HYPERSPECTRAL IMAGE COMPRESSION

PROJECT REPORT

Submitted by

KARTHIKEYAN.M

Register No: 13MAE07

In partial fulfillment for the requirement of award of the degree

of

MASTER OF ENGINEERING

in

APPLIED ELECTRONICS

Department of Electronics and Communication Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY
(An autonomous institution affiliated to Anna University, Chennai)

COIMBATORE - 641 049

ANNA UNIVERSITY: CHENNAI 600 025

APRIL - 2015

BONAFIDE CERTIFICATE

Certified that this project report titled **"HYPERSPECTRAL IMAGE COMPRESSION"** is the bonafide work of **KARTHIKEYAN.M [Reg. No. 13MAE07]** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr.A.VASUKI
PROJECT SUPERVISOR
Department of ECE
Kumaraguru College of Technology
Coimbatore-641 049

SIGNATURE
Dr. RAJESWARI MARIAPPAN
HEAD OF THE DEPARTMENT
Department of ECE
Kumaraguru College of Technology
Coimbatore-641 049

The Candidate with **Register No. 13MAE07** was examined by us in the project viva –voice examination held on..

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

My very first gratitude goes to God Almighty for giving me life, strength and the enablement to carry out this work.

We express our sincere thanks to our beloved Joint Correspondent, **Shri. Shankar Vanavarayar,** for his kind support and for providing necessary facilities to carry out the project work. I wish to express my deep sense of gratitude to **Dr.R.S.Kumar PhD,** Principal, Kumaraguru College of Technology, Coimbatore, for providing the facilities to conduct this study.

I express my humble gratitude to **Dr.Rajeswari Mariappan PhD**, Head of the Department, Electronics and Communication Engineering, Kumaraguru College of Technology, Coimbatore, for facilitating conditions for carrying out the research work smoothly.

I wish to express my sincere thanks to **Mrs.S.Sasikala M.tech (PhD).**, Associate Professor, PG Project Co-ordinator, Department of Electronics and Communication Engineering, Kumaraguru College of Technology, Coimbatore, for her valuable suggestions.

I am indeed grateful to my project guide, **Dr.A.Vasuki PhD,** Professor, Department of Electronics and Communication Engineering, Kumaraguru College of Technology, Coimbatore, for her immense contribution, guidance, support and constructive criticism not only during this project but also during this two years of my master program.

I would like to **Prof. S. Govindaraju M.E,** for his technical guidance and many valuable suggestions provided throughout the project work.

Lastly, I would like to thank my friends and family for standing by me and encouraging me throughout the project.

ABSTRACT

Hyperspectral remote sensing endeavors to explore the surface of the Earth using sensors that generally deliver large amounts of data. The data are usually collected by a satellite or an airborne instrument and sent to a ground station that processes it. The main bottleneck of this approach is the bandwidth of channel between the satellite and the station, which considerably limits the information that can be sent and processed in real time. A possible way to overcome this problem is to include onboard computing resources able to preprocess the data, reducing its size by orders of magnitude. To serve this purpose the project needs low complexity, lossless, high quality, efficient compression scheme. This work uses two stage spectral/spatial compression techniques with spectral unmixing to reduce spectral redundancy followed by Lossless Predictive Coding based spatial compression. Linear Spectral Unmixing (LSU) relies on the identification of pure spectral signatures via endmember extraction algorithm. Thus by using this LSU method spectrally crucial information about mixed pixels and subpixel can be well preserved over compression. The project prefers computationally efficient Vertex Component analysis for end member extraction which can be implemented parallel in FPGA using systolic array design. The quality of reconstructed image from the proposed method is analyzed through simulation and compared with existing method. The quality metrics used for performance assessment are Maximum Absolute Error, Mean Absolute error, Mean Square Error and Relative Root Mean Square Error.

.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACRONYMS	ABBREVIATIONS
ATGP	Automatic Target Generation Process
AVIRIS	Airborne Visible/Infrared Imaging Spectrometer
CCSDS	Consultative Committee for Space Data Systems
DWT	Discrete Wavelet Transform
FPGA	Field Programmable Gate Array
ICA	Independent Component Analysis
IEA	Iterative Error Analysis
JPEG	Joint Photographic Expert Group
KLT	Karhunen–Loève transform
LSU	Linear Spectral Unmixing
LPC	Linear Predictive Coding
MAD	Maximum Absolute Difference
MAE	Mean Absolute Error
MSE	Mean Square Error
MEST	Manual End Member Selection Tool
PCA	Principle Component Analysis
PPI	Pixel Purity Index
RRMSE	Relative Root Mean Square Error
SPIHT	Set Partitioning In Hierarchical Trees
VCA	Vertex Component Analysis

CHAPTER 1

INTRODUCTION

1.1 HYPERSPECTRAL IMAGERY

Hyperspectral images are sequences of spatial images, generated by imaging spectrometers or sounders, which record the spectral intensities of the light reflected Because of their three dimensional nature, (two spatial and one spectral), hyperspectral images are often referred to as data cubes. Hyperspectral images are spectrally over determined; they provide ample spectral information to identify and distinguish between spectrally similar (but unique) materials. Consequently, hyperspectral imagery provides the potential for more accurate and detailed information extraction than is possible with other types of remotely sensed data.

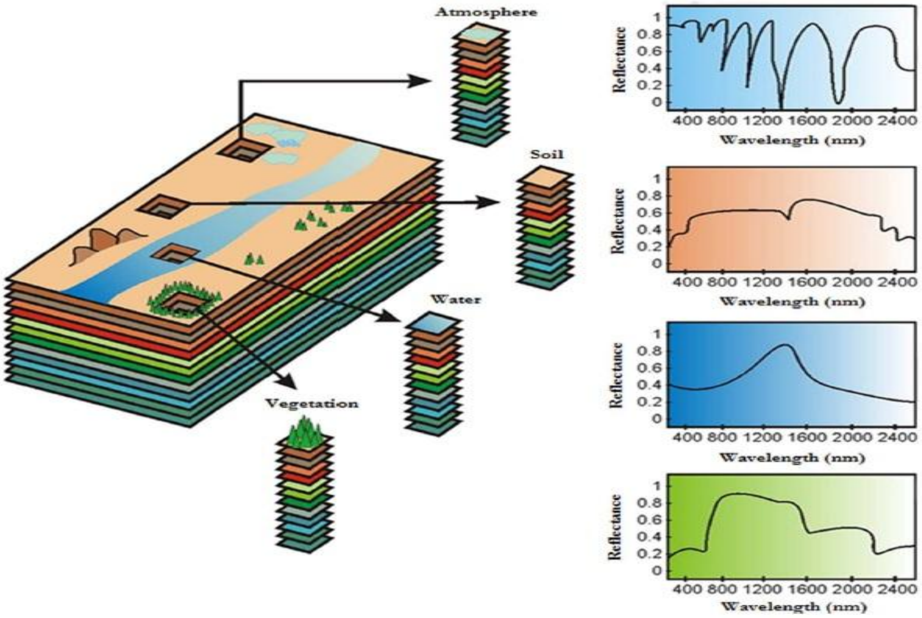


Figure 1.1. Hyperspectral image cube exposing various spectral signatures.

Most multispectral imagers (e.g. Landsat, SPOT) measure reflectance of Earth's surface material at a wide wavelength bands separated by spectral segments where, number of measurements are taken. In contrast, most hyperspectral sensors measure reflected radiation as a series of narrow and contiguous wavelength bands.

Although most hyperspectral sensors measure hundreds of bands, it is not the number of measured wavelength bands that qualifies a sensor as hyperspectral but rather the narrowness and contiguous nature of the measurements. A hyperspectral sensor is one that oversamples the phenomena of interest. Because of this, the number and spacing of bands required to qualify a sensor as hyperspectral somewhat depends on the spectral characteristics of the materials under study. In general, hyperspectral sensors measure bands at 10 to 20 nm intervals.

Projects utilizing hyperspectral imagery usually have one of the following objectives:

- Target Detection
- Material Mapping
- Material Identification
- Mapping Details Of Surface Properties

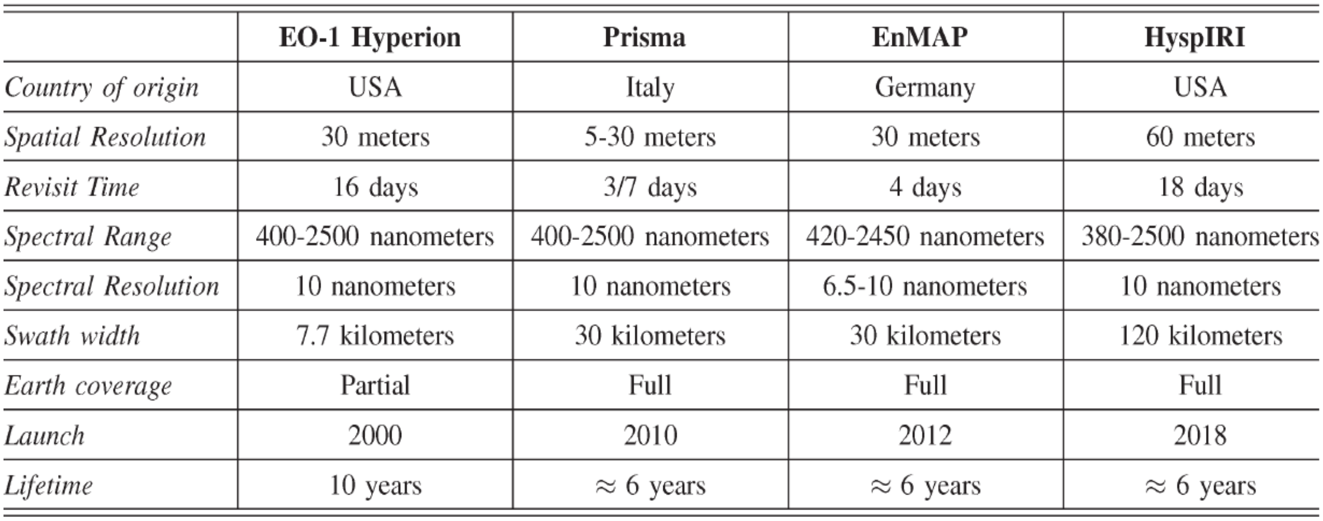
In these cases, the additional information provided by hyperspectral imagery often provides results not possible with multispectral or other types of imagery. Understanding its advantages and applications there is increasing space borne remote sensing with hyperspectral imaging instruments.

HySI (Hyper Spectral Imaging camera – ISRO chandrayaan-I) is used for mapping the lunar surface

- in 32 contiguous bands in the Very Near Infra-Red (VNIR)
- spectral range of 0.4-0.95 m region
- a spectral resolution of 15nm and spatial resolution of 80m
- Swath coverage of 20Km.

Hyperspectral raw image are made available in AVIRIS website for researchers. Hyperspectral imaging spectrometers and sounders have a large number of channels ranging from several hundred as in JPL's Airborne Visible/Infrared Imaging Spectrometer (AVIRIS), which contains 224 spectral bands, to thousands as in the case of AIRS, which has 2378 infrared channels.

Table 1.1. Overview of some hyperspectral remote sensing instruments



	EO-1 Hyperion	**Prisma**	**EnMAP**	**HyspIRI**
Country of origin	USA	Italy	Germany	USA
Spatial Resolution	30 meters	5-30 meters	30 meters	60 meters
Revisit Time	16 days	3/7 days	4 days	18 days
Spectral Range	400-2500 nanometers	400-2500 nanometers	420-2450 nanometers	380-2500 nanometers
Spectral Resolution	10 nanometers	10 nanometers	6.5-10 nanometers	10 nanometers
Swath width	7.7 kilometers	30 kilometers	30 kilometers	120 kilometers
Earth coverage	Partial	Full	Full	Full
Launch	2000	2010	2012	2018
Lifetime	10 years	$\approx$ 6 years	$\approx$ 6 years	$\approx$ 6 years

These instruments can generate high data rates that cannot be sustained with existing communication links. Efficient on-board compression is needed to reduce bandwidth and storage requirements.

1.2. HYPERSPECTRAL IMAGE COMPRESSION

Hyperspectral image compression has received considerable interest in recent years because hyperspectral imaging sensors have become very popular and the acquired hyperspectral image data are enormous and highly correlated. By realizing the importance of hyperspectral data compression, many efforts have been devoted to designing and developing compression algorithms for hyperspectral imagery]. Two types of data compression can be performed, lossless and lossy, in accordance with redundancy removal. More specifically, lossless data compression is generally considered as data compaction which eliminates unnecessary redundancy without loss of information. By contrast, lossy data compression removes unwanted redundancy or insignificant information which results in entropy reduction. Which type of compression should be used depends heavily upon the application under study. For example, in medical imaging, lossless compression is preferred. However, in this case only small compression ratios can be achieved (typically, 3:1 or below). On the other hand, video processing such as high definition television (HDTV) can greatly benefit from lossy compression. For remotely sensed imagery, both types of compression have been investigated in the past. Many algorithms have been developed for hyperspectral image compression in the past. Two approaches are generally taken [7]. One is to consider a hyperspectral image as an image cube and then apply 3-D compression which is directly extended from 2-D compression. Most notable techniques arising from the approach of this type are

- 3D-JPEG2000 which is an extension of JPEG2000
- 3D set partitioning in hierarchical trees (SPIHT)
- 3D DWT based compression

A second approach is to perform compression spectrally and spatially. More specifically, instead of performing 3D compression on an image cube as the first approach does, the spectral/spatial compression performs a two-stage compression process, spectral compression in the first stage followed by spatial compression in the second stage. In the first stage of spectral compression, principal components analysis (PCA) is generally used to de-correlate spectral information provided by hundreds of spectral bands. The resulting spectral de-correlated image cube is then compressed by either 3D or 2D compression to achieve spatial compression. Surprisingly, a simple PCA-based spectral/2D spatial compression can perform at least as well as 3D compression. This suggests that spectral/spatial compression may be more appropriate and effective than 3D compression in hyperspectral image compression [7].

1.3. PROBLEM DEFINITION

- Remote sensing mission involving hyperspectral imaging instruments will generate voluminous data. Recent internal studies at NASA estimate that volume of 1–5 TB of raw data per day can be expected from future imaging instruments (HyspIRI).
- Available bandwidth connection between the satellite and the station, which drastically limits the information that can be sent and processed in real time

1.4. OBJECTIVES OF THE PROJECT

- To compress hyper-spectral image (3d cube) while, at the same time, being able to retain information that is spectrally and spatially crucial.
- To choose or design a computationally and functionally efficient algorithm for hyper-spectral image compression.
- To incorporate spectral unmixing as preprocessing stage of compression to spectrally de-correlate hyperspectral image.
- To design FPGA architecture for the proposed algorithm for onboard real time compression of hyper-spectral image

The rest of this project report is organized as follows. Chapter 2 provides inferences from literature survey on various methods of hyperspectral image compression .Chapter 3 gives a description of existing methods of endmembers extraction. The chapter 4 describes the method of using Linear Spectral Unmixing via Vertex Component Analysis as preprocessing stage of hyperspectral image compression, method of lossless predictive coding utilized for spatial compression. The chapter 5 suggests FPGA architecture for parallel implementation of Vertex Component Analysis algorithm. Chapter 6 explains experimental results of proposed compression scheme and its comparison with existing method. Finally, in Chapter 7, conclusions drawn from the experiments are discussed.

CHAPTER 2

LITERATURE REVIEW

This chapter deals with review of literature about hyperspectral image compression, onboard compression schemes and endmember extraction algorithms and its comparisons in terms of performance and computational complexity.

Firouz Abdullah Al-Wassai [17] in her paper "Major Limitations of Satellite images" emphasis that Current sensor technology allows the deployment of high resolution satellite sensors, but there are a major limitation of Satellite Data and the Resolution Dilemma as the following:

- There is a tradeoff between spectral resolution and SNR.
- There is a tradeoff between radiometric resolution and SNR.
- There is a tradeoffs related to data volume and spatial & spectral resolution
- There is a tradeoff between the spatial and spectral resolutions.

Hyperspectral image being high resolution image, tradeoffs related to data volume and spatial & spectral resolution has to be faced. Hence hyperspectral image compression is main focus of this work.

Bharath Ramakrishna, Jing Wang [7] in their paper summarized various hyper-spectral compression techniques viz.., 'as a whole 3D image compression' and 'two stage spectral/spatial compression. Latter method is proven to be efficient and retaining spectrally crucial data. **M. Klimesh, A. Kiely, H. Xie and N. Aranki [21]** in their paper describes spectral ringing artifacts in wavelet based hyperspectral image compression and remark that using a wavelet transform for spectral decorrelation of hyperspectral data has other shortcomings as well. For example, the spectral dependencies that exist are not limited to the small spectral neighborhood exploited by the wavelet transform. **Pekka Toivanen, Olga Kubasova, and Jarno Mielikainen [24]** proposed an idea of correlation based band ordering as preprocessing step in spectral image compression. **Ahmed Hagag, Mohamed Amin, Fathi E, Abd El-Sam [2]** presents multispectral compression scheme with band ordering followed 2D wavelet transform in spectral and spatial dimensions. They infer that the correlation- based band ordering gives 5% higher compression ratios than natural ordering does.

Antonio Plaza, Sergio Sánchez, Abel Paz and Javier Plaza [5] have inter-compared FPGAs versus GPUs in the role of compressing hyperspectral data (in lossy fashion) using spectral unmixing concepts. Paper explains that both platforms offer a low-weight solution compliant with mission payload requirements, although power consumption is higher in the GPU solution than in the FPGA solution. In both cases the solutions can be easily reused to deal with other different processing problems (e.g. lossless compression).**Carole Thiebaut, Emmanuel Christophe et al [10]** in their survey paper titled "CNES Studies of On-Board Compression for Multispectral and Hyperspectral Images" presents various onboard compressions available for spectral satellite images. Multispectral and hyperspectral compression involves two redundancies spatial and spectral. Thus needs preprocessing stages to reduce spectral redundancy. Some preprocessing methods are

i. KLT(Karhunen–Loève transform)with low complexity –lossy multispectral image compression

ii. DWT (discrete wavelet transform) - hyperspectral image compression

iii. ICA (Independent Component Analysis) - both multi and hyperspectral image compression

iv. Spectral unmixing – hyperspectral image compression

i. The decorrelation stage was performed by a KLT on spectral channels. The choice of this transform is justified by several reasons: transform decorrelation performs better than a predictive method. A transform specifically suited for spectral application but KLT is signal dependent.

ii. The DWT is a very powerful tool for decorrelating data and for performing a coding of the coefficients. It is used in compression recommendations such as JPEG2000 and

CCSDS standards. It is not suited to multispectral data because of the small number of bands but could be useful for hyperspectral data. Moreover, this transform is fixed and not data dependent which is better for on-board computation.

iii. ICA matrix is performed on wavelet coefficients first, outside JPEG2000 coding procedure, and then, ICA matrix is used as the multicomponent transform by JPEG2000. The rate allocation procedure is the one of JPEG2000. An on-going study will lead to the development of a coding algorithm suited to the transformed coefficients after ICA on spectral axis and DWT on spatial one. This survey elaborates those available methods viz

- KLT followed by CCSDS compression with low complexity rate allocation algorithm.
- Spectral anisotropic DWT followed by DWT on each band
- Modified EZW AND SPIHT to 3D DWT

are performing better than JPEG 2000 for onboard implementation. Thus the two stage spectral/spatial compression may be more appropriate and effective than 3D compression in hyperspectral image compression.

Agnieszka C. Miguel, Alexander Chang et al [1] in their paper proposed wavelet based predictive coding for onboard compression. An algorithm for lossy compression of hyperspectral images to greatly reduce the bit rate required to code images, this method uses linear prediction between the bands. Once the prediction is formed, it is subtracted from the incoming band, and the residual (difference image) is compressed using the Set Partitioning in Hierarchical Trees (SPIHT) algorithm. **Nazeeh Aranki [22]** in his PhD dissertation presented many methods for parallel implementation of DWT and hyperspectral compression methodology. He proposed method of parallel implementation of 2D DWT by partitioning image into image blocks by recursive filtering i.e. lifting factorization. Disadvantage of this method is the need for data exchange between neighboring block. He also proposed method for 3D DWT implementation by cascaded line based wavelet transform and also presented an embedded and scalable SoC implementation for the ICER-3D-HW compression algorithm on FPGAs and addressed challenges related to the intensive I/O of the algorithm and the 3D nature of the data and its volume, and provided solutions to speed up the design and extend FPGA implementation methodology to a system on a chip (SoC) FPGA-based implementations. **Hassan Ghassemian [18]** in his paper investigates a new on-board unsupervised feature extraction method that reduces the complexity and costs associated with the analysis of multispectral images. It is possible to reduce data redundancy by an unsupervised object-feature extraction process, where the object-features, rather than the pixel-features, are used for multispectral scene representation. The proposed algorithm partitions the observation space into exhaustive set of disjoint objects. Then, pixels belonging to each object are characterized by object features.

Antonio Plaza, Pablo Martínez et al [4] in their paper have compared various end member extraction algorithms viz

- pixel purity index (PPI)
- N-FINDR
- manual end member selection tool (MEST)
- iterative error analysis (IEA)
- automated morphological end member extraction (AMEE)

Dominique Lavenier, Erwan Fabiani et al [14] proposed a systolic implementation of the PPI algorithm. It is based on a linear systolic array connected to a host processor through its external I/O bus system. **Antonio Plaza et al** in another paper [8] proposed a method of real time implementation of NFINDR algorithm for end member extraction in reconfigurable FPGA. Algorithm has minimal computations which makes it apt choice for FPGA implementation. Determinant is calculated using matrix triangulation method. **Antonio Plaza** in this paper [8], proposed an FPGA-based data compression technique which relies on the concept of spectral unmixing, one of the most popular approaches to deal with mixed pixels and sub pixel targets in hyperspectral analysis. The proposed method uses a two-stage approach in which the purest pixels in the image (endmembers) are first extracted and then used to express mixed pixels as linear combinations of endmembers. The procedure of the proposed onboard compression algorithm proceeds as follows 1) Endmember extraction (PPI) 2) Linear spectral unmixing 3) construct fractional abundance image and then 4) lossless predictive coding Carlos **Gonz´alez, Daniel Mozos et al [8]** presented an inter-comparison of the experimental results of FPGA implementation of the PPI and N-FINDR algorithms. Their experimental results, conducted on a Virtex-4 XC4VFX60, demonstrate that architecture can extract endmembers with highly satisfactory spectral purity. **José M. P. Nascimento and José M. Bioucas Dias [19]** in their paper proposed VCA algorithm and compared it with NFINDR and PPI. It proves that VCA is computationally efficient than NFINDR and PPI.

From the literature survey it is evident that two stage compression methods using spectral unmixing methodology are efficient over retaining spectral information. Hardware realization of VCA algorithm [20] is proves to be computationally efficient over implementing it parallel. FPGA realization of these algorithms outperformed software version and FPGA is preferable over GPU for implementing spectral unmixing techniques.

CHAPTER 3

ENDMEMBER EXTRACTION METHODS

Spectral decorrelation preprocessing stage is essential in hyperspectral compression. Some of methods are PCA, ICA, VD (virtual dimensionality and spectral unmixing via endmember extraction. Spectral unmixing method is more efficient in retaining the spectrally crucial information by capturing pure spectral signatures. Thus spectral unmixing is chosen as preprocessing step. This section provides survey of end member extraction algorithms available.

Linear unmixing is a common practice in remote sensing image analysis. Linear mixture analysis is particularly useful in the understanding of a hyperspectral image, because its high dimensionality relaxes the dimensionality limitation when linear mixture analysis is applied on traditional multispectral imagery. In many practical circumstances when endmember signatures are unknown, these signatures have to be estimated in an unsupervised fashion. Endmember extraction algorithms are designed for this purpose. Many endmember extraction algorithms identify distinctive pixels from the image data set itself as endmember signatures owing to mathematical tractability. They implicitly assume that pure pixels are present in an image scene. These algorithms may use different criteria when searching for distinctive pixels. Two major criteria are

- Multidimensional geometry- based simplex volume maximization
- Pixel spectral signature similarity.

The well-known NFINDR algorithm employs the first criterion, and it is based on the fact that in a p-dimensional space the p-volume simplex formed by the purest pixels has a larger volume than any other volume defined by any other combination of pixels. In the second criterion, a certain pixel spectral similarity metric is defined. For instance, the automatic target generation process (ATGP) searches the most distinctive pixels by using the orthogonal subspace projection (OSP) approach, and the pixel whose projection in the orthogonal subspace of a set of pixels is the one most dissimilar to these pixels will be selected as an endmember. Vertex component analysis (VCA) is similar to ATGP in adopting the idea of OSP maximization to search the vertices for a simplex.

3.1. AUTOMATED TARGET GENERATION PROCESS

The basic idea of the ATGP [11] is to search the most distinctive pixels by using the following pixel spectral similarity metric: the pixel whose projection in the orthogonal subspace of other pixels is the most dissimilar pixel. The detailed algorithm is the following.

1. Select an initial signature, denoted *e0*.
2. Apply the OSP operator $p_{\mathrm{e_o}}^{\perp}$ to all pixel vectors *r*. Set *i =0* and *U₀ =e0*. Here, $p_{\mathrm{e_o}}^{\perp} = I - e_0(e_0^{\perp}e_0)^{-1}e_0^{\perp}$ where *I* is an identity matrix.
3. Find the first endmember signature, denoted *e₁*, which has the maximum orthogonal projection $e_1 = \arg\{\max{}_r[\,(\mathrm{p}_{\mathrm{e_o}}^{\perp}\mathrm{r})^{\mathrm{T}}(\mathrm{p}_{\mathrm{e_o}}^{\perp}\mathrm{r})]\}$ .Set *i=1* and *U₁= e₁*.
4. Find the ith endmember at the ith stage by $e_i = \arg\{\max{}_r[\,(\mathrm{p}_{\mathrm{U_{i-1}}}^{\perp}\mathrm{r})^{\mathrm{T}}(\mathrm{p}_{\mathrm{U_{i-1}}}^{\perp}\mathrm{r})]\}$, where the endmember signature matrix $U_{i-1} = [e_1 e_2 \mathrm{....e_{i\text{-}1}}\,]$ is generated at the (i-1)th stage.
5. Terminate the algorithm if the number of extracted endmembers is enough, or go to step 4 for the next end member.

3.2. N-FINDR

The N-FINDR (winter, 1999) method finds the set of pixels that define the simplex with the maximum volume, potentially inscribed within the dataset. First, a dimensionality reduction of the original image is accomplished by using PCA. Next, randomly selected pixels qualify as endmembers, and a trial volume is calculated as follows. Let **E** be defined as

$$E = \begin{Vmatrix} 1 & 1 & ..... & 1 \\ e1 & e2 & ..... & eE \end{Vmatrix}$$

where *ei* are endmember column vectors, and *E* is the number of endmembers used to calculate the simplex volume. The volume of the simplex formed by the endmembers is proportional to the determinant of **E**

$$V = \frac{1}{(E-1)} abs(|E|) \tag{3.1}$$

In order to refine the initial volume estimate, a trial volume is calculated for every pixel in each endmember position by replacing that end member and recalculating the volume. If the replacement results in a volume increase, the pixel replaces the end member. This procedure, which does not require any input parameters, is repeated until there are no replacements of endmembers left. NFINDR is efficient than PPI but it is not computationally intensive. Thus for onboard parallel processing with less computations PPI is preferred for this work.

3.3. PIXEL PURITY INDEX

Pixel Purity Index (PPI), designed to search for a set of vertices of a convex hull. The concept of the original PPI is rather simple. The algorithm proceeds by generating a large number, *k*, of *L*-dimensional random vectors called “skewers.” All pixel vectors in the input data are then projected onto each of the skewers, and the pixel vectors resulting in the maximum projection scores increase their pixel purity score. After many repeated projections, those pixels with score above a certain cut-off threshold, *t*, are declared “pure.” These potential endmember spectra are then loaded into an interactive *L*-dimensional visualization tool to manually select pixel vectors that correspond to pure pixel vectors. The PPI algorithm has been widely used in the remote sensing community due to its publicity and availability, provided by Research Systems’ ENVI software package, which also provides a tool called “L *D*imensional visualizer” that, allows a trained user to select endmembers by visual inspection.

The inputs to PPI are a hyperspectral data cube **F** with *N* dimensions; a maximum number of endmembers to be extracted, *E*; the number of random skewers to be generated during the process, *K*; a cut-off threshold value, t_v, used to select as final end members only those pixels that have been selected as extreme pixels at least t_v times throughout the process; and a threshold angle, t_a, used to discard redundant endmembers. The output of the algorithm is a set of *E* final endmembers $\{\mathbf{e}_e\}$ *Ee* =1. The algorithm is summarized as follows:

1. Produce a set of *K* randomly generated unit vectors $\{\mathbf{skewer}_j\}K_j$=1.

2. For each $\mathbf{skewer}_j$, all sample pixel vectors $\mathbf{f}_i$ in the original data set **F** are projected onto $\mathbf{skewer}_j$ via dot products of $|\mathbf{f}_i \cdot \mathbf{skewer}_j|$ to find sample vectors at its extreme (maximum and minimum) projections, thus forming an extrema set for $\mathbf{skewer}_j$ which is denoted by I_S*extrema*($\mathbf{skewer}_j$). Despite the fact that a different **skewer***j* would generate a different extrema set I_S*extrema* (**skewer***j*), it is very likely that some sample vectors may appear in more than one extrema set. In order to deal with this situation, an indicator function of a set *S* is defined and is denoted by *IS*(**x**), to denote membership of an element **x** to that particular set as follows:

$$Is(\mathbf{f}i) = \begin{cases} 1, & \mathbf{x} \in S \\ 0, & \mathbf{x} \notin S \end{cases} \tag{3.2}$$

3. Calculate the PPI score associated to the pixel vector **f***i* using the following equation

$$NPPI(fi) = \sum_{j=1}^{k} ISextrema(skewerj)(fi) \tag{3.3}$$

4. Find the pixels with value of *NPPI* (**f***i*) above *tv*, and form a unique set of endmembers $\{\mathbf{e}_e\}^{Ee}{}_{=1.}$

3.4. VERTEX COMPONENT ANALYSIS

The VCA [19] is an autonomous method to unmix linear mixtures of hyperspectral datasets and is based on the geometry of convex sets. By this algorithm endmembers are extracted based on two concepts viz.

1) The endmembers are the vertices of a simplex

2) The affine transformation of a simplex is also a simplex.

VCA is a fully automatic algorithm and it works with and without dimensionality reduction preprocessing step. Has mentioned before, there are many real situations where the hyperspectral vectors live in a subspace of very low dimension compared with the available number of bands ($p \ll L$) and, thus, it is advantageous, in terms of signal-to-noise ratio (SNR), memory usage, and computational complexity, to represent the spectral vectors in a signal subspace basis. The VCA algorithm iteratively projects data onto a direction orthogonal to the subspace spanned by the endmembers already determined. The new endmember signature corresponds to the extreme of the projection. The algorithm iterates until all endmembers are exhausted. Fig. 3.1 illustrates the VCA method working on a simplex defined by a mixture of four endmembers where circles and dots represent pure-pixels (endmembers signatures) and mixed pixels [19]. In the first iteration, data is projected onto the first direction **f**1. The extreme of the projection corresponds to endmember $\mathbf{m}a$. In the next iteration, endmember $\mathbf{m}b$ is found by projecting data onto direction **f** 2, which is orthogonal to $\mathbf{m}a$. Then, a new direction **f**3, orthogonal to the subspace spanned by $\mathbf{m}a$ and $\mathbf{m}b$ is generated and the endmember $\mathbf{m}c$ is found by seeking the extreme of the projection of the dataset onto **d**3. VCA algorithm iterates until all p endmembers are found. Fig 3.1 represents Endmebers as circles and mixed pixels as dots.

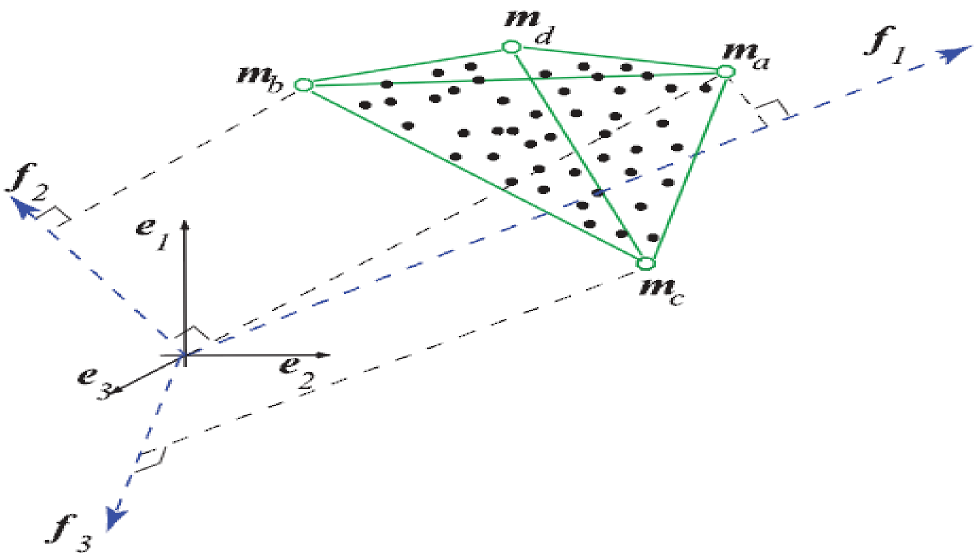


Fig 3.1 Three-dimensional diagram of a hyperspectral mixture of four endmembers illustrating the VCA algorithm.

The algorithms for end-member extraction may also differ in the implementation mode: i.e., parallel or sequential. In the parallel model, endmembers are determined simultaneously; in the sequential mode, endmembers are determined one after another. If the parallel mode is used, then the number of endmembers to be selected needs to be predetermined before algorithm execution; if the sequential mode is used, the algorithm can be terminated or re-executed with greater flexibility. The original N-FINDR algorithm is implemented in the parallel mode, and its sequential version is the simplex growing algorithm. In general, the sequential mode involves lower computational complexity, while the parallel mode may be more effective in extracting distinctive pixels. Endmember extraction algorithms may also differ in data preprocessing steps. For instance, the original N-FINDR algorithm must apply a dimensionality reduction process, such as the minimum noise fraction transform; otherwise, it cannot calculate a simplex volume due to the mathematical formula it chooses; The number of flops is measured, in order to compare the computational complexity of VCA, PPI, and N-FINDR algorithms.

Table 3.1 presents approximated expressions for the number of flops used by each algorithm provided by [19].

Table 3.1.Computational Complexity of VCA, N-FINDR and PPI algorithms

ALGORITHM	**COMPLEXITY**
VCA	$2p^2N$
PPI	$2psN$
NFINDR	$p^{\eta+1}N$

p- *no of endmembers;* ***N –*** *no of columns of 2D signature matrix;* ***2.3<η<2.9; s****-no of skewers*

The VCA algorithm projects all data (*N* vectors of size) onto orthogonal directions. N-FINDR computes *pN* times the determinant of a *p X p* matrix, whose complexity is p^η, with $2.3<\eta<2.9$. Assuming that $N>>p>2$, VCA complexity is lower than that of N-FINDR. Concerning PPI, given that the number of *skewers (s)* is much higher than the usual number of endmembers, the PPI complexity is much higher than that of VCA. From literature conclusion drawn is that the VCA algorithm has always the lowest complexity.

CHAPTER 4

METHODOLOGY

Hyperspectral image compression can be done in two-ways by considering spatial redundancy alone viz 3D-DWT, 3D-SPIHT or by considering both the spectral and spatial redundancies. Latter method of spectral/spatial compression is more appealing in retaining spectral contents after reconstruction. The preferred method involves reduction of spatial and spectral redundancy. Moreover spectral redundancy can be reduced by dimensionality reduction viz... PCA, ICA, Virtual Dimensionality etc... But these methods are computationally complex which makes hardware realization difficult and they are not specific about the spectrally crucial data in the image.

Spectral information is to be retained to recover mixed pixel and sub pixel targets from hyper spectral images. Thus spectral unmixing via end member extraction method is preferable preprocessing method for hyperspectral image compression which retains mixed pixel and sub-pixel target after reconstruction. After analyzing various methods for endmember extraction, found NFINDR and PPI algorithm to be suitable for FPGA implementation. NFINDR involves volume calculation which can be computational simplified as determinant computation via triangulation method [9] but NFINDR involves PCA as first step which makes it computational complex than PPI [3] .PPI can be implemented parallel by means of systolic array [3] which makes its realization computationally efficient and makes it possible for real time performance via parallel processing.

From the literature about spectral/spatial compression method of hyperspectral image compression it is inferred that many efforts have been devoted to designing and developing compression algorithms for hyperspectral imagery. Unfortunately, most available approaches have largely overlooked the impact of mixed pixels and sub pixel targets, which can be accurately modeled and uncovered by resorting to the wealth of spectral information provided by hyperspectral image data. Spectral unmixing based preprocessing makes reconstruction to retain this spectral information.

4.1. PROPOSED METHOD

The proposed algorithm is a two stage algorithm with spectral unmixing to reduce spectral redundancy and Linear predictive coding to reduce spatial redundancy. The method of decomposing hyperspectral image into fractional abundance image is the basic idea of this approach. These fractional abundance images are then compressed using lossless predictive coding. The proposed compression technique which relies on the concept of spectral unmixing is one of the renowned approaches to deal with mixed pixels and sub pixel targets in hyperspectral image analysis. The spectral unmixing is a two-stage approach in which the purest pixels in the image (endmembers) are first extracted and then used to express mixed pixels as linear combinations of endmembers.

Antonio Plaza et al in his work [6] proposed compression scheme using pixel purity index (PPI) as endmember extraction procedure. In this work PPI method is replaced with VCA and performance of algorithm and degradation of reconstructed image is studied.

The proposed compression algorithm proceeds as follows

1. End member extraction by Vertex Component Analysis (VCA)
2. Linear spectral unmixing constructs fractional abundance images.
3. Lossless predictive coding is applied to fractional abundance images.

End members are extracted from raw image using Vertex Component Analysis. With ‘E’ end members, ‘E’ fractional abundance image is got by spectral unmixing.No of endmembers E must be less than no of spectral bands so that spectral compression is achievable. Flow of the project work is given in Fig 4.1.

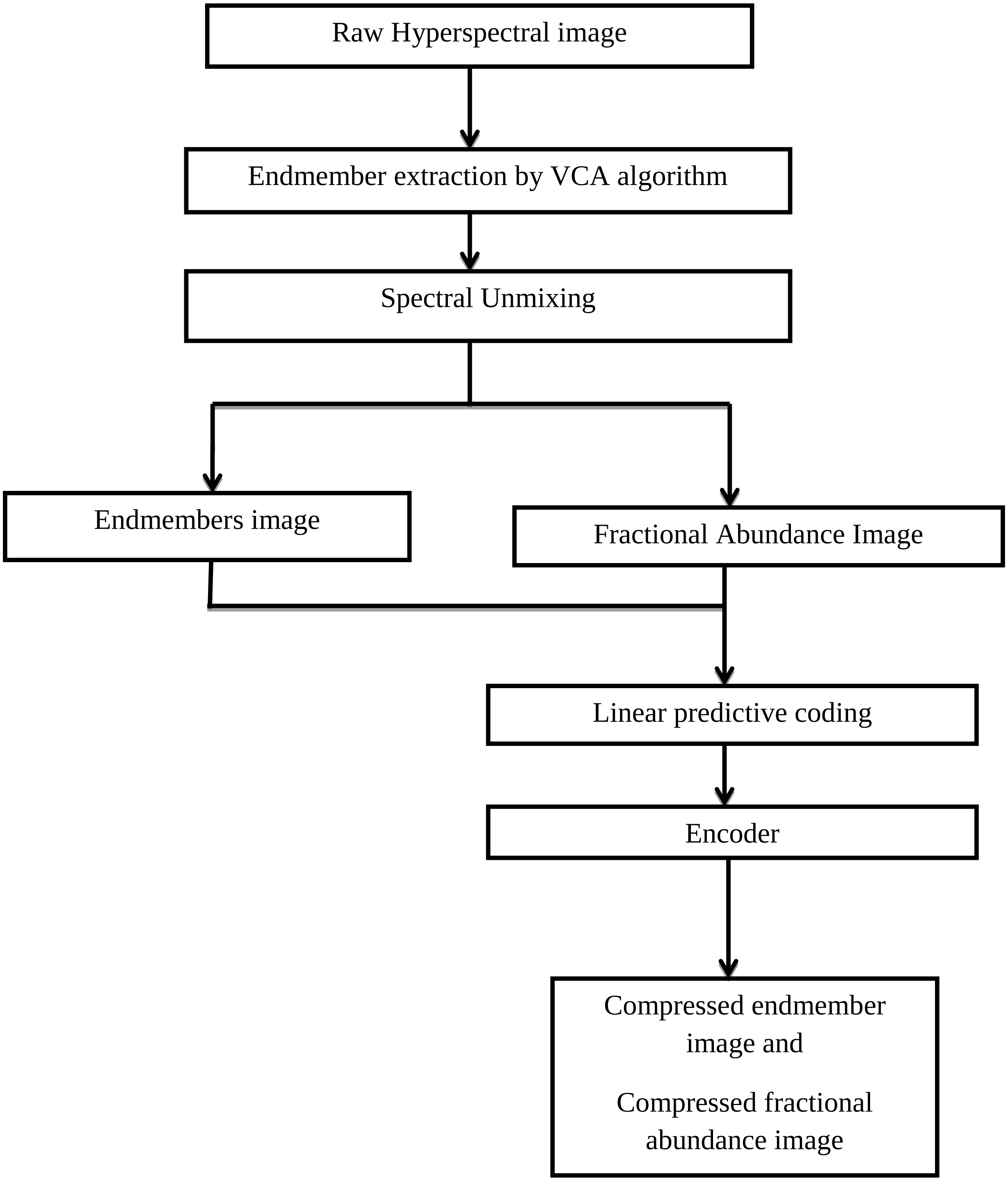

Raw Hyperspectral image
Endmember extraction by VCA algorithm
Spectral Unmixing
Endmembers image
Fractional Abundance Image
Linear predictive coding
Encoder
Compressed endmember image and
Compressed fractional abundance image

Fig. 4.1. Proposed method.

4.2. VERTEX COMPONENT ANALYSIS

The VCA is an autonomous method to unmix linear mixtures of hyperspectral datasets and is based on the geometry of convex sets. By this algorithm endmembers are extracted based on two concepts viz.

- The endmembers are the vertices of a simplex
- The affine transformation of a simplex is also a simplex.

Algorithm 1: Vertex Component Analysis (VCA)

```
INPUT $p,\ R = [r1, r2, ...., rN]$
$SNR_{th} = 15 + 10\log 10(p) dB$
if $SNR > SNR_{th}$ then
    $d := p;;$
    $X := U_d^T R;$ {Ud obtained by SVD}
    $u := mean(X);$ {u is a vector}
    $[Y]_{:,j} := [X]_{:,j} / ([X]_{:,j}^T u);$ {Projective projection}
else
    $d = p - 1;$
    $[X]_{:,j} := U_d^T (|R|_{:,j} - \ddot{r});$ {Ud obtained by PCA}
    $c := \arg\max_{j=1....N} \| [X]_{:,j}^T \|;$
    $c := [c | c | ... | c];$ {c is a 1XN vector}.
```

```
$Y := \begin{bmatrix} X \\ C \end{bmatrix}$
end if
$A := [e_u \,|\, 0 \,|\, .... \,|\, 0 \,|]; \{e_u = [0....0.1]^T$ A is a *pXp* auxiliary matrix}
for i := 1 to p do
$\mathrm{w} := \mathrm{randn}(0, \mathrm{I}_\mathrm{p}); \{$w is a zero-mean random Gaussian Vector of covariance $\mathrm{I}_\mathrm{p}\}$
$f := \{\{I - AA^{\#})w/(\| I - AA^{\#})w\|);$ {f is a vector orthonormal to the subspace spanned by [A]:;1:i.}
$v := f^T Y;$
$k := \arg\max_{j=1....N} \| [v]^T_{:,j} \|;$ {Find the projection extreme}
$|A|_{:,i} := |Y|_{:,k}$
$[indices]_i := k;$ {Stores the pixel index.}
end for
if SNR > $\mathrm{SNR}_{\mathrm{th}}$ then
$\hat{M} := U_d \,|X|_{:,indices}; \{\hat{M}$ is an $L$ $X$ $p$ estimated mixing matrix}
Else
$\hat{M} := U_d \,|X|_{:,indices} + \ddot{r};$
end if
```

VCA can be done with and without dimensionality reduction. VCA algorithm outperforms PPI in terms of computational complexity as well as performance. VCA has low computational complexity than PPI as well as NFINDR [8], it is evident from approximated no of flops used by these algorithms [VCA: $=2p^2N$; *PPI:* $=2psN$; *N-FINDR:* $=p^{\eta+1}N$] that VCA is less complex than others. It is feasible to implement VCA based method in hardware with near real time performance.

4.3. LINEAR SPECTRAL UNMIXING

Linear spectral unmixing is a commonly accepted approach to mixed-pixel classification in hyperspectral imagery. This approach involves two steps. First, to find spectrally unique signatures of pure ground components, usually known as end members and second, to express mixed pixels as linear combinations of end member materials. Fig 4.2 illustrates this process

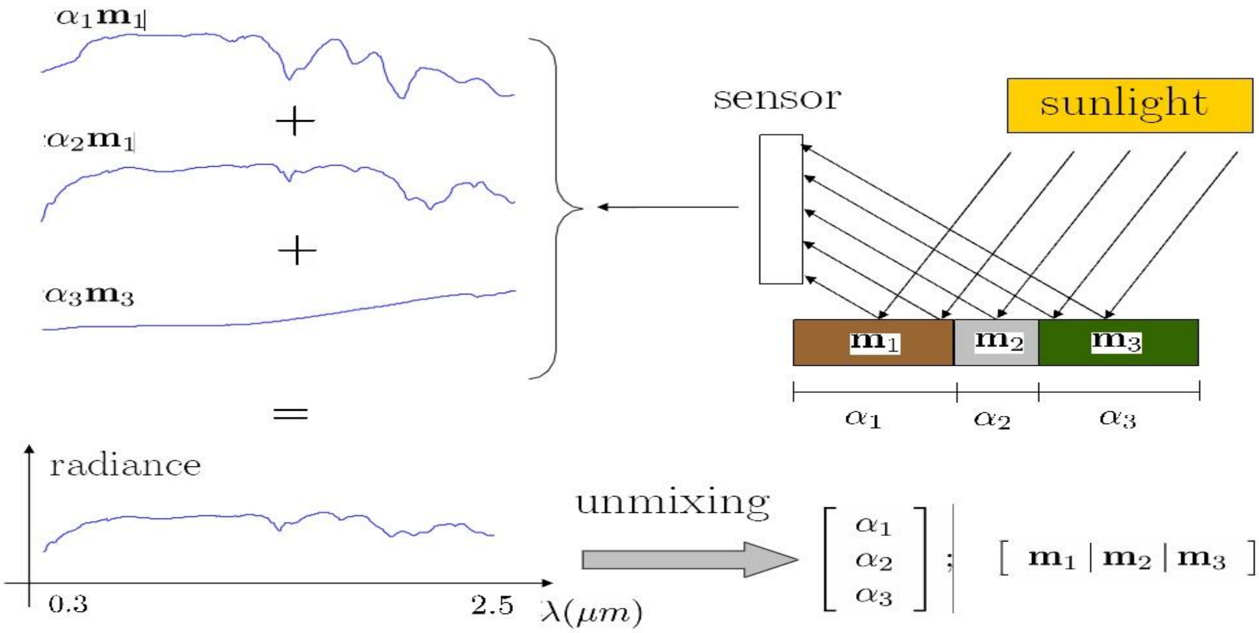


Fig. 4.2. Systematic view of spectral mixing/unmixing process

For each sample pixel vector **f***i* in **F**, a set of abundance fractions specified by **a*i* = {*ai*1, *ai*2,……, *aiE*}** is obtained using the set of end members ***{ee} Ee* =1**, so that fi can be expressed as a linear combination of end members as follows

$$fi = e1.ai1 + e2.ai2 + ... + eE.aiE \qquad (4.1)$$

In the Figure 4.2 [α1, α2, α3] are abundance images and [m1, m2, m3] are endmembers

4.4. LOSSLESS PREDICTIVE CODING

Lossless predictive coding predicts the value of each pixel by using the values of $\hat{x}$ its neighboring pixels. Therefore, every pixel is encoded with a prediction error rather than its original value. Typically, the errors are much smaller compared with the original value so that fewer bits are required to store them. Figure 4.3 represents lossless predictive coding scheme

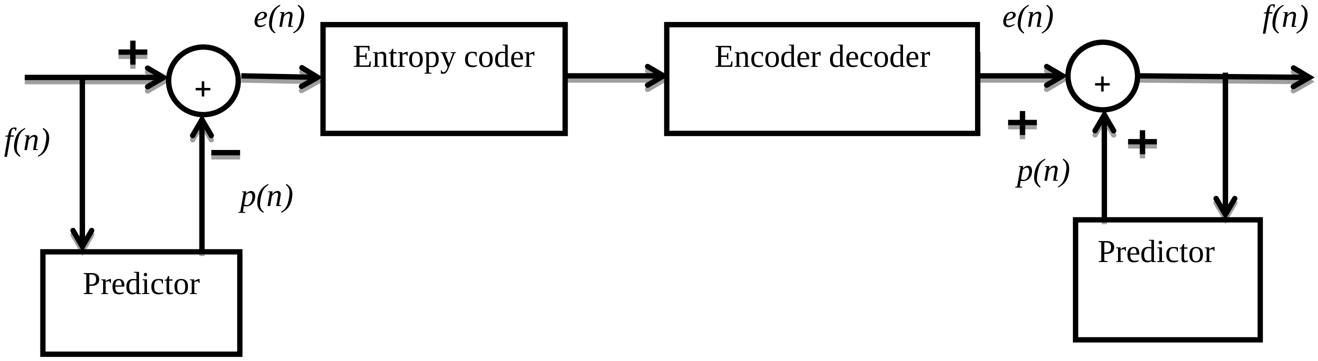


Fig.4.3. Lossless Predictive coding

The predictor error $e(n) = x(n) - f(n)$ where the predictor $p(n) = round\left[\sum_{i=1}^{m} \alpha_i f(n-i)\right]$; where m is order of linear operations and α_i denote the prediction coefficient. For an instance Consider m=1 and =1 i.e. 1-D predictor is used. Fractional abundance image of each end member is compressed spatial with this predictor. In spatial domain x and y being rows and column the 1-D prediction is given by $p(x,y) = round\left[\sum_{i=1}^{m} \alpha f(x, y-i)\right]$ And error $e(x,y) = p(x,y) - f(x,y)$.The error has to be encoded for which arithmetic coding is used.

4.5. ENCODER: ARITHMETIC CODING

Arithmetic coding is a data compression technique that encodes data by creating a code which represents a fraction in the unit interval [0, 1]. The algorithm is recursive. On each recursion, the algorithm successively partitions subintervals of the unit interval [0, 1]. This differs from other forms of entropy encoding such as Huffman coding, which separates the input into component symbols and replacing each with a code.

Huffman coding, which is the most famous method for source coding or data compression, guarantees a coding rate R within 1 bit of the entropy H. That is, the Huffman code for a source S with an average code word length L satisfies

$$H(S) \leq L \leq H(S) + 1 \qquad (4.2)$$

Moreover, if we encode the output of the source in longer blocks of symbols, we are guaranteed the average code word length per input symbol closer to the entropy. In other words, suppose we encode the sequence by generating a code word for every n symbol, and then we have

$$H(S) \leq L_n \leq H(S) + \frac{1}{n} \qquad (4.3)$$

Where L_n denotes the average code word length per input symbol.

However, there is still another problem. The latter approach becomes impractical since it causes an exponential growth in the size of the codebook when we try to obtain

Huffman codes for long sequences of symbols. In other words, the complexity of this approach increases exponentially with block length.

Arithmetic coding is a method of encoding without this inefficiency. In arithmetic coding, instead of using a sequence of bits to represent a symbol, we represent it by a subinterval of the unit interval$[0, 1]$. In other words, we encode the data into a number in the unit interval$[0, 1]$, and this technique can be implemented by separating the unit interval into several segments according to the number of distinct symbols. The length of each segment is proportional to the probability of each symbol, and then the output data is located in the corresponding segment according to the input symbol.

This provides a way of assigning code words to particular sequences without having to generate code words for all sequences and alleviates the inefficiency and the complexity. Moreover, the code for a sequence of symbols is an interval whose length decreases as we add more symbols to the sequence. This property allows us to have a coding scheme that is incremental, that is, the code for an extension to a sequence can be calculated simply from the code for the original sequence.

In order to distinguish a sequence of symbols from another sequence of symbols, we need to tag it with a unique identifier. A possible set of tags for representing sequences of symbols is the numbers in the unit interval [0, 1]. Because the number of numbers in the unit interval is infinite, it is able to assign a unique tag to each distinct sequence of symbols. In order to do this we need a function that will map sequences of symbols into the unit interval. Apparently, a possible function that maps random variables and sequences of random variables, into the unit interval is the cumulative distribution function (cdf) of the random variable associated with the source. This is the function we will use in developing the arithmetic code.

In arithmetic coding a source ensemble is represented by an interval between 0 and 1 on the real number line. Each symbol of the ensemble narrows this interval. As the interval becomes smaller, the number of bits needed to specify it grows. Arithmetic coding assumes an explicit probabilistic model of the source. It is a defined-word scheme which uses the probabilities of the source messages to successively narrow the interval used to represent the ensemble. A high probability message narrows the interval less than a low probability message, so that high probability messages contribute fewer bits to the coded ensemble. The method begins with an unordered list of source messages and their probabilities. The number line is partitioned into subintervals based on cumulative probabilities.

A small example shown in table 4.1 will be used to illustrate the idea of arithmetic coding. Given source messages {*A*, *B*, *C*, *D*, *#*} with probabilities {0.2, 0.4, 0.1, 0.2, 0.1}. The symbol *A* corresponds to the first 1/5 of the interval [0, 1); *B* the next 2/5; *D* the subinterval of size 1/5 which begins 70% of the way from the left endpoint to the right. When encoding begins, the source ensemble is represented by the entire interval [0, 1). For the ensemble *AADB#*, the first *A* reduces the interval to [0, 0.2) and the second *A* to [0, 0.04) (the first 1/5 of the previous interval). The *D* further narrows the interval to [0.028, 0.036) (1/5 of the previous size, beginning 70% of the distance from left to right). The *B* narrows the interval to [0.0296, 0.0328), and the *#* yields a final interval of [0.03248, 0.0328). The interval, or alternatively any number *i* within the interval, may now be used to represent the source ensemble.

Two equations may be used to define the narrowing process described above:

$$newleft = \mathrm{prevleft} + \mathrm{msgleft} * \mathrm{prevsize} \qquad (4.4)$$

$$newsize = \mathrm{msgsize} * \mathrm{prevsize} \qquad (4.5)$$

Table 4.1 -- The Arithmetic coding model.

Source message	Probability	Cumulative Probability	Range message probability
A	0.2	0.2	[0,0.2)
B	0.4	0.6	[0.2,0.6)
C	0.1	0.7	[0.6,0.7)
D	0.2	0.9	[0.7,0.9)
#	0.1	1.0	[0.9,1.0)

The first equation states that the left endpoint of the new interval is calculated from the previous interval and the current source message. The left endpoint of the range associated with the current message specifies what percent of the previous interval to remove from the left in order to form the new interval. For *D* in the above example, the new left endpoint is moved over by 0.7 * 0.04 (70% of the size of the previous interval). The second equation computes the size of the new interval from the previous interval size and the probability of the current message (which is equivalent to the size of its associated range). Thus, the size of the interval determined by *D* is 0.04 * 0.2, and the right endpoint is 0.028+0.008=0.036 (left endpoint + size).

The size of the final subinterval determines the number of bits needed to specify a number in that range. The number of bits needed to specify a subinterval of [0, 1) of size *s* is -log *s*. Since the size of the final subinterval is the product of the probabilities of the source messages in the ensemble (that is, $s = -PROD_{i=1}^{N} p(source..message.i)$, where *N* is the length of the ensemble, we have $-\log s = -SUM_{i=1}^{N} \log p(source..message.i) = -SUM_{i=1}^{N} p(a(i)\log(p(a(i))$, where *n* is the number of unique source messages *a* (1), *a* (2), ... *a* (*n*). Thus, the number of bits generated by the arithmetic coding technique is exactly equal to entropy, *H*. This demonstrates the fact that arithmetic coding achieves compression which is almost exactly that predicted by the entropy of the source.

In order to recover the original ensemble, the decoder must know the model of the source used by the encoder (eg. the source messages and associated ranges) and a single number within the interval determined by the encoder. Decoding consists of a series of comparisons of the number *i* to the ranges representing the source messages. For this example, *i* might be 0.0325 (0.03248, 0.0326, or 0.0327 would all do just as well). The decoder uses *i* to simulate the actions of the encoder. Since *i* lies between 0 and 0.2, he deduces that the first letter was *A* (since the range [0, 0.2) corresponds to source message *A*). This narrows the interval to [0, 0.2). The decoder can now deduce that the next message will further narrow the interval in one of the following ways: to [0, 0.04) for *A*, to [0.04, 0.12) for *B*, to [0.12, 0.14) for *C*, to [0.14, 0.18) for *D*, and to [0.18, 0.2) for *#*. Since *i* falls into the interval [0, 0.04), he knows that the second message is again *A*. This process continues until the entire ensemble has been recovered.

Thus the proposed methodology uses VCA for endmembers extraction. With the endmembers extracted fractional abundance matrix is estimated by linear spectral unmixing. Fractional abundance matrix of each endmember is then checked for spatial redundancy with help of Lossless predictive coding and generated residuals are encoded with arithmetic coding. This chapter has well explained each stages of the methodology.

CHAPTER 5

FPGA ARCHITECTURE

5.1 GENERAL ARCHITECTURE

Input memory block (Fig 4.1) is a dual port memory (internal embedded RAM of Xilinx devices can be used). Y is a matrix with pixel vectors of hyperspectral image as columns. A_generator generates ‘A’ auxiliary matrix (*p x p*) and pseudo random vector generator is used to generate vector ‘w’. f_generator generates vector ‘f’ and projection of ‘f’ on matrix Y is done in the block image projection as given in the algorithm in Section II. Then the index of the projection extreme is sent back to memory for endmember selection and final endmembers of VCA is given by Y (indices).

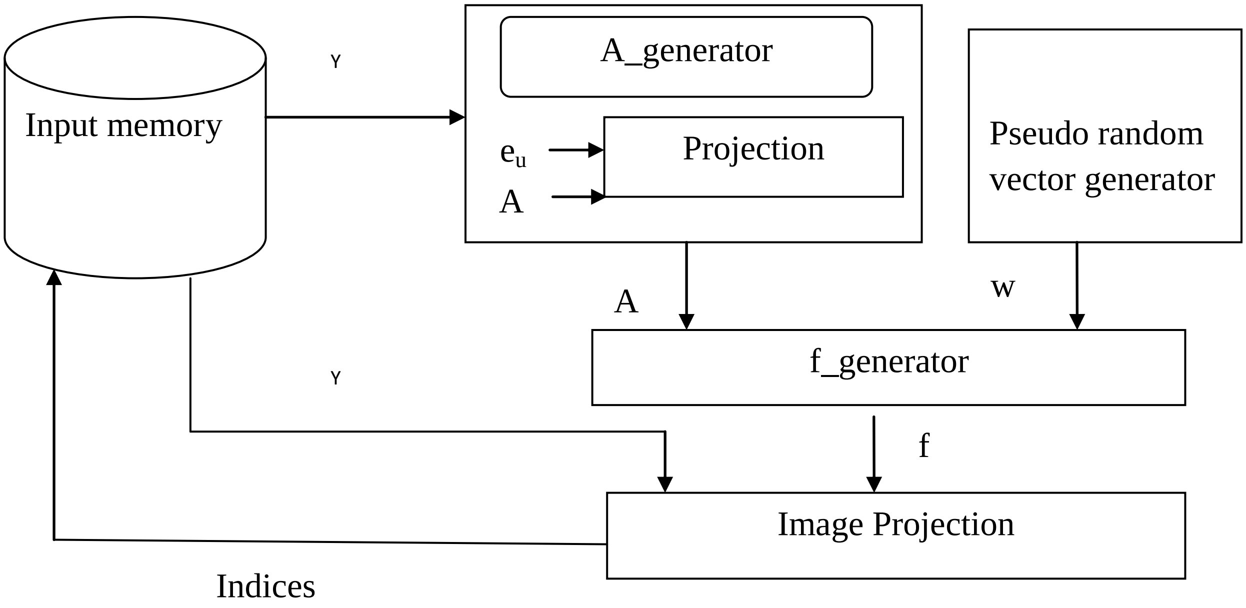


Fig 5.1 Architecture for VCA.

The computational intensive part of VCA is the the matrix-vector product f^TY in image projection block. To alleviate the computational burden it is desirable to implement it in parallel. It is worth noting that each pixel projection can be done independently from the other pixels projections the proposed parallel implementation [22], where the dataset Y is divided into blocks, and each block has 32 threads. Since VCA computational complexity is proportional to the number of pixels of the dataset, one way to reduce the computational burden is to reduce the number of pixels to be projected on each iteration. It is found that on each iteration vectors that belong to the subspace spanned by the endmembers found so far have null projection onto the new direction, thus they can be removed since they are not candidates to be endmembers.

5.2 SYSTOLIC ARRAY ARCHITECTURE FOR IMAGE PROJECTION

A systolic architecture is an arrangement of processors i.e. PEs in an array where data flows synchronously across the array between neighbors, usually with different data flowing in different directions. PE at each step takes input data from one or more neighbors (e.g. Left and Top), processes it and, in the next step, outputs results in the opposite direction (Right and Bottom). The architecture of PE is shown in figure 5.2

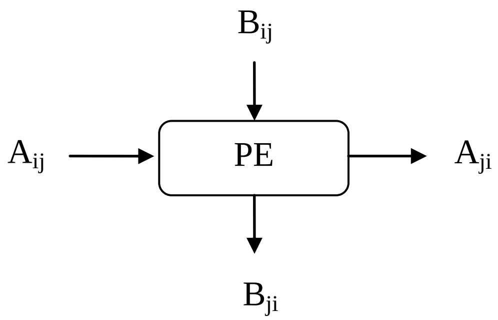


Fig 5.2 Architecture for Processing Element.

A, B and C are the matrices with order *m* x *k, k* x *n* and *m* x *n* respectively. Each PE of systolic array computes the multiplication of elements and accumulates to the corresponding element and then elements will be passed to neighbor PE in the systolic array. First elements a_{ij} in row i of matrix A are injected first into PE as pipeline with the sequence of $a_{j,k}$ and the input time $a_{i+1,j}$ to the element of is one time unit later than $a_{j,k}$. Similarly, elements $b_{i,j}$ in column j of matrix B are injected first into PE as pipeline with the sequence of $b_{k,j}$ and the input time to the element of the sequence of $b_{k,j+1}$ is one time unit later than $b_{k,}$j.

The array architecture given above takes input data in parallel into first PEs in the array and processes the Multiplication and Accumulation on them and then outputs result to the next level PEs of array. Systolic arrays do not lost their speed due to their connection like any other parallelism. Where, each cell (PE) is an independent Processor (CPU) and has its own registers and Arithmetic and Logic Units (ALUs) i.e. Multiplication and Accumulation unit. The cells share the information with their neighbors, after performing the necessary operations on the data. Systolic Array Architecture (SAA) for Matrix Multiplication is shown in the Figure 5.3.

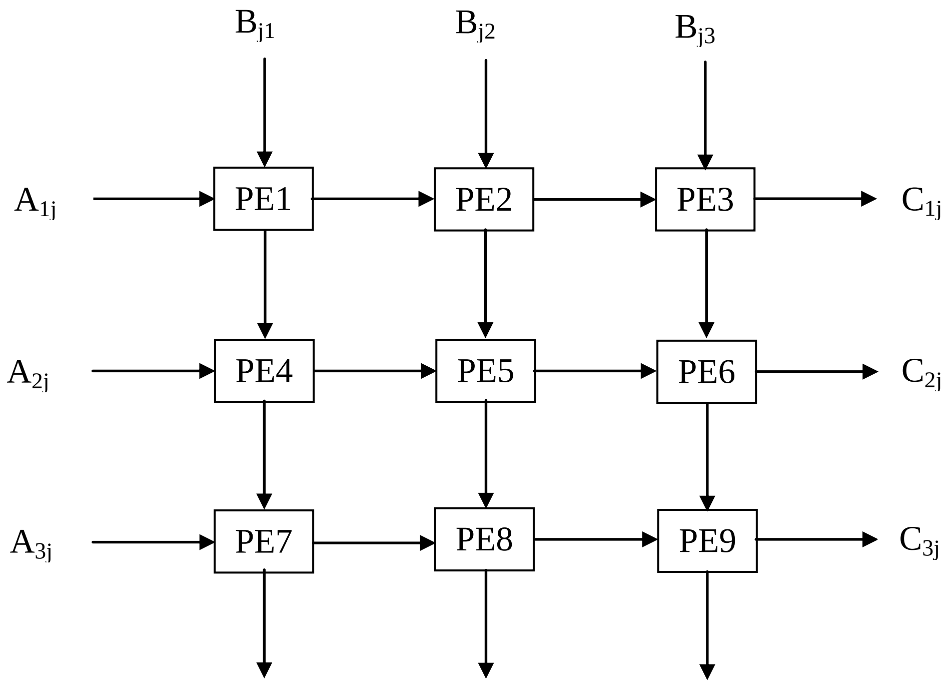


Fig 5.3 Architecture for Systolic Array

In Fig 5.3 each cell takes inputs from left and top, multiplies them and accumulates in the local register which is inside the each PE. After N^2 clock pulses the result would be stored in each PE. The proposed systolic array architecture needs N^2 magnitude Multipliers, 2N magnitude Accumulators and 4N registers are needed to compute matrix multiplication where N is order of matrix. This architecture is adopted for multiplication of each block of the matrix-vector product f^T Y.

This chapter has explained methods to reduce the computational burden of VCA by pixel reduction. This chapter has also suggested systolic array based architecture which is computationally efficient, parallel and well pipelined for implementation of spectral compression scheme proposed.

CHAPTER 6

RESULTS AND DISCUSSI0NS

Experimental results provided in this chapter shows images extracted from various band of a hyperspectral image and comparison of reconstructed images from 242, 200, 100 and 50 end members extracted by PPI and VCA and its corresponding abundance fraction.

Tool used: MATLAB R2012b

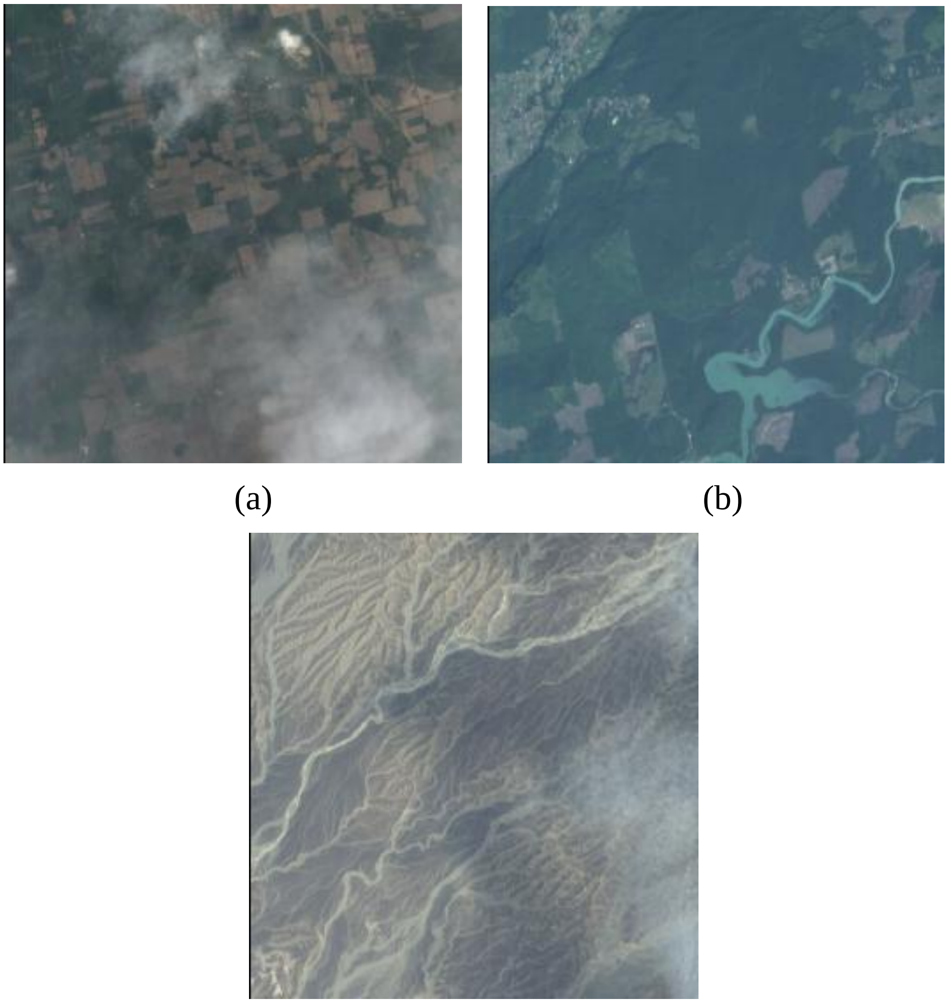
Raw images used: LakeMonana.bip, ErtaAle.bip, Mt.St.Helens.bip

Raw images are Hyperion Level 0 (raw, uncalibrated) image provided by the EO-1 Mission, NASA/USGS. Each image has width 256 cross-track samples, 242 spectral channels, and the height indicated in the table below. The Hyperion imager produces 12-bit data samples. In the files here, each sample is stored as a 2-byte unsigned integer in little-endian byte order, samples arranged in BIP order. A false-color browse image is included with each image.

Image Dimensions	**Abbreviation**	**Alternative Name**
M-N-3	BSQ	Band-Sequential or Band-Interleaved
3-M-N	BIP	Band-Interleaved-by-Pixel or Pixel-Interleaved
M-3-N	BIL	Band-Interleaved-by-Line or Row-Interleaved

BSQ, BIL, and BIP represent ways of storing 24-bit images in memory or on disk. The initials stand for band-sequential, band-interleaved-by-line, and band-interleaved-by-pixel, respectively. These image formats are also sometimes called "band-interleaved", "row-interleaved", and "pixel-interleaved", respectively. Images are stored in one format or another to ease image manipulations. For example, the BSQ (band-sequential) format is optimal for accessing the image spatial information (XY) or color band information. The BIP (band-interleaved-by-pixel) format is optimal for accessing the image spectral (Z) information. The BIL (band-interleaved-by-line) format is a compromise format, for the most part, allowing fairly easy access to both spatial and spectral information.

With Matlab programming 256x256x242 block of Hyperion images viz. LakeMonana.bip, ErtaAle.bip, Mt.St.Helens.bip [Fig. 6.1] are analyzed with both VCA and PPI based spectral unmixing.

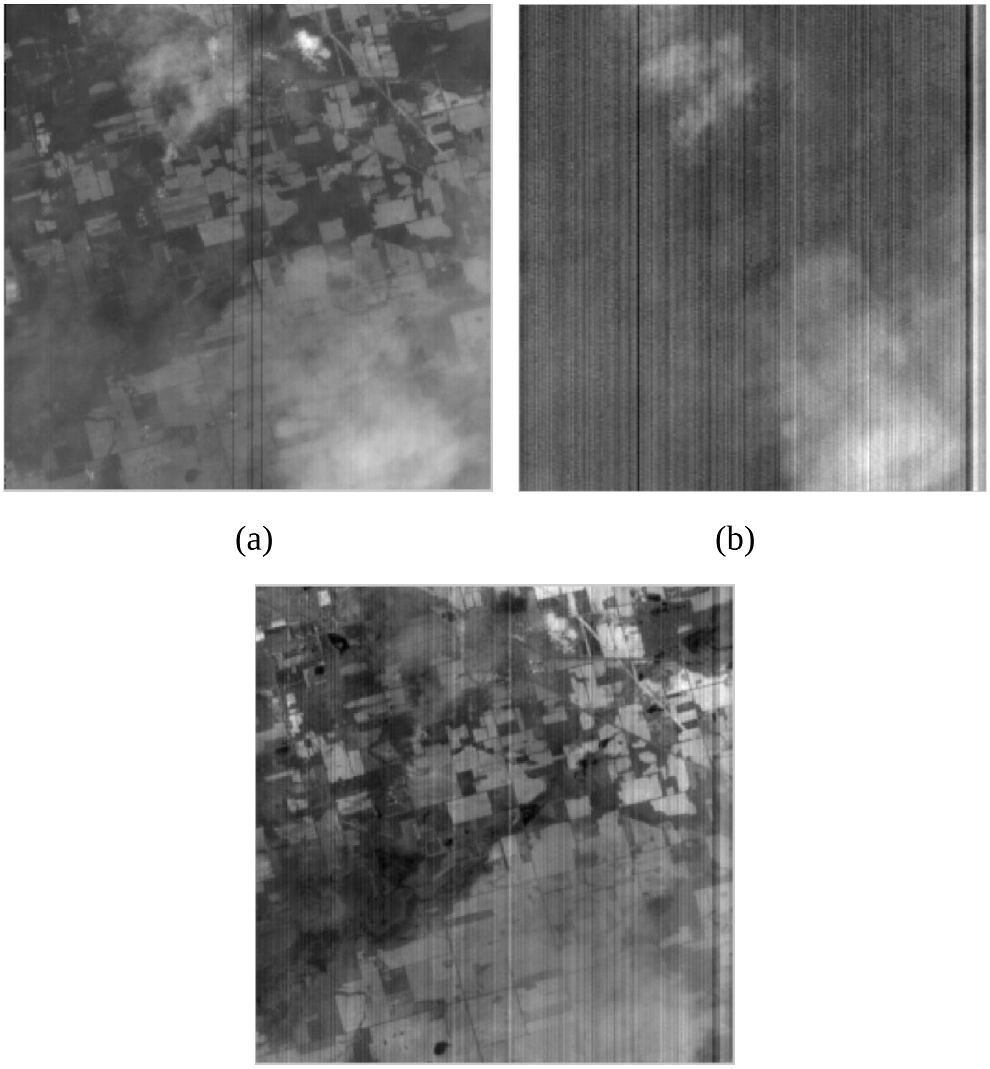


(a) (b)

(c)

Fig. 6.1. a)Lakemonana b)Mt.St.Helens c)ErtaAle

Images are reconstructed from end member image and its fractional abundance image over multiplication. Some of the bands among 242 bands of Lakemonana are displayed in Fig 6.2.

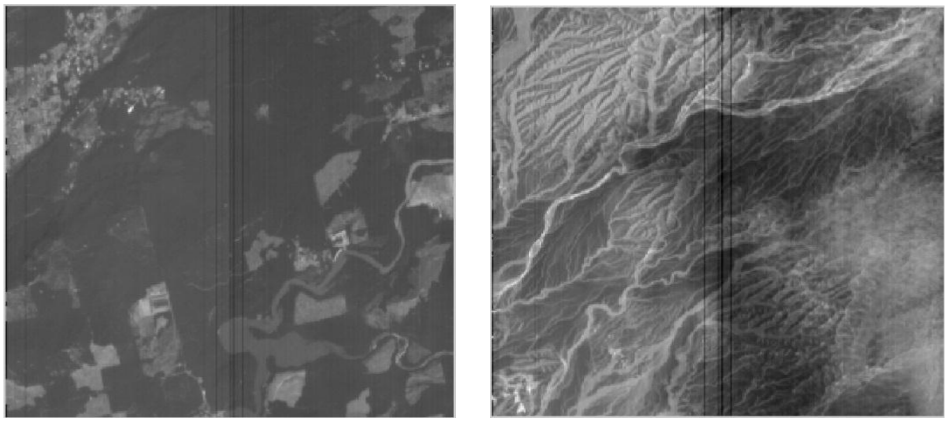


(a) (b)

(c)

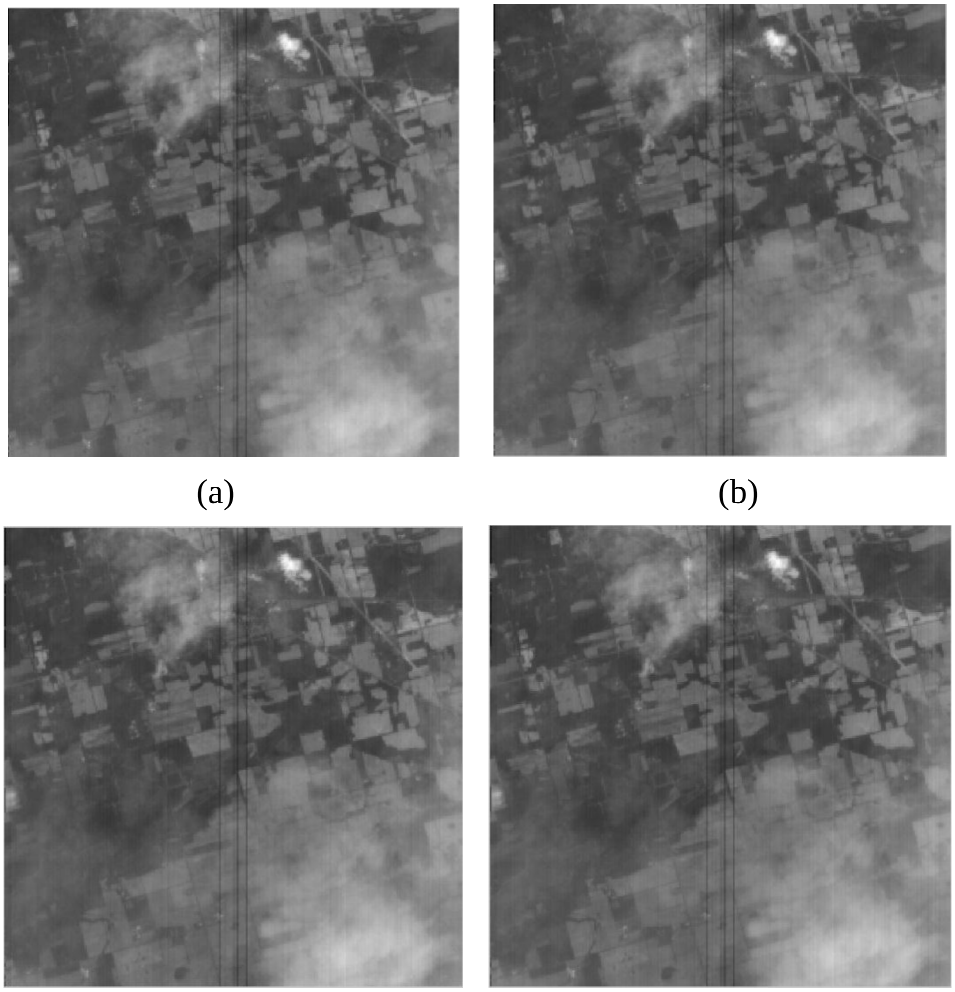
Fig. 6.2. Lakemonana a)30th band b)175th band c)210th band

30th band of images is used for analysing psychovisual variations between original image and reconstructed image after spectral unmixing. Fig 6.3 shows 30th band of other two hyperion images used. Fig 6.4 shows 30th band reconstructed lakemonana images from various PPI methods.



(a) (b)

Fig. 6.3. 30th band of a) Mt.St.Helens b) ErtaAle



(a) (b)

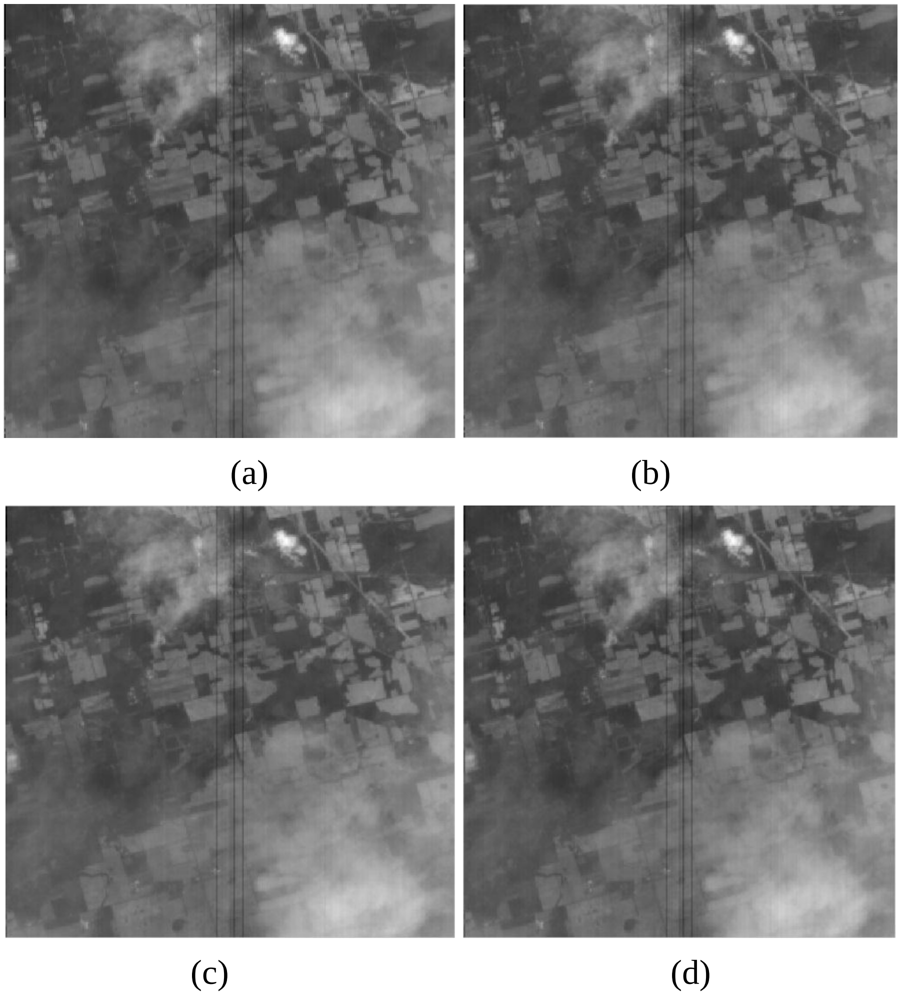
(c) (d)

Fig. 6.4.Lakemonana 30th band reconstructed a)PPI_242 b) PPI_200 c)PPI_100 d)PPI_50

Fig 6.5 shows 30th band reconstructed lakemonana images from various VCA methods.

VCA_242: Reconstructed image from 242 endmembers extracted by VCA and fractional abundance matrix.

VCA_200: Reconstructed image from 200 endmembers extracted by VCA and fractional abundance matrix.


(a) (b)

(c) (d)

Fig. 6.5. Lakemonana 30th band reconstructed a)VCA_242 b)VCA_200 c)VCA_100 d)VCA_50.

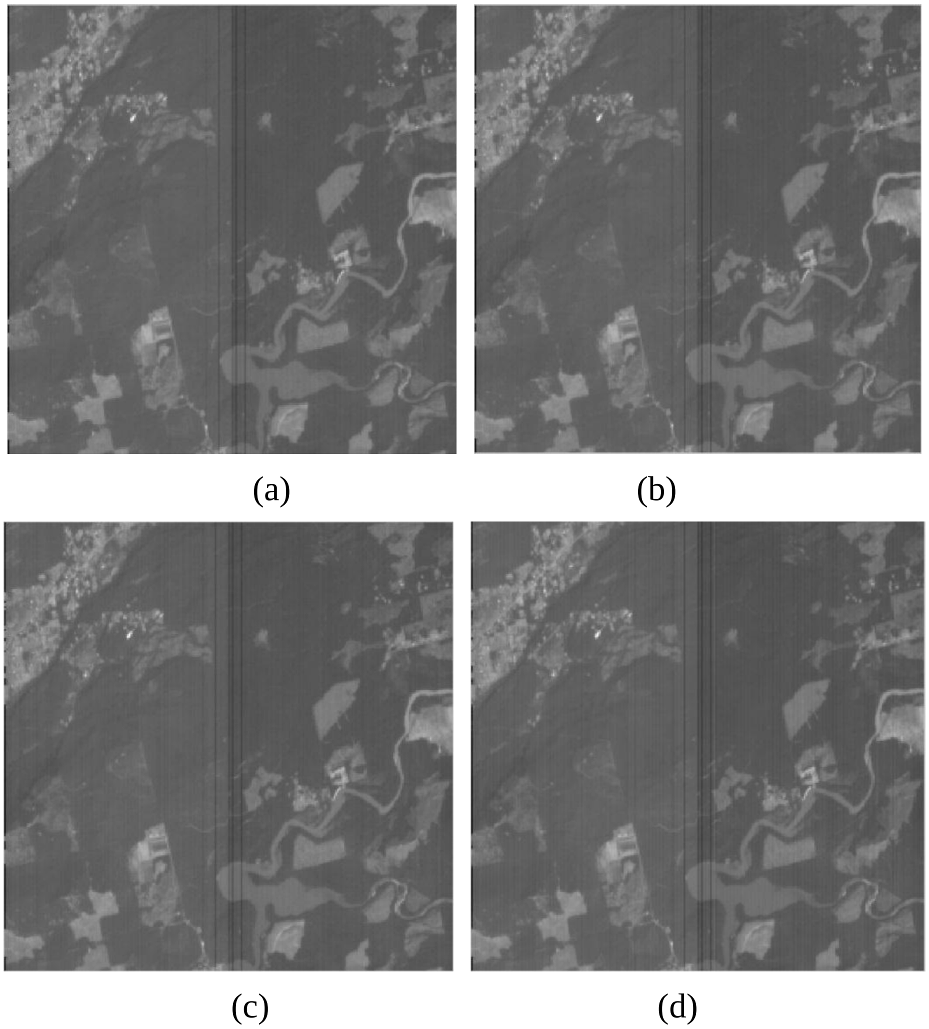
VCA_100: Reconstructed image from 100 endmembers extracted by VCA and fractional abundance matrix.

VCA_50: Reconstructed image from 50 endmembers extracted by VCA and fractional abundance matrix.

Fig 6.6 shows 30th band reconstructed Mt.St.Helens images from various PPI methods.

PPI_242: Reconstructed image from 242 endmembers extracted by PPI and fractional abundance matrix.

PPI_200: Reconstructed image from 200 endmembers extracted by PPI and fractional abundance matrix.



(a) (b)

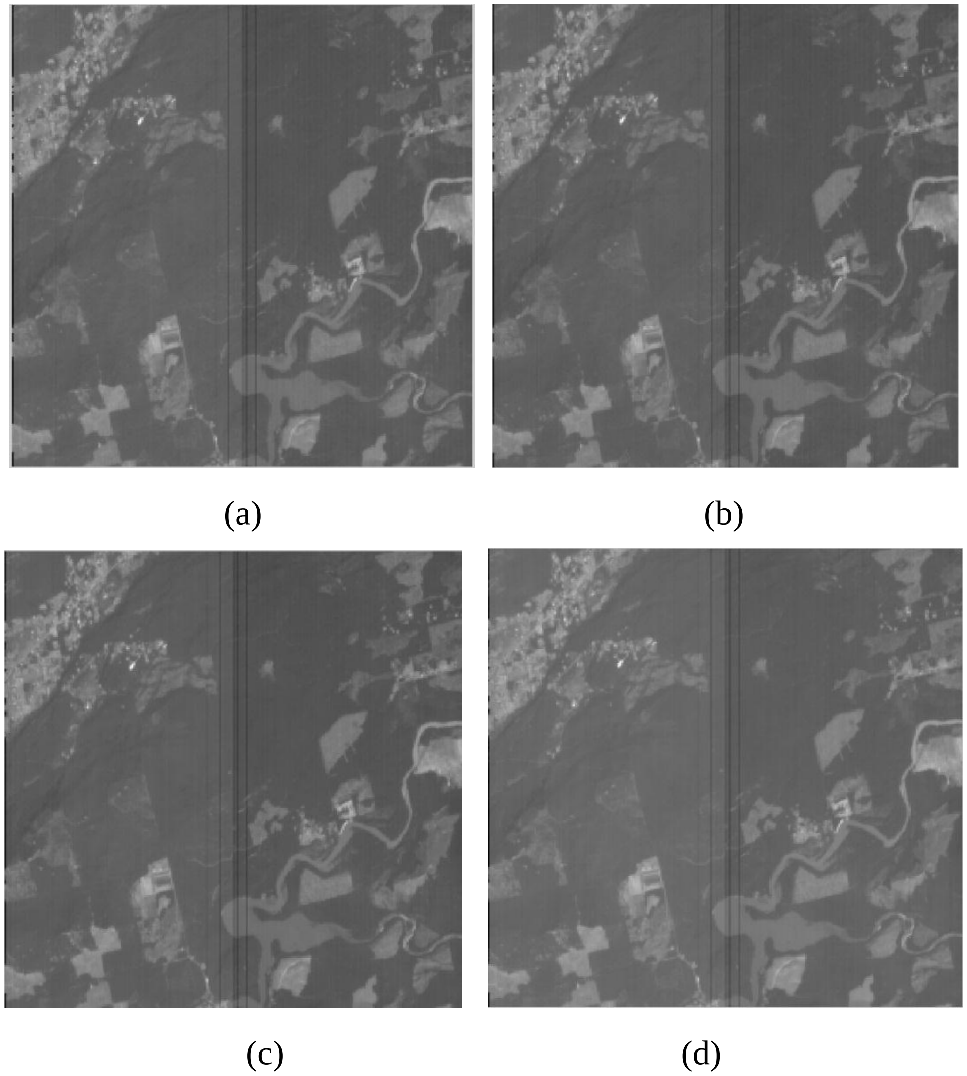
(c) (d)

Fig. 6.6.Reconstructed 30th band of Mt.St.Helens a)PPI_242 b)PPI_200 c)PPI_ 100 d)PPI_50.

PPI_100: Reconstructed image from 100 endmembers extracted by PPI and fractional abundance matrix.

PPI_50: Reconstructed image from 50 endmembers extracted by PPI and fractional abundance matrix.

Fig 6.7 shows 30th band reconstructed Mt.St.Helens images from various VCA methods.



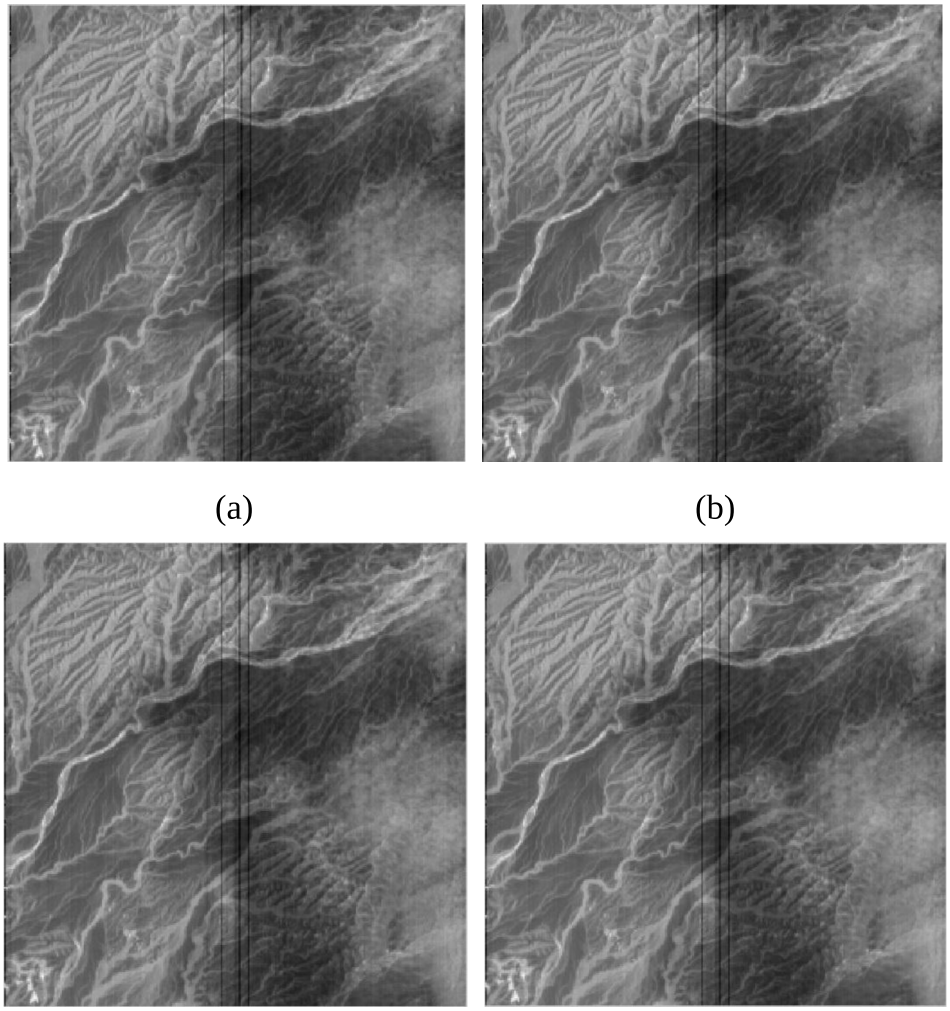
(a) (b)

(c) (d)

Fig. 6.7.Reconstructed 30th band of Mt.St.Helens a)VCA_242 b)VCA _200 c)VCA_100 d)VCA_50.

The Figures 6.4 to Fig 6.9 show 30th bnd of reconstructed images from various endmembers extracted from pixel purity index (PPI) algorithm and Vertex component analysis (VCA) algorithms and its corresponding fractional abundance matrices.Reconstructed image *Y1= E*F,* where E is endmember matrix and F is fractional abundance matrix.

Fig 6.8 shows 30th band reconstructed ErtaAle images from various VCA methods. The psycho visual comparison of reconstructed band of three types of images Lakemonana (residential area and cloud) , Mt.St.Helens (vegetation, water and mountain) and Ertale (desert sand) is analysed.

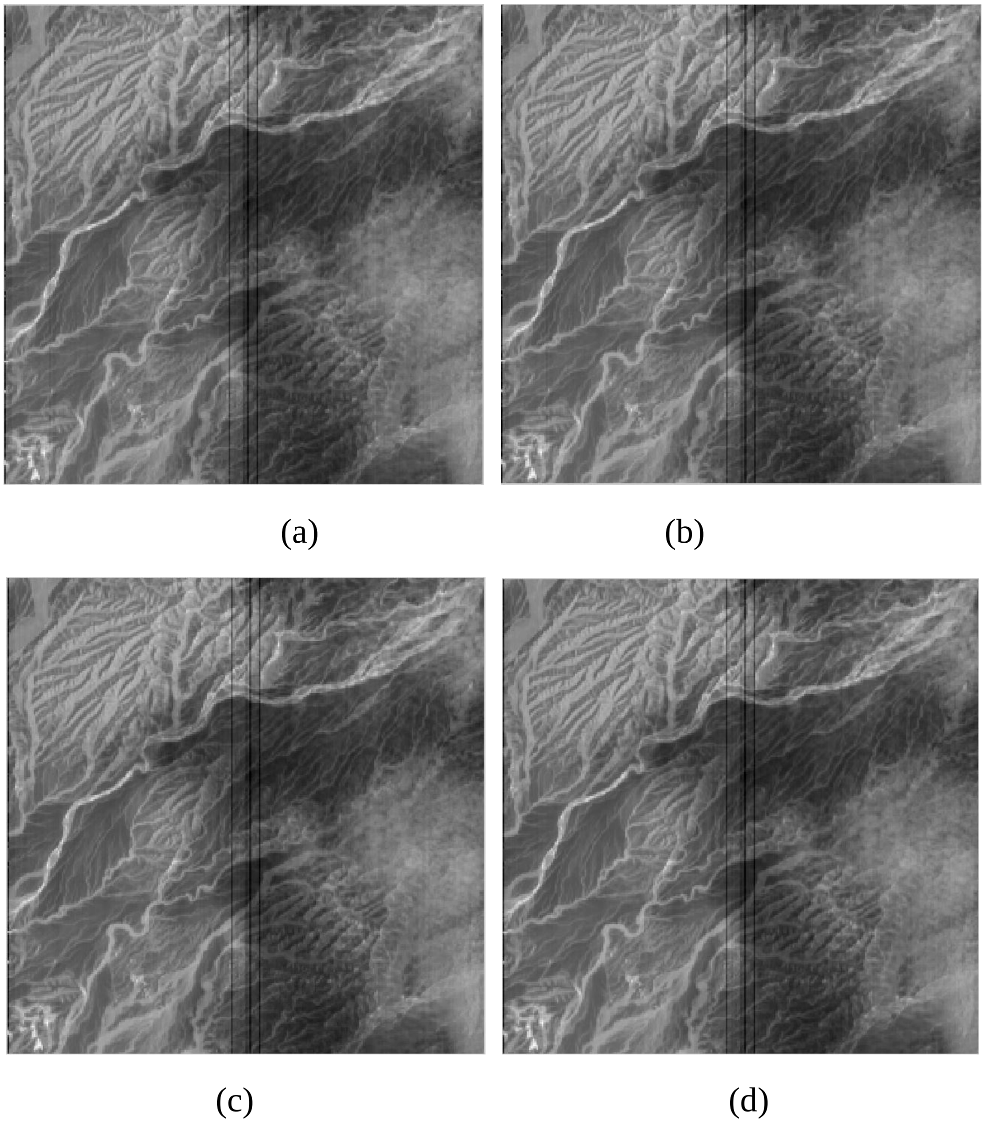


(a) (b)

(c) (d)

Fig. 6.8.Reconstructed 30th band of ErtaAle a)PPI_242 b)PPI_200 c)PPI_100 d)PPI_50.

Fig 6.9 shows 30th band reconstructed ErtaAle images from various VCA methods. The reconstructed images shows only slight differences over psychovisual perceptions thus statistical measures are used to compare the results.



(a) (b)

(c) (d)

Fig. 6.9.Reconstructed 30th band of ErtaAle a)VCA_242 b)VCA_ 200 c)VCA_100 d)VCA_50.

With fractional abundance matrix and endmember matrix, hyperspectral image can be reconstructed. The degradation of image over reconstruction can be evaluated by various quality metrics [15], [16] .The evaluated quality metrics are MAD (Maximum

Absolute Difference), MAE (Mean Absolute Error), MSE (Mean Square Error), RRMSE (Relative Root Mean Square Error).

$I(x,y,\lambda)$ denotes original image and $\tilde{I}(x,y,\lambda)$ denotes reconstructed image. The no of rows, columns and spectral bands of hyperspectral image are denoted by n_x, n_y, n_λ respectively. The error is denoted by e.

$$e = I(x,y,\lambda) - \tilde{I}(x,y,\lambda) \tag{6.1}$$

The formulae of MAD, MAE, MSE, and RRMSE are given by

$$MAD = \max_{x,y,\lambda} \{e(x,y,\lambda\} \tag{6.2}$$

$$MAE = \frac{1}{n_x n_y n_\lambda} \sum_{x,y,\lambda} |e(x,y,\lambda)| \tag{6.3}$$

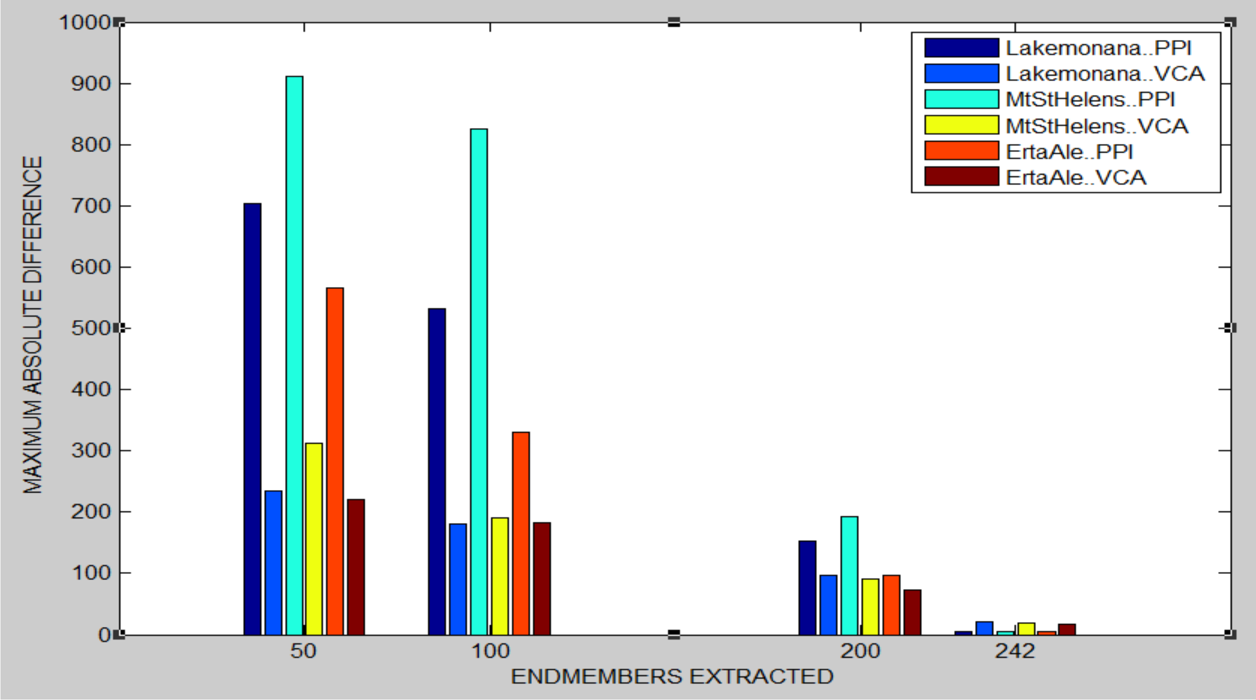


Fig. 6.10.Comparison of MAD values of reconstructed image from various no of endmembers extracted via PPI and VCA.

$$RRMSE = \sqrt{\frac{1}{n_x n_y n_\lambda} \sum_{x,y,\lambda} \left(\frac{e(x,y,\lambda)}{I(x,y,\lambda)} \right)^2} \qquad (6.4)$$

$$MSE = \frac{1}{n_x n_y n_\lambda} \sum_{x,y,\lambda} | e(x,y,\lambda) |^2 \qquad (6.5)$$

From Fig 6.10 it is clear that Maximum absolute difference of reconstructed images increases over reducing no of endmembers.comparing MAD value of PPI based endmember extraction and VCA based end member extraction,VCA based method exhibits lower MAD value than PPI based method.

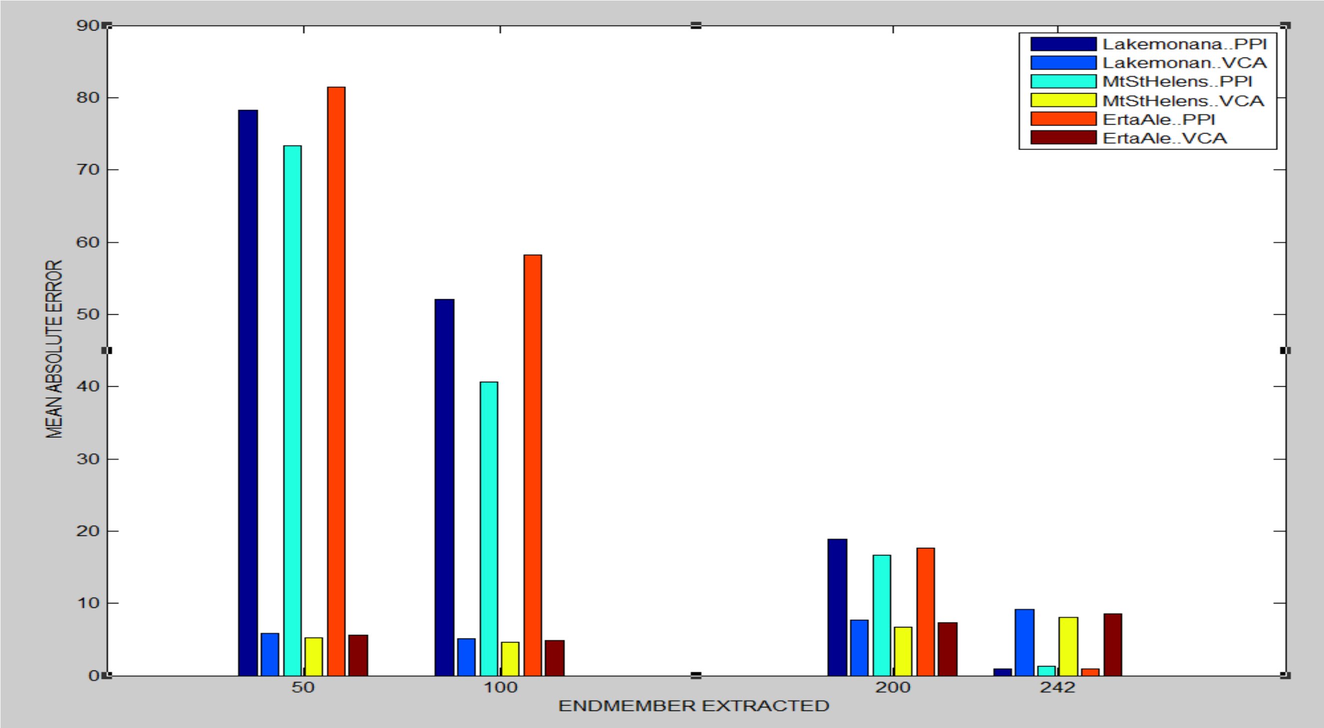


Fig. 6.11.Comparison of MAE values of reconstructed image from various no of endmembers extracted via PPI and VCA.

From Fig 6.11 it is clear that Mean Absolute Error (MAE) of reconstructed images from PPI based method increases steadily over decreasing the endmembers being extracted. MAE value of reconstructed image by VCA based method is comparitively lower than that of PPI based method.In VCA method MAE of reconstructed image from

242 endmember is lowest ,200 endmembers MAE is above 5 ,for 100 and 50 endmembers its below 5.

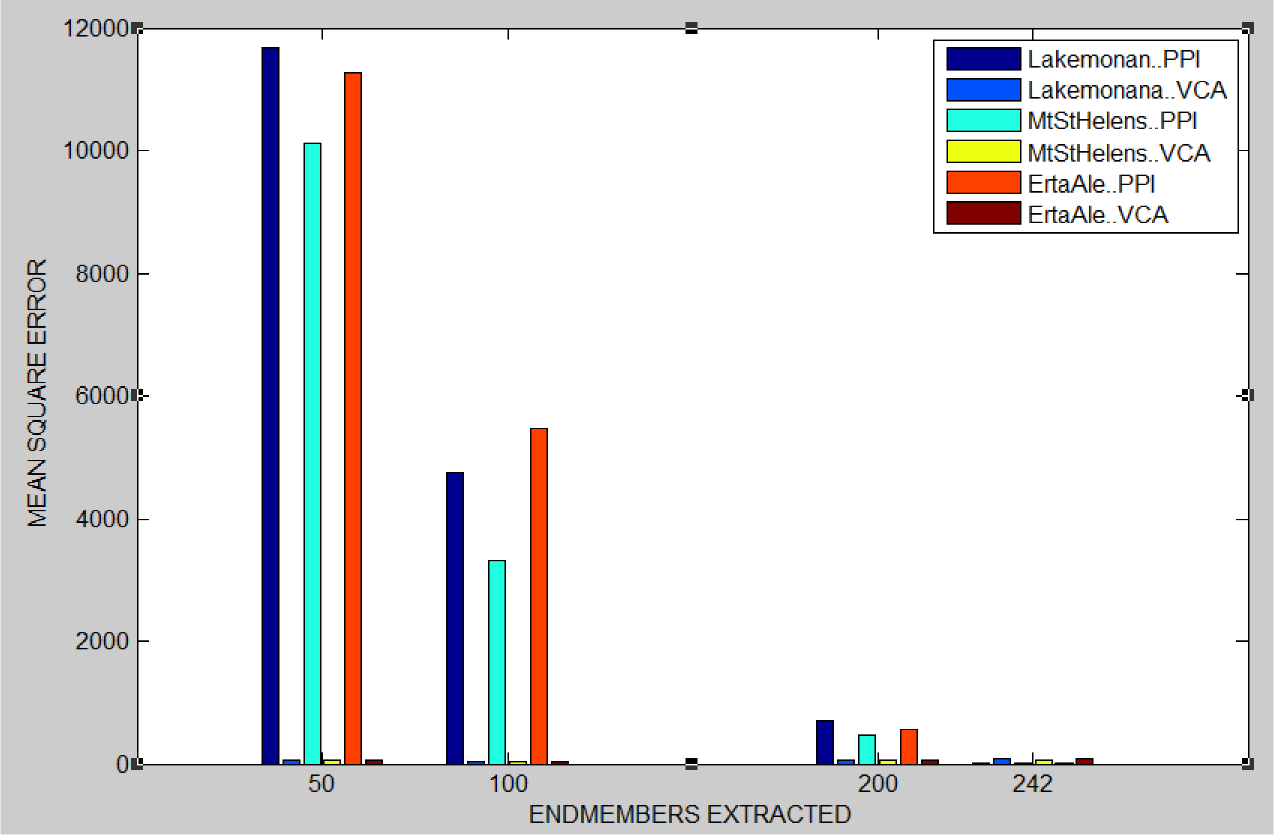


Fig. 6.12.Comparison of MSE values of reconstructed image from various no of endmembers extracted via PPI and VCA.

From Fig 6.12 & 6.13 it is clear that MSE and RRMSE values of reconstructed images from VCA based method is lower than that of PPI based method. Thus degradation over reconstruction from VCA method is lower than that of PPI method

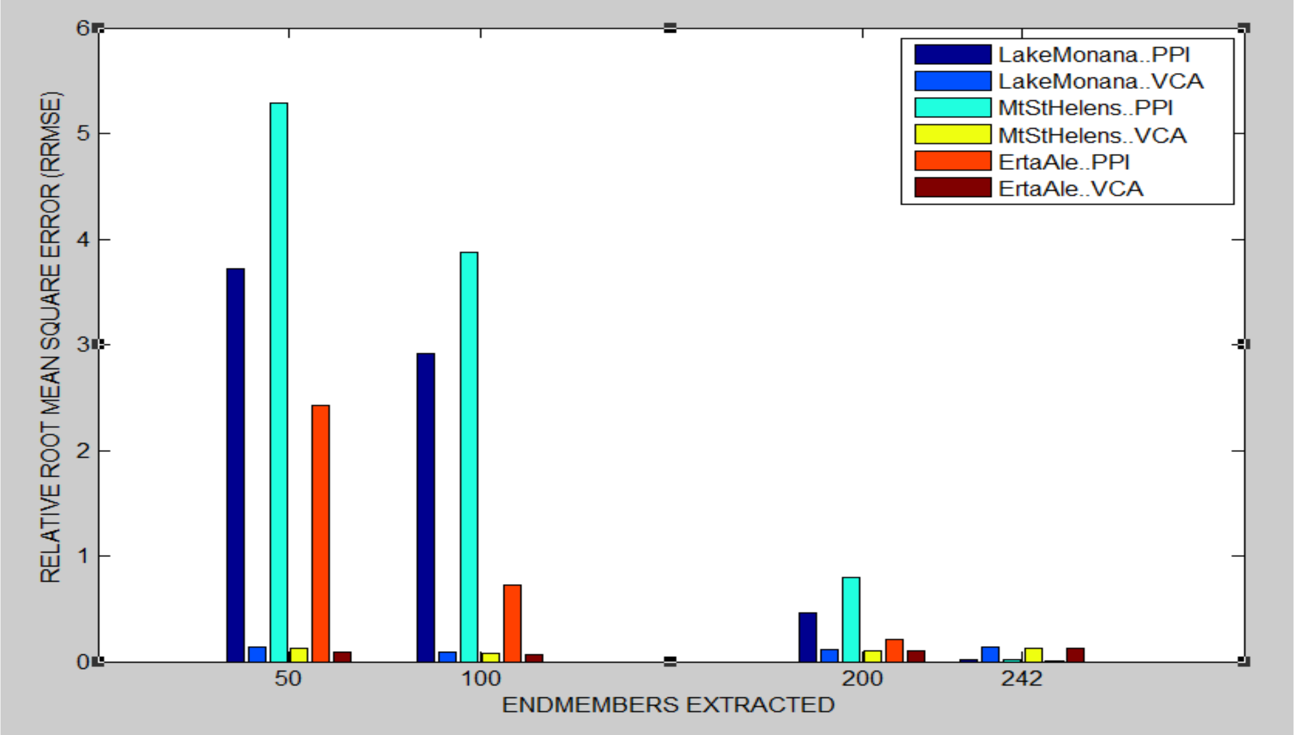


Fig. 6.13.Comparison of RRMSE values of reconstructed image from various no of endmembers extracted via PPI and VCA.

Table 6.1. Degradation comparison of reconstructed images from PPI and VCA endmembers and its quantized fractional abundance image

Quality metric/method	**Reconstructed image from 242 end members and quantized fractional abundance**	
	Lakemona_PPI	***Lakemonana_VCA***
MAD	281	1668
MAE	41.95	810.40
MSE	4.0272e+03	7.4103e+05
RRMSE	0.6995	11.6178

From table 6.1. it is inferred that quantisation of fractional abundance image leads to adverse degradation of reconstructed image(Lakemonana) which is evident from MSE values in range of 1000 and 10000. Thus for compression quantisation of fractional abundance image is not preferrable.Reduction in no of endmembers is only prefered. Figure 6.14 & 6.15 shows the comparison of MSE and RRMSE among VCA based method with various number of endmembers extracted.

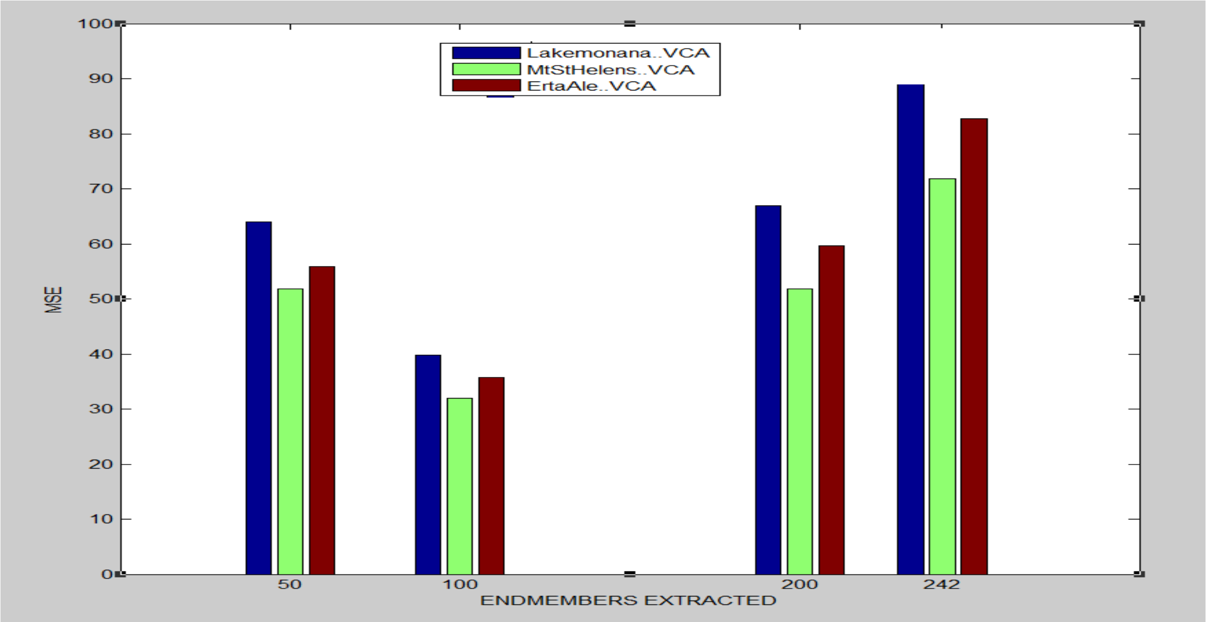


Fig. 6.14.Comparison of MSE of reconstructed image from various no of endmembers extracted via VCA.

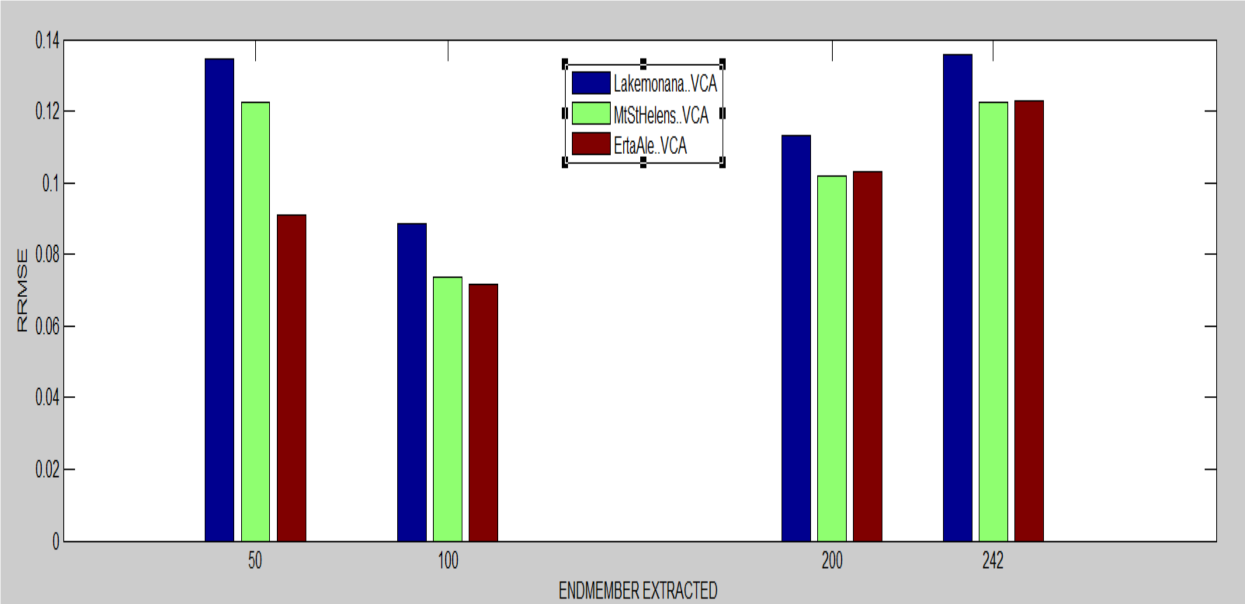


Fig. 6.15.Comparison of RRMSE of reconstructed image from various no of endmembers extracted via VCA.

Compression ratio is given by

$$CR = \frac{Uncompressed\_size}{compressed\_size} \qquad (6.6)$$

Space saving achieved is given by

$$S = 1 - \frac{compressed\_size}{uncompressed\_size} * 100 \qquad (6.7)$$

Table 6.2. Comparison of no of pixels of raw image and spectrally unmixed images

Image	**Row*Column*Bands**	**No Of Pixels**
Lakemonana subset	256*256*242	15859712
After Spectral Unmixing		
Abundance image	256*256*100	6553600
End member image	242*100	24200
no of pixels reduced = 15859712 - (6553600+ 24200) = 9281912 pixels		

Thus by VCA based method with 100 endmembers extracted, instead of sending 242x256x256 image one has to send 100x242 endmember image and 100X256X256 fractional abundance image only. The reduction in no of pixels =242X256X256-[(100X242) + (100X256X256)] = 9281912 pixels. Thus by spectral unmixing spectral redundancy is reduced that can be inferred from the table 6.2.

Table 6.3. Comparison of size of raw image and compressed image with 100 endmembers extracted by VCA

Image	Size
Raw image LakeMonana (256x256x242 subset)	31.5MB
After Spectral Unmixing	
Fractional_abundance image	13.1MB
Endmember image	49KB
After Lossless Predictive Coding	
Encoded Fractional_abundance image	10.86MB
Endmember image	40KB

Table 6.3 provides information about compression achieved after spectral and spatial compression methods. After spectral unmixing each fractional abundance image is checked for spatial redundancy through lossless predictive coding and errors are encoded using arithmetic coding. There are 100 fractional abundance images of size 256x256 each. These images are converted to integer by scaling and LPC is applied. Size of each fractional abundance image obtained=256*256*2=131072bytes=131kb.Through LPC it is been reduced to 80kb to 110kb. The spectral compression achieved is inversely proportional to number of endmembers being extracted. It increases with reduction in number of endmembers and the compression achieved in spatial compression is data dependent. For 100 endmembers being extracted with VCA the compression ratio obtained 2.40:1. The space saving achieved after spectral compression is 58.33%.The proposed two stage spectral/spatial compression of hyperspectral image has achieved compression ratio of 2.9:1 and space savings achieved of about 65.51% after both the stages and it is inferred from bar charts that the proposed method has well retained its quality over reconstruction from compressed images of VCA based method.

CHAPTER 7

CONCLUSION

In this work two stage spectral/spatial method of compressing hyperspectral image has been proposed. Linear Spectral Unmixing via Vertex Component Analysis algorithm has been used as preprocessing stage for mitigating spectral redundancy. Lossless predictive coding with arithmetic encoder is used for reducing spatial redundancy. From the experimental results, it is evident that the reduction of spectral and spatial redundancy is possible without compromising quality over reconstruction. The proposed method of spectral compression outperforms existing compression algorithm with Pixel Purity Index (PPI) in terms of reconstruction quality. The reconstruction quality of both the methods has been examined with quality metrics viz MSE, MAD, MAE and RRMSE. Moreover this VCA algorithm based spectral reduction retains spectral crucial information since it is utilizing extraction of pure signatures from various bands of hyperspectral image. The compression ratio achieved after first stage spectral compression is 2.4:1 and compression ratio achieved after both spectral and spatial compression is 2.9:1. Thus proposed method is proved to have achieved high compression ratio and have retained good quality over reconstruction. This work has suggested low complexity parallel implementation architecture for implementing Vertex Component Analysis in spectral compression stage. FPGA implementation of proposed hyperspectral image compression with suggested architecture can be done in future.

REFERENCES

[1]. Agnieszka C. Miguel, Alexander Chang, Richard E. Ladner, Scott Hauck, Eve A. Riskin "On-Board Satellite Implementation of Wavelet-Based Predictive Coding of Hyperspectral Images".

[2]. Ahmed Hagag, Mohamed Amin, Fathi E. Abd El-Samie, (2013) "Multispectral image compression with band ordering and wavelet transforms", *Springer-Verlag*, London.

[3]. Antonio Plaza, (2007) "Towards Real-Time Compression of Hyperspectral Images Using Virtex-II FPGAs", *Springer-Verlag Berlin Heidelberg*, LNCS 4641, pp. 248-257.

[4]. Antonio Plaza, Pablo Martínez, Rosa Pérez, and Javier Plaza (2004) "A Quantitative and Comparative Analysis of Endmember Extraction Algorithms From Hyper spectral Data", *IEEE Transactions On Geoscience And Remote Sensing*, Vol. 42, No. 3, pp.650-663.

[5]. Antonio Plaza, Sergio Sánchez, Abel Paz and Javier Plaza "GPUs versus FPGAs for Onboard Payload Compression of Remotely Sensed Hyperspectral Data".

[6].AVIRIS airborne imager data available at NASA Jet Propulsion Lab websites http://aviris.jpl.nasa.gov/data/free_data.html and http://compression.jpl.nasa.gov/hyperspectral.html.

[7]. Bharath Ramakrishna, Jing Wang, Chein-Chang, Antonio Plaza, Hsuan Ren Chein-Chi Chang, James O.Jensen, (2005)"Spectral/Spatial Hyperspectral Image Compression in Conjunction with Virtual Dimensionality", *Proceedings of SPIE* Vol. 5806,SPIE, Bellingham, WA.

[8]. Carlos Gonz´alez, Daniel Mozos, Javier Resano and Antonio Plaza (2011) " FPGA Implementation of Endmember Extraction Algorithms from Hyperspectral Imagery: Pixel Purity Index versus N-FINDR " , *High-Performance Computing in Remote Sensing Proc. of SPIE* , Vol. 8183, 81830 F-1-12.

[9]. Carlos González, Daniel Mozos, Javier Resano, and Antonio Plaza (2012),"FPGA Implementation of the N-FINDR Algorithm for Remotely Sensed Hyper spectral Image Analysis", *IEEE Transactions On Geoscience And Remote Sensing*, Vol. 50, No. 2.

[10]. Carole Thiebaut, Emmanuel Christophe, Dimitri Lebedeff, Christophe Latry "CNES Studies of On-Board Compression for Multispectral and Hyperspectral Images".

[11]. Chein-I Chang, ChiaHsien Wen, ChaoCheng Wu (2013) "Relationship exploration among PPI, ATGP and VCA via theoretical analysis" , *Int. J. of Computational Science and Engineering*, Vol.8, No.4, pp.361-367.

[12]. David Valencia, Antonio Plaza, Miguel A. Vega-Rodríguez, and Rosa M. Perez (2005) "FPGA Design and Implementation of a Fast Pixel Purity Index Algorithm for Endmember Extraction in Hyperspectral Imagery", *Proc. of SPIE* , Vol. 5995, pp. 599508-1 -10.

[13]. David Landgrebe, (1999) "Some Fundamentals and Methods for Hyperspectral Image Data Analysis", *SPIE Photonics West*, San Jose CA.

[14]. Dominique Lavenier, Erwan Fabiani, Steven Derrien, Charles Wagner "Systolic array for computing the pixel purity index (PPI) algorithm on hyper spectral images".

[15]. Emmanuel Christophe, Dominique L´eger, and Corinne Mailhes (2008), " New Quality Representation for Hyperspectral Images" *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XXXVII. Part B7. Beijing, pp.315-320.

[16]. Emmanuel Christophe, Dominique Léger, Corinne Mailhes (2005), "Quality Criteria Benchmark for Hyperspectral Imagery" *IEEE Transactions On Geoscience And Remote Sensing*, Vol. 43, No. 9, pp.2103-2114.

[17]. Firouz Abdullah Al-Wassai, N.V. Kalyankar, (2006) "Major Limitations of Satellite images".

[18]. Hassan Ghassemian "On-Board Satellite Image Compression by Object-Feature Extraction".

[19]. José M. P. Nascimento, José M. Bioucas Dias (2005), “Vertex Component Analysis: A Fast Algorithm to Unmix Hyperspectral Data”, *IEEE Transactions on Geoscience and Remote Sensing*, Vol. 43, No. 4, pp.898-910.

[20]. Jose M. Rodrıguez Alves, Jose M. P. Nascimento, Jos´e M. Bioucas-Dias, Vıtor Silva, Antonio Plaza (2012), “Parallel Implementation of Vertex Component Analysis forHyperspectral Endmember Extraction”, *IEEE International Symposium on Geoscience and Remote Sensing*, IGARSS, pp.4078-4081.

[21]. M. Klimesh, A. Kiely, H. Xie and N. Aranki, (2005) “Spectral Ringing Artifacts in Hyperspectral Image Data Compression”, *IPN Progress Report* 42-160.

[22]. Nazeeh Aranki (2007) “Parallel Implementations Of The Discrete Wavelet Transform And Hyperspectral Data Compression On reconfigurable Platforms Approach, Methodology And Practical Considerations”.

[23]. Peg Shippert, (2004) “Why Use Hyperspectral Imagery?”, Photogrammetric Engineering & Remote Sensing, *Journal of the American Society for Photogrammetry and Remote Sensing*, Vol. 70, No. 4, pp. 377 – 380.

[24]. Pekka Toivanen, Olga Kubasova, and Jarno Mielikainen, (2005) “Correlation-Based Band-Ordering Heuristic for Lossless Compression of Hyperspectral Sounder Data”, *IEEE Geoscience And Remote Sensing Letters*, Vol. 2, No. 1.

LIST OF PUBLICATIONS

- Presented a paper entitled **"Spectral Unmixing With Vertex Component Analysis"** in the third International Conference on Signal Processing, Communication And Networking (ICSCN) held during 26th to 28th, march 2015 at Anna University, Madras Institute of Technology Campus, Chennai.