor of Science degree in 1997, and a Master of Science degree in 2000 from the Electrical Engineering department. During his undergraduate study, he designed the water processing control system, which has been practically used in Baoshan Iron and Steel Co., China. During his graduate study, he developed an optimal structure-adaptive neural network based on rough set theory. In 2001, he came to the University of Tennessee at Knoxville as a graduate student in Electrical and Computer Engineering Department. During the summer of 2001, he joined the Advanced Imaging & Collaborative Information Processing lab as a graduate research assistant where he has been working on the project Smart Automated Target Recognition using Weighted spectral and Geometric Information. In his research on hyperspectral image analysis, he proposed a new dimensionality reduction technique named ICA based band selection, presented a new parallel ICA algorithm, and successfully synthesized the parallel ICA based band selection on FPGA.