On the Use of Cluster Computing Architectures for Implementation of Hyperspectral Image Analysis Algorithms

David Valencia, Antonio Plaza, Pablo Martínez and Javier Plaza
Department of Computer Science, University of Extremadura, Cáceres, Spain
{davaleco, aplaza, pablomar, jplaza}@unex.es

Abstract

Hyperspectral sensors represent the most advanced instruments currently available for remote sensing of the Earth. The high spatial and spectral resolution of the images supplied by systems like the Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS), developed by NASA Jet Propulsion Laboratory, allows their exploitation in diverse applications, such as detection and control of wild fires and hazardous agents in water and atmosphere, detection of military targets and management of natural resources. Even though the above applications require a response in real time, few solutions are available to provide fast and efficient analysis of these types of data. This is mainly caused by the dimensionality of hyperspectral images, which limits their exploitation in analysis scenarios where the spatial and temporal requirements are very high. In the present work, we describe a new parallel methodology which deals with most of the previously addressed problems. The computational performance of the proposed analysis methodology is evaluated using two parallel computer systems, an SGI Origin 2000 shared memory system located at the European Center of Parallelism of Barcelona, and the Thunderhead Beowulf cluster at NASA's Goddard Space Flight Center.

1. Introduction

The development of advanced instruments for remote observation of the Earth has created a growing interest in the design of efficient techniques for the interpretation of the images provided by these sensors. In particular, hyperspectral sensors are characterized by their high resolution in both spatial and spectral domains [1]. For instance, the Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS), developed by NASA Jet Propulsion Laboratory [2] covers the range of wavelengths from 0.4 to 2.5 mm using 224 spectral channels, with a spatial resolution of 20 meters per pixel and a nominal spectral resolution of 10 nm. As shown in Fig. 1, the analytic capability of AVIRIS allows for the collection of a detailed spectral signature for each pixel in the image, where the spectral signature at each pixel is given by a series of reflectance values obtained by the sensor at different wavelengths.

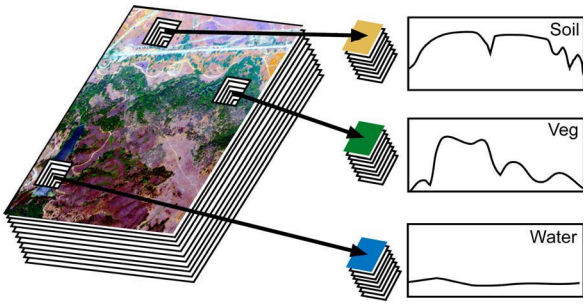

Fig. 1. Concept of hyperspectral imaging.

Despite the significant technological evolution of hyperspectral instruments, developments in techniques for analysis of the data provided by those sensors have not been so notorious. In particular, design of analysis techniques able to take advantage of both spatial and spectral information contained in the data is still a challenge for the scientific community [3]. While integrated spatial/spectral developments hold great promise for Earth science image analysis, they create new processing challenges. The price paid for the wealth spatial and spectral information available from hyperspectral sensors is the enormous amounts of data that they generate. As a result, analysis techniques in Earth observation studies often require lengthy durations to calculate desired quantities. Several applications exist, however, where having the desired information calculated in near real-time is highly desirable. Such is the case of applications aimed at detecting and/or tracking natural disasters such as forest fires, oil spills, and other types of chemical contamination, where timely classification is highly desirable.

Parallel computing techniques have been widely used in image analysis and remote sensing tasks [4–8].

Proceedings of the 10th IEEE Symposium on Computers and Communications (ISCC 2005)
1530-1346/05 $20.00 © 2005 **IEEE**

Massively parallel Beowulf clusters, based on the Linux Operating System, were conceived at NASA's Goddard Space Flight Center (NASA/GSFC) in Maryland with the goal of creating cost-effective systems to satisfy specific computational requirements in the Earth and space sciences community [9].

In this work, we describe a new parallel algorithm for efficient exploitation of hyperspectral image data. The main contribution of the proposed algorithm is its natural parallel framework for integration of spatial and spectral information. Section 2 describes the proposed method, which relies on multi-channel mathematical morphology concepts. Section 3 provides a detailed description of its parallel implementation. Section 4 conducts a study of the computational performance of the parallel implementation in two parallel computers: an SGI Origin 2000 located at the European Center of Parallelism of Barcelona and the Thunderhead system at NASA/GSFC. Finally, section 5 concludes with some remarks and hints at plausible future research.

2. Methodology

The proposed method for hyperspectral analysis can be included in the category of spectral unmixing approaches [10]. In the next subsection we describe the problem of the spectral mixing and then introduce a set of morphological operations oriented to resolve this problem. This section concludes with a description of our proposed algorithm, which will be parallelized in the following section.

2.1. Spectral Unmixing

Mixed pixels are predominant in hyperspectral images and result as mixtures of more than one distinct substance. They exist mainly due to available spatial resolution, often not sufficient to separate different materials. Spectral unmixing is a commonly used procedure in which the measured spectrum of a mixed pixel is decomposed into a collection of spectrally pure constituent spectra, or endmembers [11], and a set of correspondent fractions, or abundances, that indicate the proportion of each endmember present in the pixel.

Identification of image endmembers is a crucial objective in hyperspectral image analysis applications. Most available techniques for endmember selection focus on analyzing the data without incorporating information on the spatially adjacent data; i.e. the hyperspectral data is treated not as an image but as an unordered listing of spectral measurements where the spatial coordinates can be shuffled arbitrarily without affecting the analysis. Subsequently, there is a need to incorporate the image representation of the data in the development of automated techniques for endmember selection and hyperspectral data exploitation. The main contribution of the method described in this work is simultaneous consideration of both spatial and spectral information. By taking into account the complementary nature of spatial and spectral information in simultaneous fashion, it is possible to alleviate the problems related to each of them taken separately.

2.2. Morphological Method

The proposed method is based on mathematical morphology [3], a classic image analysis technique that is generalized to the case of multidimensional data in this subsection. Two basic operations articulate classic MM theory: erosion and dilation. They are respectively based on the selection of the maximum and minimum value of a neighborhood or spatial region around each pixel of the image, where the shape and size of the considered region are determined by the spatial properties of a neighborhood function called structuring element (SE). The main challenge in order to extend these operations to the case of hyperspectral image data is the lack of an ordering relation between the pixels of the image, which can be seen as L-dimensional (L-D) vectors where L is the number of spectral channels (see Fig. 1). Following a usual notation, let $\boldsymbol{f}$ be an image defined on an L-D space and let B a so-called SE. We impose an ordering relation by defining a cumulative distance between one particular pixel $\boldsymbol{f}(x,y)$, where $\boldsymbol{f}(x,y)$ denotes an L-D vector at discrete spatial coordinates $(x,y)\in Z^2$, and all the pixel vectors in the spatial neighborhood given by B (B-neighborhood) as follows:

$$D_B[\boldsymbol{f}(x,y)] = \sum_s \sum_t \mathrm{Dist}[\boldsymbol{f}(x,y), \boldsymbol{f}(s,t)]$$

where Dist is the spectral angle distance [11]. As a result, $D_B[\boldsymbol{f}(x,y)]$ is given by the sum of Dist scores between $\boldsymbol{f}(x,y)$ and every pixel vector in the B-neighborhood. Based on the cumulative distance above, the erosion of $\boldsymbol{f}$ by B, denoted by $(\boldsymbol{f} \ominus B)(x,y)$, selects the pixel vector that produces the minimum value for D_B [3]. On the other hand, the dilation of $\boldsymbol{f}$ by B, denoted by $(\boldsymbol{f} \oplus B)(x,y)$, selects the pixel vector that produces the maximum value for D_B.

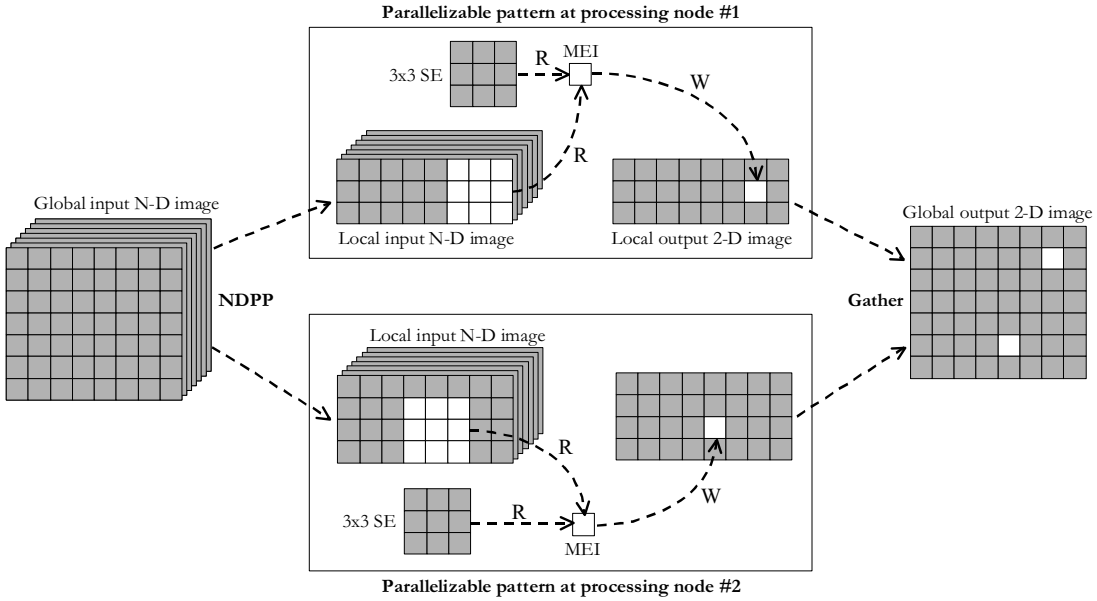

Fig. 2. Concept of spatia/spectral parallelizable pattern, and proposed partitioning scheme.

2.3. Endmember Extraction Algorithm

The proposed method to detect endmembers is called Automated Endmember Extraction Algorithm (AMEE) [3]. This method, entirely developed in our laboratory, allows for the analysis of hyperspectral images in automated fashion. The inputs are an N-D image $\boldsymbol{f}$, a structuring element B, a number of iterations I_{MAX}, and the number of endmembers p:

1. Set $i = 1$ and initialize a morphological eccentricity index $MEI(x,y) = 0$ for each $\boldsymbol{f}(x,y)$.
2. Move B through all the pixels of $\boldsymbol{f}$, defining a local spatial search area, and calculate the maximum $(\boldsymbol{f} \oplus B)(x,y)$ and the minimum $(\boldsymbol{f} \Theta B)(x,y)$ pixel at each B-neighborhood.
3. Update the resulting MEI score at each pixel selected as a local maximum using the spectral angle between the result of the dilation and the result of the erosion: $\boldsymbol{f}(x',y') = (\boldsymbol{f} \oplus B)(x,y)$.
4. Set $i = i+1$. If $i = I_{max}$ then go to step 5. Otherwise, set $\boldsymbol{f} = \boldsymbol{f} \oplus B$ and go to step 2.
5. Select the pixels $\{\boldsymbol{e}_j\}_{j=1}^{p}$ with higher MEI score.

3. Parallel Implementation

The parallel implementation of the proposed method in section 2.3 has been developed using partitioning techniques in the spatial domain. In this section we will fully justify our choice for the partitioning scheme and further introduce the concept of spatial/spectral parallelizable pattern (SSPP). The section concludes with a short summary of the operations realized by the proposed parallel implementation.

3.1. Partitioning Scheme

Two different approaches are considered in order to partition the data: partitioning in the spatial domain and partitioning in the spectral domain. The first option divides the hyperspectral image in multiple blocks, in a way that the pixels for each block preserve its entire spectral identity. The second option divides the original image in blocks constituted by several bands, in a way that we can preserve the spatial identity for each band but all the pixels in each block lose their spectral identity. In other words, if the partitioning scheme adopted were in the spatial domain, the information of a single pixel in the image would be scattered across several different processing units.

If we take in account the fundamental characteristics of our method, which works with all of the spectral information associated to each pixel, the selection of a partitioning scheme in the spectral domain is critical and could substantially increase the costs of communication between processors [4]. The overhead introduced by the communication increases with the number of processors, thus introducing load balance problems [12]. On other hand, the spatial

information is particularly relevant in the local neighborhood around each pixel [3]. This is a reason why a partitioning scheme in the spatial domain is able to preserve most of the information required for our morphological processing.

At this point, we introduce the concept of spatial/spectral parallelizable pattern (SSPP), which is defined as the maximum amount of information that the parallel system can process without the need for additional communication and/or coordination between processors [2]. Such patterns are automatically generated by an N-dimensional parallel partitioning module (NDPP), as Fig. 2 describes using two computing units. At the end of the process, the NDPP fuses the various local images obtaining a resulting 2-D image used as a baseline to extract a final set of endmembers.

An issue of major importance in the design of SE-based parallel image processing applications is the possibility to access pixels out of the spatial domain of the partition available in the processor. In our parallel implementation, only the pixels of the SE which fall inside the image domain are considered for the morphological processing. In addition, when the pixel located in a remote processor is required in the calculation of the MEI index associated with another pixel in a given processor, we replicate the information necessary to avoid such border in the first processor, thus introducing redundant information in the system. According to our preliminary experiments, the cost of processing such redundant information is inferior to the overhead introduced by communication among different processors. Given the characteristics of the implementation proposed in section 2, which relies on the utilization of an SE of 3x3 pixels iteratively, the number of redundant pixels R introduced in the processing of a hyperspectral image is given by

$$R = 2\times\left[\left(2^{\frac{\log_2 N}{2}}\right)-1\right]\times I_F + 2\times\left[\left(2^{\frac{\log_2 N}{2}}\right)-1\right]\times I_C$$

where N is the number of processors, I_F is the number of rows in the original image and I_C is the number of columns in the original image. For example, in order to process an AVIRIS image of 512x512 pixels with 16 processors, redundant pixels are given by $R = 2\times\left[\left(2^2\right)-1\right]\times 512 + 2\times\left[\left(2^2\right)-1\right]\times 512 = 6144$. If we assume that each pixel has 224 spectral values, each of them coded using two bytes, the total amount if redundant information introduced in the system is 2,625 MB (6144x224x2) which, compared with the total size of the original image in bytes (about 114 MB), can be considered insignificant. As noted above, the amount of redundant information is below 2.5 % of the total information present in the original image. It is important to point out that the amount of redundant information grows as the number of processors increases, a fact that introduces a limit to the performance of the parallel code which is directly related to the problem of having more redundant information than pixels to process inside a certain SE.

3.2. Summary of Operations

The parallel implementation described in Fig. 2 is based on a partitioning scheme in the spatial domain in which one of the processors acts as the master node (NDPP) in charge of the I/O operations. The partitioner has been implemented so that it automatically determines the optimum size for the SSPPs to be distributed between the different processors. The NDPP sends to each processor a portion (SSPP) of the original image. Each processor works locally with its corresponding portion. Once it has finished the local processing, each processor sends the results back to the NDPP. Finally, the NDPP compounds the partial results and carries out the process of selecting the final endmembers using the information provided by each of the processors. Performance data for the parallel algorithm are given in section 4.

4. Experimental Results

This section describes the performance of the parallel implementation in section 3 in terms of its computational efficiency (speedup) compared with the serial version of the code, and also in terms of its accuracy in the fully automated classification of hyperspectral images. In a first subsection we describe the parallel computers used in the study, while in the second subsection we discuss the obtained results in the analysis of a well-known AVIRIS image.

4.1. Parallel Computers

Two parallel computers have been used to evaluate the computational performance of the morphological algorithm proposed. The first is an SGI Origin 2000 cache coherent shared memory system with non uniform (latency) memory access, located at the European Center of Parallelism of Barcelona. It is composed of 64 MIPS R10000 processors (each of them with 4 Mb of cache and 12 GB of main memory) connected through an intercommunication network of 1.2 Gbps. The theoretical peak performance of the system is 32 Gflops. The operating system used during our experiments was Irix 5.6, and the software was

compiled using mpicc available from MIPSpro 7.3.1.2 suite. The second parallel computer used in the study is a Beowulf type cluster named Thunderhead located at the Applied Information Sciences Branch of NASA/GSFC. This system consists of 256 nodes, each of them with two 2.4 GHz Intel Xeon processors. Each node has 1 Gb of local memory. The communication network is Myrinet at 2 GHz (optical fibre). The operating system in Thunderhead is Linux Red Hat 8.0, and MPICH is the communication library.

4.2. Results and Discussion

To empirically investigate the scaling properties of the parallel algorithm, we have used a hyperspectral image obtained by the AVIRIS sensor in June 1992 over a small area (145 lines by 145 samples and 220 spectral bands) gathered over the Indian Pines test site in Northwestern Indiana. The data set represents a very challenging classification problem due to the presence of mixed pixels.

Using the information provided by available ground truth information, we have analyzed the cost-performance accuracy of the proposed morphological approach. Our classification scheme consisted of the following steps: 1) Endmember extraction via morphological operations, 2) Classification of each pixel as belonging to a class given by the most abundant endmember in the pixel. The following parameters were considered: I_{MAX} was set to 1, 3, 5 and 7 iterations, respectively. B is a 3x3-pixel structuring element of fixed size, and p, the maximum number of endmembers to be detected that was set to $p = 16$ after calculating the intrinsic dimensionality of the data using the Harsanyi-Farrand-Chang (HFC) method in [1]. Setting $I_{MAX} = 7$ resulted in an overall accuracy of more than 90% and very high classification scores for all the individual ground-truth classes. Classification accuracies do not significantly improve for $I_{MAX} > 7$.

In order to analyze the scalability of the parallel code, Fig. 3 plots the speedup factors as a function of the number of available processors N at the SGI Origin 2000 and Thunderhead computers. The factors were calculated as follows. First, the real time required to complete a task on N parallel processors, $T(N)$, was approximated by $T(N) = A_N + \frac{B_N}{K}$, where A_N is the sequential (non-parallelizable) portion of the computation and B_N is the parallel portion. In our parallel application, A_N corresponds to the sequence of operations implemented by the NDPP module, and B_N corresponds to the selection of endmembers. Then, we can define the speedup for N processors, S_N, as $S_N = \frac{T(1)}{T(N)} \approx \frac{A_N + B_N}{A_N + (B_N / N)}$, where $T(1)$ denotes single processor time. The relationship above is generally known as Amdahl's Law. It is obvious from this expression that the speedup of a parallel algorithm does not continue to increase with increasing the number of processors. The reason is that the sequential portion A_N is proportionally more important as the number of processors increase and, thus, the performance of the parallelization is degraded for a large number of processors. Since only the parallel portion B_N scales with the time required to complete the calculation and the serial component remains constant, there is a theoretical limit for the maximum parallel speedup achievable for N processors, which is given by $S_\infty^N = \lim_{N\to\infty} S_N = \frac{A_N + B_N}{A_N} = 1 + \frac{B_N}{A_N}$.

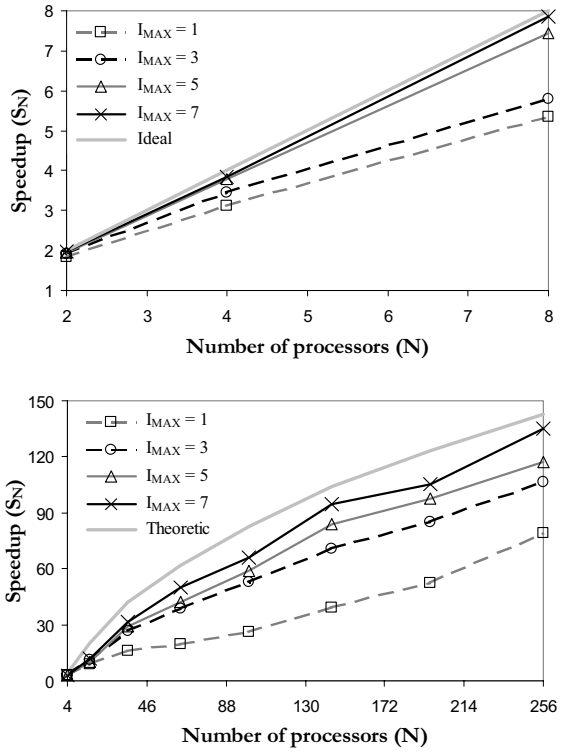

Fig. 3. Parallel performance in the SGI Origin 2000 (top) and Thunderhead Beowulf cluster (bottom).

It should be noted that a maximum number of 8 and 256 processors were respectively utilized in the SGI Origin 2000 and Thunderhead, respectively. Due to

queue limitations, we could only access half of the processors available at Thunderhead. Although we are aware that it may not be fair to compare scalability of systems with such a large difference in the processor count, we emphasize that this limitation was due to system availability at the time of the experiments.

From results in Fig. 3, we can conclude that the proposed parallel algorithm achieves significant speedups when compared to the serial implementation in the two parallel computers. As shown in Fig. 3 (top), the small system even shows a slight superlinear scaling, probably due to cache reuse. Also, the measured speedups tend to be higher for large values of I_{MAX}, a fact that reveals that the proposed scheme scales better when the number of morphological operations to be accomplished is very high.

N	$I_{MAX} = 1$	$I_{MAX} = 3$	$I_{MAX} = 5$	$I_{MAX} = 7$
1	311	947	1528	1925
4	124	321	557	685
16	45	95	144	156
36	26	46	61	71
64	19	29	41	43
100	12	20	26	29
144	9	15	20	23
196	6	11	17	20
256	4	10	14	18

Table 1. Execution time (seconds) measured at NASA/GSFC Thunderhead Beowulf cluster.

From Fig. 3, it is also clear that the speedups produced by the parallel algorithm are close to the correspondent theoretic values, especially when I_{MAX} is set to a high number. Table 1 shows the execution times in seconds of the proposed algorithm with the AVIP92 scene for several combinations of number of iterations and number of processors in Thunderhead. As shown by Table 1, the utilization of 256 processors allows near real-time processing of the AVIRIS scene: only 18 seconds were required to produce an overall classification score above 90%, which is a good result in light of the complexity of the scene.

5. Conclusions and Future Work

We have described a parallel algorithm able to process high-dimensional hyperspectral images in near real time. Experimental results suggest that our parallel algorithm provides adequate results in both the quality of the solutions and the time to obtain them. Specifically, the proposed algorithm can produce a fast response in applications with near real-time requirements. The proposed algorithm may be of great utility to detect and monitor forest fires, such as those that recently happened in the Extremadura region in SW Spain. In this regard, our future research line is to integrate the proposed parallel algorithm onto an automated forest fire tracking system in conjunction with Junta de Extremadura (local government).

References

[1] C.-I Chang, *Hyperspectral imaging: spectral detection and classification*, Kluwer Academic/Plenum Publishers, New York, 2003.

[2] A. Plaza, P. Martínez, R. Pérez, J. Plaza, "Parallel implementation of endmember extraction algorithms from AVIRIS hyperspectral imagery," *NASA/JPL Airborne Earth Science Workshop*, Pasadena, CA, 2004.

[3] A. Plaza, P. Martínez, R. Pérez, J. Plaza, "Spatial/spectral endmember extraction by multidimensional morphological operations," *IEEE Trans. Geosci. Remote Sensing*, vol. 40, pp. 2025-2041, 2002.

[4] F.J. Seinstra, D. Koelma, J.M. Geusebroek, "A software architecture for transparent parallel image processing," *Parallel computing*, vol. 28, pp. 967-923, 2002.

[5] M. K. Dhodhi et al., "D-ISODATA: A distributed algorithm for unsupervised classification of remotely sensed data on network of workstations," *Journal of Parallel and Distributed Computing*, vol. 59, pp. 280-301, 1999.

[6] K. Itoh, "Massively-parallel Fourier-transform spectral imaging and hyperspectral image processing," *Optics & Laser Technology*, vol. 25, pp. 202, 1993.

[7] G. Aloisio and M. Cafaro, "A dynamic earth observation system," *Parallel Computing*, vol. 29, pp. 1357-1362, 2003.

[8] K. A. Hawick et al., "Distributed frameworks and parallel algorithms for processing large-scale geographic data," *Parallel Computing*, vol. 29, pp. 1297-1333, 2003.

[9] J. Le Moigne, W. J. Campbell, and R. F. Cromp, "Automated parallel image registration based on correlation of wavelet features," *IEEE Trans. Geosci. Remote Sensing*, vol. 40, pp. 1849–1864, 2002.

[10] A. Plaza, P. Martínez, R. Pérez, J. Plaza, "A new approach for mixed pixel classification in hyperspectral imagery based on extended morphological profiles," *Pattern Recognition*, vol 37, pp. 1097-1116, 2004.

[11] A. Plaza, P. Martínez, R. Pérez, J. Plaza, "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Trans. Geosci. Remote Sensing*, vol. 42, pp.650-663, 2004.

[12] P.L. Aguilar, A. Plaza, R.M. Perez, P. Martinez, "Morphological endmember identification and its systolic array design," Chapter 3 in *Neural Networks and Systolic Array Design*, D. Zhang & S.K. Pal, Eds, pp. 47-69, 2002.