
Parallel Spatial-Spectral Processing of Hyperspectral Images

Antonio J. Plaza

Department of Technology of Computers and Communications
University of Extremadura, Avda. de la Universidad s/n
E-10071 Cáceres, Spain
aplaza@unex.es

Summary. Hyperspectral image processing has been a very active area in remote sensing and other application domains in recent years. Despite the availability of a wide range of advanced processing techniques for hyperspectral data analysis, a great majority of available techniques for this purpose are based on the consideration of spectral information separately from spatial information, and thus the two types of information are not treated simultaneously. In this chapter, we describe several spatial-spectral techniques for dimensionality reduction, feature extraction, unsupervised and supervised classification, spectral unmixing and compression of hyperspectral image data. Most of the techniques addressed in this chapter are based on concepts inspired by mathematical morphology, a theory that provides a remarkable framework to achieve the desired integration. Parallel implementations of some of the proposed techniques are also developed to satisfy time-critical constraints in remote sensing applications, using NASA's Thunderhead Beowulf cluster for demonstration purposes throughout the chapter. Combined, the different topics covered by this chapter offer a thoughtful perspective on future potentials and emerging challenges in the design of robust spatial-spectral techniques for hyperspectral image analysis.

7.1 Introduction

Hyperspectral imaging is concerned with the measurement, analysis, and interpretation of spectra acquired from a given scene (or specific object) at a short, medium or long distance by an airborne or satellite sensor [1]. The concept of hyperspectral imaging originated at NASA's Jet Propulsion Laboratory in California, which developed instruments such as the Airborne Imaging Spectrometer (AIS), then called AVIRIS, for Airborne Visible Infra-Red Imaging Spectrometer [2]. This system is now able to cover the wavelength region from 0.4 to 2.5 μm using more than two hundred spectral channels, at nominal spectral resolution of 10 nm. As a result, each pixel vector collected by a hyperspectral instrument can be seen as a *spectral signature* or *fingerprinth* of the underlying materials within the pixel (see Fig. 7.1).

The special characteristics of hyperspectral datasets pose different processing problems, which must be necessarily tackled under specific mathematical formalisms, such as feature extraction, classification and segmentation, spectral

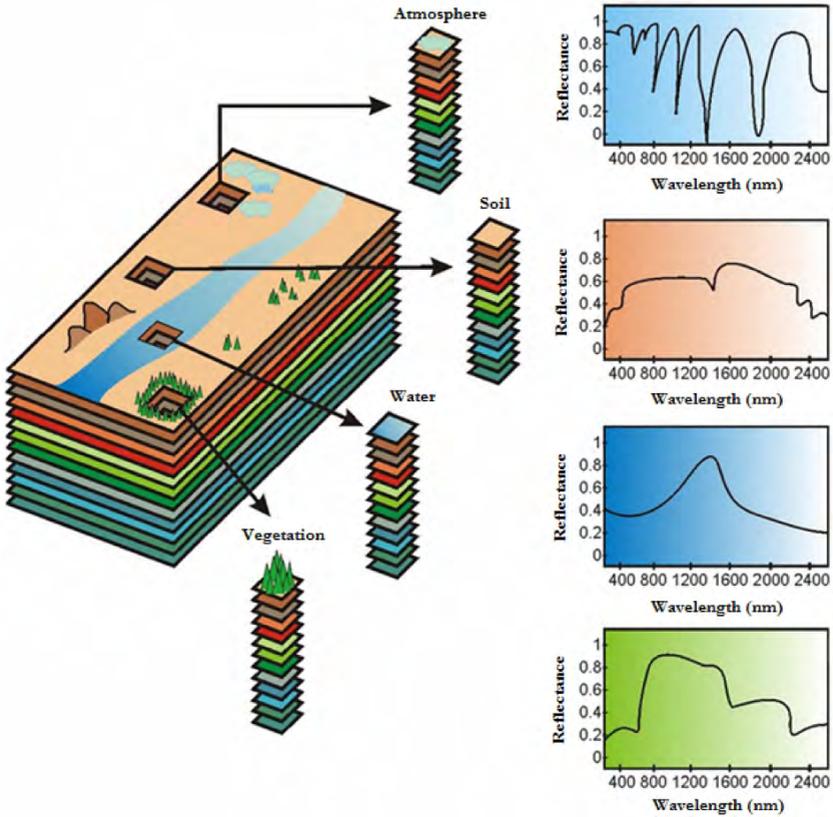


Fig. 7.1. The concept of hyperspectral imaging

mixture analysis, or data compression [3]. For instance, several computational intelligence-related techniques have been applied to extract relevant information from hyperspectral data during the last decade [4]. Taxonomies of remote sensing data processing algorithms (including hyperspectral analysis methods) have been developed in the literature [5]. It should be noted, however, that most available hyperspectral data processing techniques have focused on analyzing the data without incorporating information on the spatially adjacent data, i.e., hyperspectral data are usually not treated as images, but as unordered listings of spectral measurements where the spatial coordinates can be randomly shuffled without affecting the analysis [6]. However, one of the distinguishing properties of hyperspectral data, as collected by available imaging spectrometers, is the multivariate information coupled with a two-dimensional pictorial representation amenable to image interpretation. Subsequently, there is a need to incorporate the image representation of the data in the development of appropriate application-oriented techniques for the understanding of hyperspectral data [7].

Specifically, the importance of analyzing spatial and spectral patterns simultaneously has been identified as a desired goal by many scientists devoted to hyperspectral data analysis [6], [8]. This type of processing has been approached in the past from various points of view. For instance, techniques have discussed the refinement of results obtained by applying spectral-based techniques to multispectral images (with tens of spectral channels) through a second step based on spatial context [9]. Such contextual classification, extended also to hyperspectral images [6], accounts for the tendency of certain ground cover classes to occur more frequently in some contexts than in others. This approach consists of two parts: the definition of a pixel neighborhood (surrounding each pixel) and the performance of a local operation so that the pixel may be changed into the label mostly represented in the window that defines the neighborhood. This simple operation separates spatial from spectral information, and thus the two types of information are not treated simultaneously.

In previous work, we have explored the application of mathematical morphology operations to integrate both spatial and spectral responses in hyperspectral data analysis [10], [11]. Mathematical morphology is a non-linear spatial processing technique that provides a remarkable framework to achieve the desired integration [12]. It was originally defined for binary images [13] and soon extended to grayscale and color image cases, but it has been seldom used to process hyperspectral images. In this chapter, we provide detailed insight on the use of extended morphological operations for integrating spatial and spectral information in different steps comprising the entire hyperspectral image processing chain, from the moment the data is collected onboard the sensor to the time in which an application-oriented outcome is provided to end-users.

Specifically, the main contribution of this chapter is the development and validation of a set of highly innovative techniques for spatial-spectral analysis of hyperspectral image scenes in the following relevant areas:

1. **Dimensionality reduction.** A new method for spatial-spectral feature extraction, based on morphological concepts, is developed and compared to a standard spectral-based technique for dimensionality reduction, i.e., the principal component transform (PCT) [3].
2. **Unsupervised classification/segmentation.** A novel methodology for unsupervised image classification/segmentation is developed by extending the morphological watershed transformation (commonly used for unsupervised segmentation of grayscale images) to hyperspectral image processing. This technique is compared to a standard approach for unsupervised classification of multicomponent data, i.e., the ISODATA algorithm [3].
3. **Spectral mixture analysis.** In order to address the problem of mixed pixels, resulting from the limited spatial resolution and also from naturally-occurring mixing phenomena, we develop two separate contributions:
 - **Endmember extraction.** First, we develop a new iterative version of a previously developed morphological method for extraction of pure spectral signatures (called endmembers in hyperspectral analysis terminology).

The algorithm allows propagation of pure pixels between subsequent iterations, as opposed to the previous version of the algorithm. This algorithm is compared to other standard endmember extraction algorithms such as Pixel Purity Index (PPI) [14] or N-FINDR [15].

- **Unmixing.** Then, we develop a new spatial-spectral unmixing procedure based on the endmembers derived by the proposed extended morphological endmember extraction method. This technique is compared to other unmixing techniques in the literature such as unconstrained and fully constrained linear spectral unmixing [16], [4].
4. **Data compression.** In order to address the problems introduced by the extremely large volume of hyperspectral image data sets, we also develop a new spatial-spectral compression algorithm based on morphological concepts and compare the algorithms with standard techniques for data compression such as Set Partitioning in Hierarchical Trees (3D-SPIHT) [17] and JPEG 2000 multicomponent [18].
 5. **Parallel implementations.** Last but not least, we develop high performance computing implementations [19] for all the spatial-spectral algorithms introduced in this chapter and thoroughly assess their performance on the 568-processor Thunderhead Beowulf cluster of computers at NASA's Goddard Space Flight Center in Maryland. The detailed cross-validation of parallel hyperspectral processing algorithms in different application domains, conducted in this chapter using a massively parallel computing platform, may help image analysts in selection of parallel algorithms for specific applications.

The remainder of the chapter is structured as follows. Section 7.2 presents the proposed framework to extend mathematical morphology to hyperspectral images. Section 7.3 provides a framework for parallel implementation of morphological spatial-spectral algorithms for hyperspectral image processing. Section 7.4 develops a collection of parallel morphological algorithms for dimensionality reduction (P-MORPHDIM), watershed-based classification (P-WSHED), morphological endmember extraction (P-MORPHEE), morphological spectral unmixing (P-MORPHSU), and morphological data compression (P-MORPHCOMP). These algorithms have been designed to be efficiently run on commodity cluster computing platforms. Section 7.5 presents experimental results and comparisons to other standardized algorithms in the literature, using two well-known, publicly available hyperspectral scenes: the AVIRIS Indian Pines scene and the AVIRIS Cuprite scene, both widely used as benchmark data to validate classification and spectral unmixing accuracy in hyperspectral imaging applications. This section also includes performance results on the Thunderhead massively parallel cluster, along with a summary of the main observations and lessons learned after the detailed quantitative and comparative assessment of algorithms conducted in this chapter. Finally, Section 7.6 concludes with some remarks and hints at plausible future research.

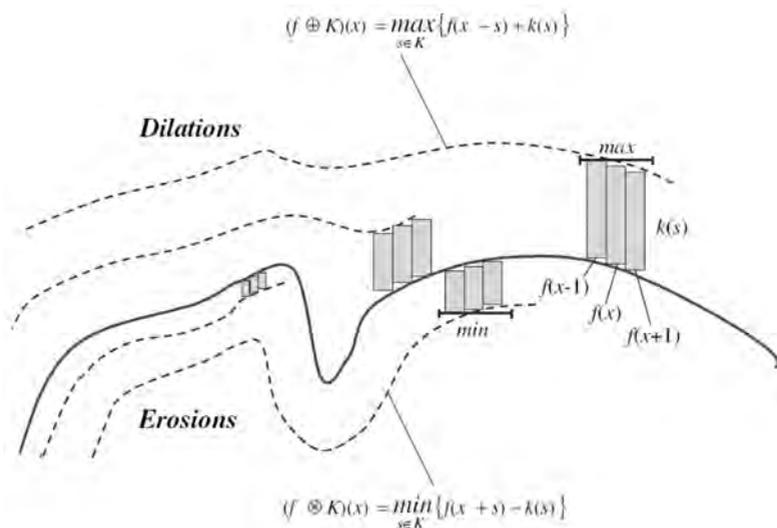


Fig. 7.2. Graphical interpretation of morphological erosion and dilation operations

7.2 Extended Mathematical Morphology

Mathematical morphology is a theory for spatial structure analysis that was established by introducing fundamental operators applied to two sets [13]. A set is processed by another one having a carefully selected shape and size, known as the structuring element (SE). The two basic operations of mathematical morphology are erosion and dilation. These operators can be graphically illustrated (in the context of greyscale morphology) by viewing the image data as an imaginary topographic relief in which the brighter the gray tone, the higher the corresponding elevation. With this assumption in mind, morphological operations can be interpreted as the result of sliding a SE over the topographical relief, so that the SE defines the new (dilated or eroded) scene based on its spatial properties such as height or width (see Fig. 7.2).

Extension of morphological operators to multichannel data such as hyperspectral imagery with hundreds of spectral channels is not straightforward. A simple approach consists in applying grayscale morphology techniques to each channel separately, an approach that has been called *marginal morphology* in the literature [20]. However, the marginal approach is often unacceptable in remote sensing applications because, when morphological techniques are applied independently to each image channel, analysis techniques are subject to the well-known problem of *false colors*; that is, it is very likely that new spectral constituents (not present in the original hyperspectral image) may be created as a result of processing the channels separately. An alternative (and perhaps more appropriate) way to approach the problem of multichannel morphology is to treat the data at each pixel as a vector [12]. Unfortunately, there is no unambiguous means of defining the minimum and maximum values between

two vectors of more than one dimension, and thus it is important to define an appropriate arrangement of vectors in the selected vector space.

In this chapter, we develop an application-driven vector ordering technique based on a spectral purity-based criterion [10], where each pixel vector is ordered according to its spectral distance to other neighboring pixel vectors in the N -dimensional data set \mathbf{f} . More specifically, we adopt a distance-based technique which utilizes a cumulative distance between one particular pixel vector $\mathbf{f}(x, y)$, where (x, y) indicates the spatial coordinates, and all the pixel vectors in the spatial neighborhood given by a SE denoted by K as follows [7]:

$$C_K(\mathbf{f}(x, y)) = \sum_{(s,t) \in K} \text{SAD}(\mathbf{f}(x, y), \mathbf{f}(s, t)), \quad (7.1)$$

where SAD is the spectral angle distance [4]. The SAD between two pixel vectors $\mathbf{f}(x, y)$ and $\mathbf{f}(s, t)$ is given by the following expression:

$$\text{SAD}(\mathbf{f}(x, y), \mathbf{f}(s, t)) = \cos^{-1} \left(\frac{\mathbf{f}(x, y) \cdot \mathbf{f}(s, t)}{\|\mathbf{f}(x, y)\| \cdot \|\mathbf{f}(s, t)\|} \right). \quad (7.2)$$

As a result, $C_K(\mathbf{f}(x, y))$ is given by the sum of SAD scores between $\mathbf{f}(x, y)$ and every other pixel vector in the K -neighborhood. At this point, we need to be able to define a maximum and a minimum given an arbitrary set of vectors $\mathbf{S} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\}$, where k is the number of vectors in the set. This can be done by computing $C_K(\mathbf{S}) = \{C_K(\mathbf{v}_1), C_K(\mathbf{v}_2), \dots, C_K(\mathbf{v}_k)\}$ and selecting \mathbf{v}_i such that $C_K(\mathbf{v}_i)$ is the minimum of $C_K(\mathbf{S})$, with $1 \leq i \leq k$. In similar fashion, we can select \mathbf{v}_j such that $C_K(\mathbf{v}_j)$ is the maximum of $C_K(\mathbf{S})$, with $1 \leq j \leq p$. Based on the definitions above, the extended erosion $\mathbf{f} \ominus K$ consists of selecting the K -neighborhood pixel vector that produces the minimum C_K value as follows [10]:

$$(\mathbf{f} \ominus K)(x, y) = \underset{(s,t) \in K}{\operatorname{argmin}} \{C_K(\mathbf{f}(x + s, y + t))\}. \quad (7.3)$$

On the other hand, the extended dilation $\mathbf{f} \oplus K$ selects the K -neighborhood pixel that produces the maximum value for C_K as follows [10]:

$$(\mathbf{f} \oplus K)(x, y) = \underset{(s,t) \in K}{\operatorname{argmax}} \{C_K(\mathbf{f}(x - s, y - t))\}. \quad (7.4)$$

For illustrative purposes, Fig. 7.3 shows a graphical representation of the performance of these two basic operators using a toy example in which a synthetic hyperspectral image is used for demonstration. As can be seen in Fig. 7.3, morphological dilation expands the spatial regions made up of pure pixel vectors in accordance with the spatial neighborhood defined by a 3×3 SE, while morphological erosion expands the regions made up of highly mixed pixel vectors in accordance with the same spatial neighborhood. In order to avoid changing the size and shape of the features in the image, a desirable feature for spatial filtering, extended morphological opening and closing operations have also been defined, respectively, as follows: $(\mathbf{f} \circ K)(x, y) = [(\mathbf{f} \ominus K) \oplus K](x, y)$, i.e., erosion followed by dilation, and $(\mathbf{f} \bullet K)(x, y) = [(\mathbf{f} \oplus K) \ominus K](x, y)$, i.e., dilation followed

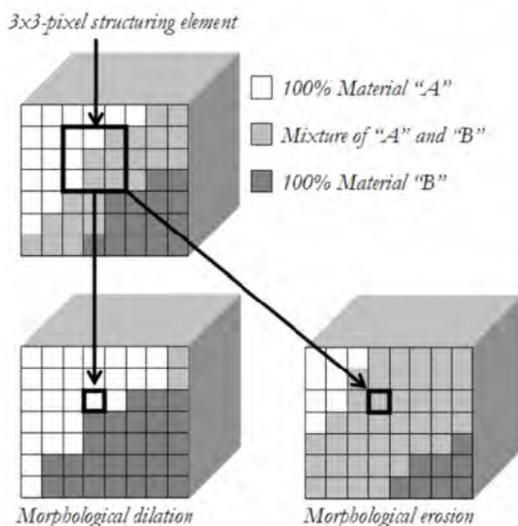


Fig. 7.3. Toy example showing the performance of extended morphological operations

by erosion [21]. With the above definitions in mind, the morphological gradient at $f(x, y)$ using K can be defined as:

$$G_K(f(x, y)) = \text{SAD}((f \oplus K)(x, y), (f \ominus K)(x, y)). \quad (7.5)$$

7.3 Parallel Implementation of Morphological Operations

This section discusses the development of parallel implementations for spatial-spectral hyperspectral image processing algorithms. An important consideration for this kind of algorithms is that, as shown by previous sections, these combined techniques make use of the information provided by the entire pixel vector *as a whole* and the spectral signature at a specific spatial location (x, y) is the minimum unit of information processed by the considered distance metric (i.e., the SAD distance). This source of information (spectral) is potentially complemented with that provided by other neighboring signatures to incorporate spatial context. The above consideration has a significant impact on the design of data partitioning strategies for parallelization purposes [22]. In particular, it has been shown in the literature that domain decomposition techniques provide flexibility and scalability in parallel multichannel image processing [23]. In parallel morphological processing for hyperspectral imaging, two types of partitioning can be exploited: spectral-domain partitioning and spatial-domain partitioning [24]:

1. **Spectral-domain partitioning** subdivides the volume into sub-volumes made up of contiguous spectral bands [see Fig. 7.4(a)], and assigns one or more sub-volumes to each processor. With this model, each pixel vector may be split amongst several processors and the communication cost associated

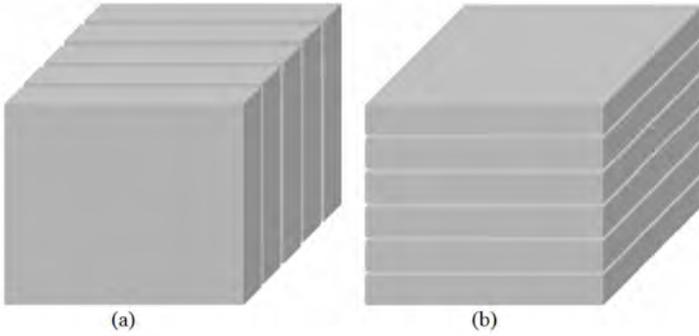


Fig. 7.4. (a) Spectral-domain, and (b) spatial-domain partitioning of a sample hyper-spectral image cube

to the computations based on spectral signatures would be increased [22]. In this option, the minimum amount of information to be processed in parallel corresponds to a portion of the entire spectral signature allocated to a given specific location (x, y) .

2. **Spatial-domain partitioning** subdivides the original data volume into slabs [see Fig. 7.4(b)] which are made up of (contiguous) pixel vector rows, thus retaining the full spectral information associated to each pixel vector at the same processor. In this option, the minimum amount of information to be processed in parallel corresponds to the entire spectral signature allocated to a given spatial location (x, y) .

In order to exploit parallelism as much as possible, we have adopted a standard master-slave parallel processing paradigm combined with spatial-domain partitioning for the parallel implementations developed in this chapter [25]. The master-slave paradigm has been shown in previous work to perform effectively in the context of parallel image processing applications. On the other hand, our selection of spatial-domain instead of spectral-domain data partitioning is due to the following major reasons:

- First and foremost, spatial-domain partitioning is a natural approach for low-level image processing as many image processing operations require the same function to be applied to a small set of elements around each entire pixel vector in the image volume.
- A second major reason for our decision of adopting a spatial-domain decomposition framework in our application is that, in spectral-domain partitioning, the calculations made for each pixel vector need to originate from several processors and thus require intensive inter-processor communication. This is generally perceived as a shortcoming for parallel design, because the overhead introduced by inter-processor communication would increase linearly with the increase in the number of processing elements, thus complicating the design of scalable parallel algorithms.

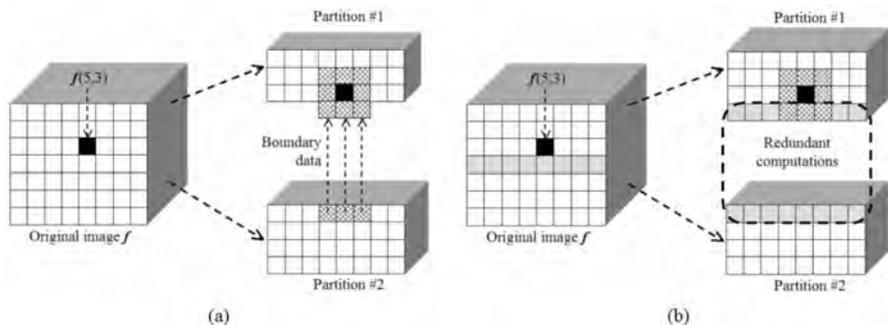


Fig. 7.5. (a) 3×3 SE computation split among two processing elements. (b) Introduction of an overlap border to minimize inter-processor communication in a 3×3 SE computation.

- A final major issue is code reusability; to reduce code redundancy and enhance portability, it is desirable to reuse much of the code for the sequential algorithm in the design of its correspondent parallel version and the spatial-domain decomposition approach greatly enhances code reuse.

A final consideration regarding hyperspectral data partitioning for morphological processing is noteworthy. In this type of processing, additional inter-processor communications will be required when the SE-based computation needs to be split amongst several different processing elements when the SE is centered around a pixel vector located in the border of one of the local partitions resulting after spatial-domain decomposition, as illustrated by Fig. 7.5(a), in which a small SE with 3×3 pixels in size is used for demonstration. In this case, the computations for the pixel vector at spatial coordinates $(5, 3)$ in the original image, denoted by $f(5, 3)$, will need to originate from two processing elements since this pixel becomes a border pixel after spatial-domain partitioning. As a result, a communication overhead involving three N -dimensional pixel vectors (located in partition #2) is required in order to complete the SE-based computation for the pixel vector $f(5, 3)$ in partition #1 (the same comment applies to all pixels in the third row of partition #1).

However, if an overlap border is carefully added to partition #1 (consisting of the entire first row of pixels allocated to partition #2), as illustrated in Fig. 7.5(b), then boundary data no longer need to be exchanged between neighboring processors in order to complete all calculations at partition #1, and therefore this data partition can be processed locally without the need for inter-processor communications. It is clear that such an overlap border would introduce redundant computations since the intersection between the two involved partitions would be non-empty. As a result, the solution above may be prohibitive for large SE sizes. Subsequently, for a given parallel platform, there is an application-dependent threshold to decide whether a redundant-information-based or data-exchange-based strategy should be adopted. This issue has been recently explored in [24]

concluding that, for small SE sizes, a redundant-computation strategy is generally preferred.

7.4 Parallel Spatial-Spectral Algorithms

This section describes several parallel algorithms for spatial/spectral processing of hyperspectral images. Four types of parallel algorithms are considered: 1) dimensionality reduction; 2) unsupervised classification; 3) spectral unmixing; and 4) data compression.

7.4.1 Parallel Dimensionality Reduction (P-MORPHDIM)

One of the most widely used dimension reduction techniques in remote sensing is the PCT [3], which computes orthogonal projections that maximize the amount of data variance, and yields a dataset in a new uncorrelated coordinate system. This rotational transform is characterized by its global nature and, therefore, it might not preserve all the information useful to obtain a good classification [11]. In addition, the PCT relies on spectral properties of the data alone, thus neglecting the information related to the spatial arrangement of the pixels in the scene.

In the following, we develop a new parallel algorithm for morphological feature extraction which integrates the spatial and spectral information in simultaneous fashion. To achieve this goal, we apply sequences of extended opening by reconstruction operations using SE's of varying width (called morphological profiles) [21]. This type of morphological sequences have been applied in the past to characterize image structures in grayscale remotely sensed image data [26]. In this chapter, we extend this concept to feature extraction from hyperspectral image data, with the goal to capture the spatial and spectral information around each pixel vector through the combination of spatial-spectral morphological operations for increasingly larger spatial neighborhoods. It is important to emphasize that the use of opening and closing operations is essential for spatial-spectral filtering [11]. With this type of operators, the image features are either completely retained or completely removed in accordance with the size and shape of the structuring element, thus allowing us to perform accurate image filtering based on spatial-spectral content. The inputs to the parallel algorithm, called P-MORPHDIM, are an N -dimensional hyperspectral image cube, \mathbf{f} ; a maximum number of filtering iterations, t , and a structuring element K with constant size of 3×3 pixels. The output is a transformed image cube, denoted by \mathbf{g} . The parallel algorithm is given by the following steps:

1. The master divides the original image cube \mathbf{f} into P spatial-domain partitions, where P is the number of processors in the system. In order to avoid excessive inter-processor communication during morphological processing, the partitions are formed using overlap borders, as described in Fig. 7.5(b).
2. Each worker performs spatial-spectral filtering on its local spatial-domain partition using the following sequence of steps:

- a) Compute an extended opening by reconstruction for each local pixel $\mathbf{f}(x, y)$ as $(\mathbf{f} \circ K)^t(x, y) = \min_{t \leq 1} \{\delta_K^t(\mathbf{f} \circ K|\mathbf{f})(x, y)\}$, with the basic operation $\delta_K^t(\mathbf{f} \circ K|\mathbf{f})(x, y) = \delta_B \delta_B \cdots \delta_B(\mathbf{f} \circ K|\mathbf{f})(x, y)$, i.e., δ_B is applied t times, and $\delta_B(\mathbf{f} \circ K|\mathbf{f})(x, y) = \min\{[(\mathbf{f} \circ K) \oplus K](x, y), \mathbf{f}(x, y)\}$.
 - b) Compute an extended closing by reconstruction for each local pixel $\mathbf{f}(x, y)$ as $(\mathbf{f} \bullet K)^t(x, y) = \min_{t \leq 1} \{\delta_K^t(\mathbf{f} \bullet K|\mathbf{f})(x, y)\}$, with the basic operation $\delta_K^t(\mathbf{f} \bullet K|\mathbf{f})(x, y) = \delta_B \delta_B \cdots \delta_B(\mathbf{f} \bullet K|\mathbf{f})(x, y)$, i.e., δ_B is applied t times, and $\delta_B(\mathbf{f} \bullet K|\mathbf{f})(x, y) = \min\{[(\mathbf{f} \bullet K) \ominus K](x, y), \mathbf{f}(x, y)\}$.
 - c) Compute the derivative of the extended opening profile as follows: $\mathbf{p}_t^\circ = \{\text{SAD}[(\mathbf{f} \circ K)^\lambda(x, y), (\mathbf{f} \circ K)^{\lambda-1}(x, y)]\}$, with $\lambda = \{1, 2, \dots, t\}$. Here, $\mathbf{f}(x, y) = (\mathbf{f} \circ K)^0(x, y)$ for $\lambda = 0$ by the definition of extended opening by reconstruction.
 - d) Compute the derivative of the extended closing profile as shown below: $\mathbf{p}_t^\bullet = \{\text{SAD}[(\mathbf{f} \bullet K)^\lambda(x, y), (\mathbf{f} \bullet K)^{\lambda-1}(x, y)]\}$, with $\lambda = \{1, 2, \dots, t\}$. Here, $\mathbf{f}(x, y) = (\mathbf{f} \bullet K)^0(x, y)$ for $\lambda = 0$ by the definition of extended closing by reconstruction.
 - e) Form a $(2t - 1)$ -dimensional morphological profile for each local pixel $\mathbf{f}(x, y)$ by combining the derivatives of the extended opening and closing profiles as follows: $\text{MP}(x, y) = \{\mathbf{p}_t^\circ(x, y), \mathbf{p}_t^\bullet(x, y)\}$. The resulting morphological profile can be seen as a spatial-spectral feature vector on which a subsequent classification procedure may be applied.
3. The master processor gathers the individual $(2t - 1)$ -dimensional profiles provided by the workers and merges them into a new data cube \mathbf{g} with $2t - 1$ components. As shown by the parallel description above, this approach requires minimal coordination between the master and the workers, namely, at the beginning and ending of the parallel process, although it is subject to a redundant computation overhead introduced by the overlap borders used by the proposed data partitioning strategy.

7.4.2 Parallel Unsupervised Classification (P-WSHED)

This subsection provides a new parallel algorithm for unsupervised classification of hyperspectral images based on the morphological watershed transformation [27], originally introduced for grayscale images. This technique relies on a marker-controlled approach that considers the grayscale image as imaginary topographic relief in which, the brighter the intensity, the higher the corresponding elevation (see Fig. 7.2). Let us assume that a drop of water falls on such a topographic surface. The drop will flow down along the steepest slope path until it reaches a minimum. The set of points of the surface whose steepest slope path reach a given minimum constitutes the catchment basin associated with that minimum, while the watersheds are the zones dividing adjacent catchment basins [12]. Despite its encouraging results in several applications, the watershed transformation has not been fully exploited in multichannel image analysis, mainly due to the lack of well-defined vector ordering strategies required by basic morphological operations. In this chapter, we develop a multichannel watershed classification algorithm which consists of three main stages:

1. **Minima selection.** In order to select ‘markers’ or minima from which the watershed transform is started, we hierarchically order all minima according to their deepness, and then select only those above a threshold. The deepness of a basin is the level the water would reach, coming in through the minimum of the basin, before the water would overflow into a neighbor basin. Deepness can be computed using morphological reconstruction [12] applied to the multichannel gradient (reconstruction is a class of morphological transformation that does not introduce discontinuities). Given the multichannel gradient $G_K(\mathbf{f})$ of an N -dimensional image \mathbf{f} processed using a SE denoted by K , the morphological reconstruction of $G_K(\mathbf{f})$ from $G_K(\mathbf{f}) \ominus K$ has a watershed transform in which the regions with deepness lower than a certain value v have been joined to the neighbor region with closer spectral properties.
2. **Flooding.** Let the set $\mathbf{S} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\}$ denote the set of p minimum pixel vectors resulting from minima selection. Let the catchment basin associated with a minimum pixel \mathbf{p}_i be denoted by $\text{CB}(\mathbf{p}_i)$. The catchment basins are progressively created by simulating the flooding process. The first pixel vectors reached by water are the points of highest deepness score. From now on, the water either expands the region of the catchment basin already reached by water, or starts to flood the catchment basin whose minima have a deepness equal to $C_K(\mathbf{v}_i)$, where \mathbf{v}_i is the deepest pixel in $\mathbf{S} - \{\mathbf{v}_j\}$. This operation is repeated until $\mathbf{S} = \emptyset$.
3. **Region merging.** To obtain the final classification, some of the regions $\{\text{CB}(\mathbf{v}_i)\}_{i=1}^p$ resulting from the watershed can be merged to reduce the number of regions. First, all regions are ordered into a region adjacency graph (RAG) [28]. Each edge in the RAG is assigned a weight, so that the weight of an edge $e(\mathbf{v}_i, \mathbf{v}_j)$ is the value of $\text{SAD}(\mathbf{v}_i, \mathbf{v}_j)$. Subsequently, regions $\text{CB}(\mathbf{v}_i)$ and $\text{CB}(\mathbf{v}_j)$ can be merged attending to spatial properties in the case of adjacent regions, and also according to pixel vector similarity criteria for non-adjacent regions. The output of the algorithm above is a 2-dimensional classification map based on the set of $\mathbf{S} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\}$ class prototypes.

Our parallel implementation of the multichannel watershed algorithm above, called P-WSHED, uses a simple master-slave model [29], in which the master processor reads the whole multichannel image \mathbf{f} and divides it into a set of P spatial-domain partitions, which are sent to different processors. The slave processors run the watershed algorithm on the respective partitions and also exchange data among themselves for uniform classification. After the classified regions become stable, the slaves send the output to the master, which combines all of them in a proper way and provides the final classification. If we assume that the parallel system has P processors available, then one of the processors is reserved to act as the master, while each of the $P - 1$ remaining processors create a local queue Q_i with $1 \leq i \leq P - 1$. The minima selection algorithm is run locally at each processor to obtain a set of minima pixels surrounded by non-minima, which are then used to initialize each queue Q_i . Flooding is then performed locally in each processor as in the serial algorithm.

It should be noted, however, that due to the image division, flooding is confined only to the local subdomain. There may exist parts of the subimage that cannot be reached by flooding since they are contained in other subimages. Our approach to deal with this problem is to first flood locally at every deepness score in the subimage. Once the local flooding is finished, each processor exchanges classification labels of pixels in the boundary with appropriate neighboring processors. Subsequently, a processor may receive classification labels corresponding to pixels in the extended subdomain. The processor must now *reflood* the local subdomain from those pixels, a procedure that may introduce changes in classification labels at the local subdomain. Communication and reflooding are again repeated until stabilization (i.e. no more changes occur).

7.4.3 Parallel Spectral Unmixing

While our focus in previous subsections has been on classification approaches, spectral unmixing has been an alluring exploitation goal since the earliest days of hyperspectral imaging [16]. No matter the spatial resolution, in natural environments the spectral signature for a nominal pixel is invariably a mixture of the signatures of the various materials found within the spatial extent of the ground instantaneous field view. In hyperspectral imagery, the number of spectral bands usually exceeds the number of pure spectral components, called endmembers in hyperspectral analysis terminology [30], and the unmixing problem is cast in terms of an over-determined system of equations in which, given the correct set of endmembers allows determination of the actual endmember abundance fractions through a numerical inversion process. Since each observed spectral signal is the result of an actual mixing process, the driving abundances must obey two constraints [4]. First, all abundances must be non-negative. Second, the sum of abundances for a given pixel must be unity. However, it is the derivation and validation of the correct suite of endmembers that has remained a challenging and goal for the past years [31], [32]. In the following subsections, we develop two novel parallel algorithms for endmember extraction and fractional abundance estimation in hyperspectral images which introduce an innovative component with regards to standard techniques in the literature: the incorporation of spatial information to the (spectrally-guided) problems of finding endmembers and approximating their abundance fractions in mixed pixels.

Parallel endmember extraction (P-MORPHEE)

Most available endmember extraction approaches, including popular and successful algorithms such as the PPI (available in Research Systems ENVI software) [14] or the N-FINDR (distributed by Technical Research Associates, Inc.) [15] have been designed from a spectroscopic viewpoint and, thus, tend to neglect the existing spatial correlation between pixels. In the following, we develop a novel parallel algorithm which integrates the spatial and spectral information in the endmember searching process. The algorithm, based on our previously developed automated morphological endmember extraction (AMEE) algorithm

[10], allows propagation of pure pixels between subsequent iterations, as opposed to the previous version of the algorithm. The inputs to the parallel algorithm, called P-MORPHEE, are the full hyperspectral data cube \mathbf{f} , a structuring element K , a maximum number of algorithm iterations I_{max} , and a number of endmembers to be extracted, p . The output is an endmember set, $\{\mathbf{e}_i\}_{i=1}^q$, with $q \leq p$. A step-by-step description of the algorithm follows:

1. The master divides the original image cube \mathbf{f} into P spatial-domain partitions, where P is the number of processors in the system. In order to avoid excessive inter-processor communication during morphological processing, the partitions are formed using overlap borders, as described in Fig. 7.5(b).
2. Each worker performs an iterative spatial-spectral endmember extraction process on its local spatial-domain partition using the following sequence of steps:
 - a) Set $i = 1$ and initialize a morphological eccentricity index [10], denoted by $MEI(x, y) = 0$, for each pixel $\mathbf{f}(x, y)$ in the local partition.
 - b) Move K through all the pixels of the local partition data, defining a local spatial search area around each pixel $\mathbf{f}(x, y)$, and calculate the maximum and minimum pixel vectors at each K -neighborhood using extended morphological erosion and dilation. Then, update the MEI at each spatial location (x, y) using the following expression:

$$MEI(x, y) = SAD[(\mathbf{f} \ominus K)(x, y), (\mathbf{f} \oplus K)(x, y)] \quad (7.6)$$
 - c) Set $i = i + 1$. If $i = I_{max}$, then return the MEI scores for all the pixels in the local partition to the master processor. Otherwise, replace the local partition with its dilation using K . This represents an optimization of the algorithm that propagates only the purest pixels at the local neighborhood to the following algorithm iteration. Then, go to step 2.
3. The master gathers the individual MEI scores provided by the workers and selects the set of p pixel vectors with higher associated MEI values (called endmember candidates). Then, the master forms a unique spectral set of $\{\mathbf{e}_i\}_{i=1}^q$ pixels, with $q \leq p$, by calculating the SAD for all pixel vector pairs.

Parallel fractional abundance estimation (P-MORPHSU)

To conclude this subsection on parallel spectral unmixing methods we outline a new parallel algorithm for fractional abundance estimation of a set of input endmembers. This method integrates the spatial and the spectral information by considering a spatial neighborhood (defined by a morphological SE denoted by K) around each mixed pixel. This method is similar to traditional approaches, in the sense that it makes use of the standard fully constrained least squares technique [4] to estimate abundance fractions. But it differs from traditional methods in the fact that the endmember set used for each pixel is adaptively

calculated based on the spatial context. The inputs to the parallel method are the full hyperspectral data cube \mathbf{f} , a structuring element K , a tolerance threshold t_{SU} , and a set of endmembers $\{\mathbf{e}_i\}_{i=1}^q$. The output is an abundance fraction estimation for each endmember in each pixel of the input data set. The parallel algorithm, called P-MORPHSU, is based on the following steps:

1. The master divides the original image cube \mathbf{f} into P spatial-domain partitions, using overlap borders as described in Fig. 7.5(b). The master also broadcasts the endmember set $\{\mathbf{e}_i\}_{i=1}^q$ to all the workers.
2. Each worker estimates the fractional abundance of each endmember in each pixel of its local partition using the following procedure:
 - a) Before unmixing a certain local pixel, say $\mathbf{f}(x, y)$, a weight is assigned to the pixels in the K -neighborhood centered at spatial coordinates (x, y) . This is done by first calculating, for each pixel in the K -neighborhood, the SAD distance to each one of the endmembers in the set $\{\mathbf{e}_i\}_{i=1}^q$, and labeling the pixel as an instance of a certain endmember (candidate) by using the minimum SAD score.
 - b) Then, a weight is assigned to each endmember candidate (the weight is inversely proportional to the minimum SAD score reported for that candidate).
 - c) Finally, all endmember candidates in the K -neighborhood are sorted by weight, and only those with associated weights above tolerance threshold t_{SU} are incorporated to the *local* endmember set which is finally used to unmix the pixel $\mathbf{f}(x, y)$ using standard FCLSU. In other words, the abundance estimation is still performed by using a fully constrained least squares technique [4], but the actual composition of the endmember set used to perform the least squares estimation may vary (for each particular pixel) depending on the spatial-spectral context around the pixel, as opposed to the traditional approach, in which the entire set of spectral endmembers is always used to unmix each pixel. Let us assume that the set of endmembers, obtained by the above procedure, is denoted by $\{\mathbf{e}_i\}_{i=1}^l$, with $1 \leq l \leq q$. The goal is to achieve a decomposition of the pixel $\mathbf{f}(x, y)$ using the set of l endmembers above as follows:

$$\mathbf{f}(x, y) = \mathbf{e}_1 \cdot a_1(x, y) + \mathbf{e}_2 \cdot a_2(x, y) + \cdots + \mathbf{e}_l \cdot a_l(x, y). \quad (7.7)$$

To achieve this goal, the pixel is multiplied by $(\mathbf{M}^T \mathbf{M})^{-1} \mathbf{M}^T$, where $\mathbf{M} = \{\mathbf{e}_i\}_{i=1}^l$ and the superscript 'T' denotes the matrix transpose operation. In the expression above, abundance sum-to-one and non-negativity constraints are imposed, i.e., $\sum_{i=1}^l a_i(x, y) = 1$ and $a_i(x, y) \geq 0$ for all (x, y) .

3. The master gathers the individual fractional abundance estimations provided by the workers and merges them together to generate a set of endmember fractional abundance estimations for all the pixels in the original hyperspectral image.

7.4.4 Parallel Data Compression (P-MORPHCOMP)

In this subsection, we describe a lossy compression algorithm for hyperspectral imagery which relies on the concept of spectral unmixing introduced in the previous subsection. The main advantage of the proposed technique with regards to other standard compression algorithms in the literature, such as 3D-SPIHT or JPEG 2000 multicomponent, is the ability to deal with mixed pixels and sub-pixel targets [33]. A mixed pixel is a mixture of two or more different substances present in the same pixel, as outlined before in this chapter. A subpixel target is a mixed pixel with size smaller than the available pixel size (spatial resolution). When hyperspectral image compression is performed, it is critical and crucial to take into account these two issues, which have been generally overlooked in the development of lossy compression techniques in the literature. The idea of the proposed data compression algorithm is to represent each pixel vector $\mathbf{f}(x, y)$ with N spectral bands by a set of $l \leq N$ fractional abundance estimations corresponding to the set of $\{\mathbf{e}_j(x, y)\}_{j=1}^l$ endmembers that contribute to the mixture in that pixel [34]. More precisely, for each N -dimensional pixel vector $\mathbf{f}(x, y)$, its associated l -dimensional abundance vector $\mathbf{a}(x, y) = a_1(x, y), a_2(x, y), \dots, a_l(x, y)$, estimated using the P-MORPHSU algorithm, is used as a fingerprint of $\mathbf{f}(x, y)$ with regards to l endmembers obtained as a subset of the full endmember set produced by the P-MORPHEE algorithm, i.e., $\{\mathbf{e}_j(x, y)\}_{j=1}^l \subseteq \{\mathbf{e}_i\}_{i=1}^q$. The proposed parallel data compression algorithm receives as input parameters a hyperspectral image cube, \mathbf{f} , and a maximum number of endmembers to be retained per pixel, q , and can be summarized by the following steps:

1. Use the P-MORPHEE parallel algorithm to adaptively obtain a set of q endmembers, $\{\mathbf{e}_i\}_{i=1}^q$ from the input hyperspectral image \mathbf{f} .
2. Use the P-MORPHSU parallel algorithm to adaptively estimate the corresponding abundance fractions $\mathbf{a}(x, y) = a_1(x, y), a_2(x, y), \dots, a_l(x, y)$ of a subset of l endmembers, with $\{\mathbf{e}_j(x, y)\}_{j=1}^l \subseteq \{\mathbf{e}_i\}_{i=1}^q$, for each pixel vector in the input scene, thus approximating the pixel signature by the following expression: $\mathbf{f}(x, y) = \mathbf{e}_1 \cdot a_1(x, y), \mathbf{e}_2 \cdot a_2(x, y), \dots, \mathbf{e}_l \cdot a_l(x, y)$. Note that this is a reconstruction of the spectral signature at $\mathbf{f}(x, y)$.
3. The master processor applies lossless predictive coding to reduce spatial redundancy within each of the fractional abundance estimates for each pixel in the input scene, using Huffman coding to encode predictive errors [34].

7.5 Experimental Results

This section provides an assessment of the effectiveness of the proposed parallel algorithms in the task of providing significant performance gains, without loss of accuracy, in the analysis of real hyperspectral data sets. Two different application domains: 1) land-cover classification in agriculture; and 2) mapping of geological features, are used to provide a detailed cross-validation of parallel algorithms for hyperspectral image analysis, addressing the current need for

application-oriented inter-comparisons of parallel in this emerging and fast growing research area. The section is organized as follows. First, we provide a brief outline of Thunderhead, a Beowulf cluster available at NASA's Goddard Space Flight Center that has been used as our baseline parallel platform. Then, we provide an overview of the hyperspectral image data sets used in this study. A detailed computational cost-performance analysis of the parallel algorithms in the context of two highly representative application domains follows. The section concludes with a summary and detailed discussion on the results obtained.

7.5.1 Parallel Computing Platform

The parallel computing platform used for experimental validation in this work is the Thunderhead system at NASA's Goddard Space Flight Center in Maryland. This Beowulf cluster can be seen as an evolution of the HIVE (Highly Parallel Virtual Environment) project, started in spring of 1997 to build a commodity cluster intended to be exploited by different users in a wide range of scientific applications. The idea was to have workstations distributed among many offices and a large number of compute nodes (the compute core) concentrated in one area. The workstations would share the compute core as though it was apart of each. Thunderhead is currently composed of 268 dual 2.4 Ghz Intel 4 Xeon nodes, each with 1 GB of memory and 80 GB of hard disk (see <http://thunderhead.gsfc.nasa.gov> for additional details). The total disk space available in the system is 21.44 Tbyte, and the theoretical peak performance of the system is 2.5728 Tflops (1.2 Tflops on the Linpack benchmark). The current estimated cost of the Thunderhead system is 1.25M U.S. dollars. Along with the 568-processor computer core (out of which 256 were used for experiments), Thunderhead has several nodes attached to the core with Myrinet 2000 connectivity. Our parallel algorithms were run from one of such nodes, called thunder1. The operating system is Linux Fedora Core, and MPICH was the message-passing library used.

7.5.2 Hyperspectral Data Sets

Two different hyperspectral data sets collected by the NASA's Jet Propulsion Laboratory AVIRIS instrument have been selected for experimental validation in this study. All the considered scenes have been geometrically corrected by JPL and have extensive ground-truth information available, thus allowing us to validate the performance of parallel algorithms in several different application domains.

AVIRIS Hyperspectral Data over Indiana's Indian Pines Region

This scene was gathered by AVIRIS over the Indian Pines test site in North-western Indiana, a mixed agricultural/forested area, early in the growing season, and consists of 1939×677 pixels and 224 spectral bands in the wavelength range 0.4–2.5 μm (574 MB in size). The AVIRIS Indian Pines data set represents a very challenging classification problem dominated by similar spectral classes and

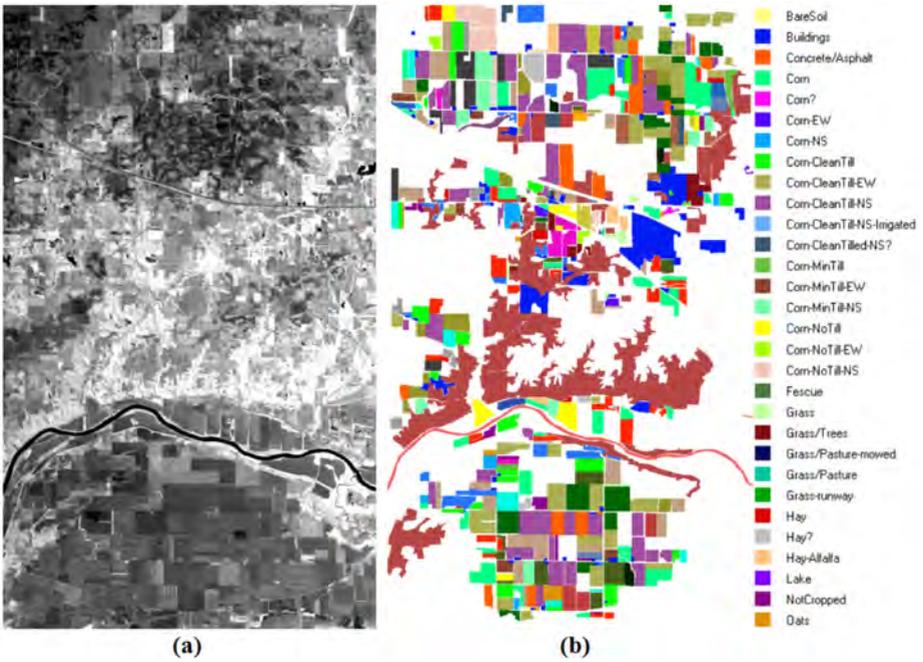


Fig. 7.6. (a) Spectral band at 587 nm wavelength of an AVIRIS scene comprising agricultural and forest features at Indian Pines region. (b) Ground-truth map with 30 mutually-exclusive land-cover classes.

mixed pixels. Specifically, the primary crops of the area, mainly corn and soybeans, were very early in their growth cycle with only about 5% canopy cover. This fact makes most of the scene pixels highly mixed in nature. Discriminating among the major crops under this circumstances can be very difficult, a fact that has made this scene an extensively used benchmark to validate classification accuracy of hyperspectral imaging algorithms. For illustrative purposes, Fig. 7.6(a) shows the spectral band at 587 nm of the original scene and Fig. 7.6(b) shows the corresponding ground-truth map, displayed in the form of a class assignment for each labeled pixel, with 30 mutually exclusive ground-truth classes. Part of these data, including ground-truth, are available online from Purdue University (for details, see <http://cobweb.ecn.purdue.edu/biehl/MultiSpec/>).

AVIRIS Hyperspectral Data over Cuprite Mining District, Nevada

Another AVIRIS scene collected over the Cuprite mining district in Nevada was also used in experiments to evaluate the proposed parallel algorithms in the context of a mineral mapping application. The data set (available from <http://aviris.jpl.nasa.gov/html/aviris.freedata.html>) consists of 614×512 pixels and 224 bands in the wavelength range $0.4\text{--}2.5 \mu\text{m}$ (137 MB in size). As opposed to the previously described Indian Pines data set, the Cuprite data set

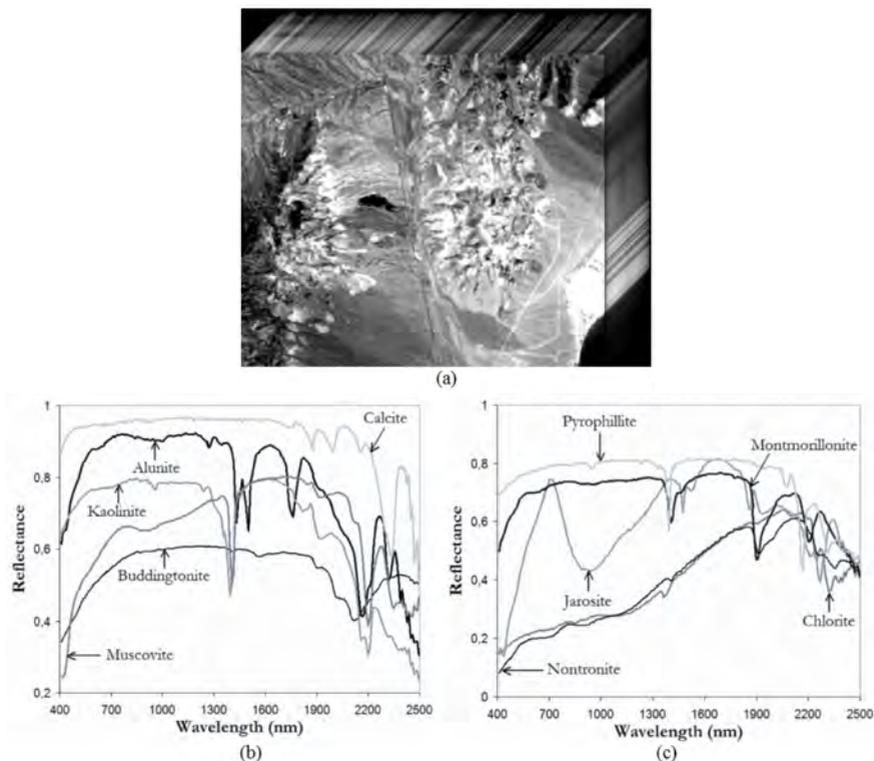


Fig. 7.7. (a) AVIRIS scene over Cuprite mining district. (b-c) Ground-truth mineral spectra provided by USGS.

is atmospherically corrected and available in reflectance units (not in at-sensor radiance), thus allowing direct comparison of pixel vectors to ground spectral signatures. The Cuprite site has been extensively mapped by the U.S. Geological Survey (USGS) in the last twenty years, and there is extensive ground-truth information available, including a library of mineral signatures collected on the field (see <http://speclab.cr.usgs.gov/spectral-lib.html>). Fig. 7.7(a) shows the spectral band at 587 nm wavelength of the AVIRIS scene. The spectra of USGS ground minerals: alunite, buddingtonite, calcite, kaolinite, muscovite [Fig. 7.7(b)], chlorite, jarosite, montmorillonite, nontronite, pyrophyllite [Fig. 7.7(c)] are also displayed. These selected spectral signatures will be used in this work to evaluate endmember extraction accuracy.

7.5.3 Experimental Assessment of Parallel Algorithms

Experiment 1: Land-Cover Classification in Agriculture

In this subsection, we discuss the performance of the proposed parallel algorithms in the context of an agricultural classification application. In order to validate

the experimental results provided by those algorithms, we resort to ground-truth measurements collected by Purdue University over the Indian Pines region [see Fig. 7.6(b)]. In this experiment, we assessed the feature extraction performance of the proposed P-MORPH dimensionality reduction technique in comparison with that of a parallel PCT (P-PCT) algorithm proposed in [35]. Also, we evaluated the classification performance of the proposed P-WSHED technique in comparison with a parallel version of the ISODATA algorithm (P-ISODATA) [36]. It should be noted that ISODATA is widely regarded as a benchmark for validation of unsupervised classification algorithms [3]. Specifically, the following experiments were conducted:

1. First, we run the P-ISODATA and the P-WSHED on the original N -dimensional hyperspectral data cube.
2. Then, we run P-ISODATA and P-WSHED on a q -dimensional, reduced data set, obtained after applying the P-PCT algorithm to the original data cube. In this experiment, we set $q = 41$ and thus retained the first 41 principal components. This decision was based on our estimation of the number of distinct signal sources in the data using the virtual dimensionality (VD) concept [37]. The fact that the VD estimated more than 30 classes for the AVIRIS Indian Pines scene was not surprising, since Fig. 7.6(b) reveals that the 30 available ground-truth classes only cover about 60% of the entire scene.
3. Finally, we run the P-ISODATA and P-WSHED on a q -dimensional, reduced data set obtained after applying the P-MORPH algorithm (with $t = 21$, resulting in $2t - 1 = 41$ components after morphological, spatial-spectral dimensionality reduction).

In all cases, we carefully adjusted algorithm parameters based on our experience with those algorithms in different application domains [22], [30]. With the above experimental settings in mind, Table 7.1 reports the overall and individual classification accuracies (with regards to some selected ground-truth classes) produced by the parallel algorithms after using different algorithm configurations (it should be noted that the overall accuracies refer to the entire set of 30 ground-truth classes available, not displayed here for space considerations). As shown by Table 7.1, when P-ISODATA and P-WSHED were combined with P-MORPH for dimensionality reduction, the classification accuracies increased significantly with respect to the cases in which the same algorithms were applied to the original image, or to a reduced version of the image using P-PCT. Overall, the P-WSHED provided significantly better classification results than P-ISODATA, with the combination P-MORPH+P-WSHED clearly resulting in the highest individual and overall classification accuracies. This indicates that the incorporation of spatial information in both feature extraction and spectral classification may offer important advantages, in particular, when the spatial information is an added value for the analysis, as it seems to be the case in the considered application environment given the spatially correlated distribution of land-cover classes in the Indian Pines region.

Table 7.1. Individual (per-class) and overall percentages of correctly classified pixels in the AVIRIS Indian Pines scene by different combinations of parallel algorithms. Sequential execution times (in seconds) measured in a single node of NASA’s Thunderhead cluster are reported for each algorithm combination.

	(Original image)		P-PCT+		P-MORPH+	
	P-ISODATA	P-WSHED	P-ISODATA	P-WSHED	P-ISODATA	P-WSHED
Bare soil	60.32	71.48	62.34	72.71	77.81	87.34
Buildings	52.45	64.56	52.89	73.28	65.23	79.12
Concrete	58.33	69.95	61.12	62.01	75.94	85.99
Corn	51.23	63.56	54.27	63.48	67.01	80.31
Fescue	56.03	67.39	55.03	64.34	69.23	79.98
Grass	62.37	74.01	60.48	72.90	73.91	86.00
Hay	55.15	67.03	53.49	65.23	72.16	85.43
Lake	58.23	69.50	59.25	73.42	73.45	87.22
Oats	53.12	62.89	56.78	66.58	69.32	83.91
Overall accuracy	55.48	66.17	57.21	68.56	72.26	84.58
Processing time	49912	60356	48007	50096	50221	52310

For illustrative purposes, Table 7.1 also reports the execution times (in seconds) measured for the different algorithms executed on a single processor of the Thunderhead system. We emphasize that these times correspond to real sequential versions of the considered algorithms, not to parallel versions using one master and one worker. As shown by the table, the two considered dimensionality reduction techniques were computationally expensive but led to significant reductions in the processing times of P-ISODATA and P-WSHED algorithms. We note that, in this example, the number of bands was reduced from more than 200 to 41 according to the VD concept. In the case of P-MORPH, the spatial-spectral feature extraction accomplished by the algorithm always helped increase the classification scores despite the significant reduction in the number of bands, thus indicating that this approach was able to retain the information relevant to the separation of the classes in all considered experiments.

To empirically investigate the scaling properties of the considered parallel algorithms, Fig. 7.8(a) plots their speedup factors as a function of the number of available processors on Thunderhead. Results in Fig. 7.8(a) reveal that the performance drop from linear speedup in both P-PCT and P-ISODATA algorithms increases significantly as the number of processors increase. This is due to the data dependencies and sequential calculations involved in those algorithms. A similar effect was observed for the P-WSHED algorithm, in which the region-growing process at each local partition may result in different processing times, thus affecting the overall scalability of the algorithm. On the other hand, we can observe in Fig. 7.8(a) that the P-MORPH algorithm was clearly the algorithm which achieved better scalability on Thunderhead. This result comes as no surprise since P-MORPH was identified as *pleasantly parallel* (despite the use of overlap borders in the data partitioning procedure), and involves very few data dependencies. This observation also resulted in faster execution times for the parallel algorithm combinations using P-MORPH for dimensionality reduction, as reported in Fig. 7.8(b). For illustrative purposes, Table 7.2 displays the execution times in seconds measured for different processors in this application case study.

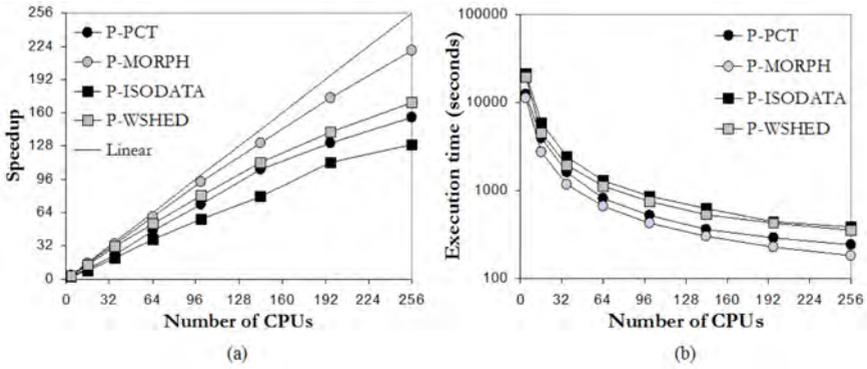


Fig. 7.8. Performance results after processing the AVIRIS Indian Pines scene with the proposed parallel algorithms on Thunderhead: (a) Parallel algorithm scalability; (b) Graphical representation of execution times as a function of the number of CPUs

Table 7.2. Execution times (in seconds) measured on Thunderhead for the parallel algorithms using the AVIRIS Indian Pines scene

CPUs	1	4	16	36	64	100	144	196	256
P-PCT	38025	12467	3978	1622	815	527	361	290	245
P-MORPH	40239	11178	2728	1183	668	430	307	231	183
P-ISODATA	49912	21330	5907	2428	1299	865	630	444	386
P-WSHED	60356	19345	4569	1933	1129	752	536	426	355

Summarizing, experimental results in this subsection reveal that significant improvements can be obtained, both from the viewpoint of classification accuracy and computational performance, by using spatial-spectral parallel algorithms in the framework of our land-cover classification application. For instance, the combination of P-MORPH for dimensionality reduction followed by P-WSHED for classification was able to provide a relatively accurate result (taking into account the high complexity of the scene and the unsupervised nature of the algorithm) in only 9 minutes using 256 processors, as opposed to several hours of computation required by the sequential version of the same algorithm combination executed on a single Thunderhead processor. Despite these encouraging results, further experimentation in other application areas is required to rigorously establish the statistics obtained in this case study.

Experiment 2: Mapping of Geological Features

In this second experiment, we have conducted a cross-validation of parallel spectral unmixing algorithms in the context of a mineral mapping application, using the well-known AVIRIS Cuprite data set for demonstration purposes. It should be noted that ground-truth information for this scene is only available in the form of a collection of USGS mineral signatures, and therefore the parallel algorithms cannot be evaluated in terms of their classification accuracy as in

the previous examples. For comparative purposes, we used the parallel versions of PPI (P-PPI) and N-FINDR (P-FINDR) developed in previous work [25] to evaluate the performance of the proposed P-MORPHEE endmember extraction algorithm. Specifically, our experimentation in this subsection comprised the following steps:

1. First, we run P-PPI and P-FINDR using their original configurations, i.e., using a dimension reduction technique (the P-PCT in our experiments) to reduce the dimensionality of the input data from N to q (with $q = 15$), obtaining a set of 15 spectral endmembers in both cases. This value was obtained using the VD concept in [37]. In order to establish a fair comparison of our P-MORPHEE algorithm with the above two parallel algorithms, we also used a reduced version of the original data cube (obtained by the P-PCT in [35]) when running our algorithm.
2. Then, we repeated the previous experiment but this time using P-MORPH instead of P-PCT to perform feature extraction from the input hyperspectral scene. Here, we used $t = 8$, resulting in 15 components, which is consistent with the dimensionality estimation provided by the VD concept.

Table 7.3 shows the SAD values between the endmembers in the final endmember set (extracted by different combinations of a parallel dimensionality reduction algorithm followed by a parallel endmember extraction algorithm) and the corresponding spectral signatures in the USGS library. In order to display the results in a more effective manner, we only report the SAD score associated to the most similar spectral endmember (out of 15 endmembers obtained for each algorithm combination) with regards to its corresponding USGS signature. It is important to emphasize that smaller SAD values indicate higher spectral similarity. As shown by Table 7.3, the P-PCT+P-FINDR combination resulted in the largest number of minimal SAD values (displayed in bold typeface in the table) among all considered combinations. Quite opposite, all the combinations which used P-MORPH for feature extraction prior to endmember extraction generally produced endmembers which were less similar, spectrally, with regards to reference USGS signatures. This is indeed a very interesting result, which indicates that spectral information is more important than spatial information in this particular application case study and, specifically, in the feature extraction task prior to endmember extraction. This results from the fact that geological features in the Cuprite mining district appear quite scattered, thus exhibiting little spatial correlation. Therefore, in this example spectral information is more important than spatial information in order to discriminate between subtle mineral signatures in this application. As a result, it is not surprising that the performance of P-PCT in this example was better than that exhibited by P-MORPH, which is particularly tuned for the integration of spatial and spectral information.

We would like to emphasize that the proposed combinations of parallel algorithms have also been evaluated from the viewpoint of their capacity to produce high-quality abundance estimations for geological features in the Cuprite mining district. This has been done by estimating the fractional abundance of

Table 7.3. SAD-based spectral similarity scores between the USGS mineral spectra and their corresponding endmember pixels produced by several combinations of a parallel dimensionality reduction algorithm followed by a parallel endmember extraction algorithm (the most similar endmember for each mineral is shown in bold typeface). Sequential execution times (in seconds), measured in a single node of NASA’s Thunderhead cluster, are also given.

	P-PCT+			P-MORPH+		
	P-PPI	P-FINDR	P-MORPHEE	P-PPI	P-FINDR	P-MORPHEE
Alunite	0.084	0.081	0.084	0.106	0.103	0.095
Buddingtonite	0.106	0.084	0.094	0.122	0.109	0.108
Calcite	0.105	0.105	0.110	0.120	0.112	0.110
Kaolinite	0.125	0.136	0.136	0.144	0.136	0.131
Muscovite	0.136	0.136	0.136	0.153	0.150	0.145
Chlorite	0.112	0.102	0.108	0.135	0.118	0.116
Jarosite	0.106	0.089	0.096	0.122	0.115	0.109
Montmorillonite	0.108	0.094	0.106	0.126	0.119	0.117
Nontronite	0.102	0.099	0.099	0.124	0.120	0.113
Pyrophyllite	0.094	0.090	0.090	0.115	0.112	0.105
Processing time	12251	10201	10573	12680	10630	11002

Table 7.4. Execution times (in seconds) measured on Thunderhead for the parallel algorithms using the AVIRIS Cuprite scene

CPUss	1	4	16	36	64	100	144	196	256
P-PCT	9506	3117	994	405	204	132	90	72	61
P-MORPH	9935	2760	674	292	165	106	76	57	45
P-PPI	2745	807	244	112	56	36	34	24	21
P-FINDR	695	302	73	25	18	12	10	8	7
P-MORPHEE	1067	333	76	33	19	12	9	6	5
P-MORPHSU	2308	615	152	66	37	24	17	12	10

endmembers provided by P-PPI, P-FINDR and P-MORPHEE using a straightforward parallel linear spectral unmixing (P-LSU) algorithm and the proposed P-MORPHSU algorithm for spatial-spectral abundance estimation. In both cases, we tested unconstrained and fully constrained versions (i.e., with sum-to-one and non-negativity restrictions) of the algorithms. Although ground-truth information on endmember fractional abundances at sub-pixel levels is not available for the Cuprite data set (this type of reference information is very difficult to be obtained in real-world scenarios), our quantitative experiments demonstrated that the use of the unconstrained P-MORPHSU generally resulted in very few negative abundance estimations, while the constrained P-MORPHSU provided very similar results to those reported by the unconstrained version of the same algorithm. In contrast, a more significant fraction of negative abundances was obtained by the unconstrained P-LSU with regards to the constrained P-LSU. It should be noted that a common indicator of poor model fitting and/or inappropriate selection of endmembers is estimation of negative abundance fractions by unconstrained linear models.

Fig. 7.9(a) shows the scalability of the considered parallel endmember extraction and spectral unmixing algorithms on Thunderhead. As shown by Fig. 7.9(a), the P-MORPHSU algorithm scaled nicely due to the lack of data dependencies

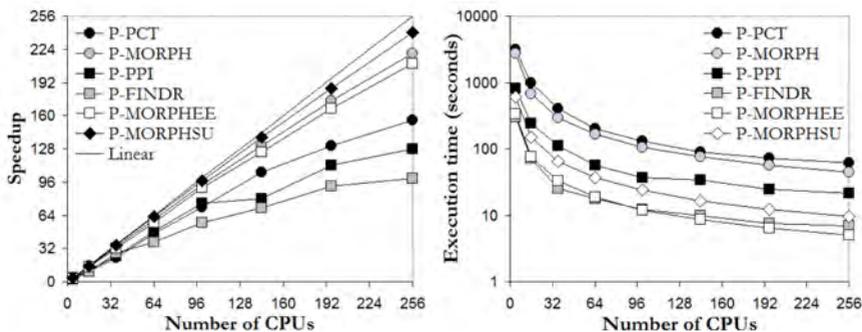


Fig. 7.9. Performance results after processing the AVRIS Cuprite scene with the proposed parallel algorithms on Thunderhead: (a) Parallel algorithm scalability; (b) Graphical representation of execution times as a function of the number of CPUs

Table 7.5. Spectral similarity scores among USGS spectra and the endmembers extracted from the original image, and from several reconstructed versions of the original image after applying the P-MORPHCOMP, JPEG2000 and 3D-SPIHT algorithms with different compression ratios

Mineral signature	Original image	P-MORPHCOMP			JPEG 2000			3D-SPIHT		
		20:1	40:1	80:1	20:1	40:1	80:1	20:1	40:1	80:1
Alunite	0.084	0.096	0.105	0.112	0.128	0.134	0.143	0.115	0.121	0.124
Buddingtonite	0.106	0.124	0.131	0.142	0.168	0.174	0.179	0.147	0.152	0.163
Calcite	0.105	0.118	0.129	0.138	0.166	0.175	0.184	0.131	0.148	0.165
Kaolinite	0.125	0.134	0.142	0.150	0.189	0.194	0.199	0.156	0.161	0.169
Muscovite	0.136	0.140	0.147	0.158	0.178	0.196	0.205	0.160	0.169	0.178
Chlorite	0.112	0.121	0.128	0.139	0.170	0.176	0.181	0.154	0.161	0.169
Jarosite	0.106	0.121	0.128	0.135	0.156	0.165	0.172	0.143	0.149	0.157
Montmorillonite	0.108	0.125	0.136	0.140	0.163	0.169	0.175	0.148	0.154	0.162
Nontronite	0.102	0.112	0.120	0.131	0.157	0.172	0.179	0.134	0.145	0.156
Pyrophyllite	0.094	0.103	0.110	0.116	0.130	0.138	0.146	0.119	0.123	0.133

and inter-processor communications in its parallel implementation. A similar comment applies to the P-MORPHEE algorithm, which scaled almost as effectively. Quite opposite, both the P-PPI and P-FINDR resulted in lower speedups due to the less *pleasingly parallel* nature of their implementation, as reported in previous work [25]. Although the sequential versions of these algorithms took less time than P-MORPHEE to produce their output in a single processor (see Table 7.3), the better scalability of P-MORPHEE as the number of processors was increased resulted in lower execution times in Fig. 7.9(b). Finally, both P-PCT and P-MORPH behaved in similar terms as in the previous experiment, with significantly reduced execution times (by a factor of approximately 4) resulting from the smaller size in MB of the scene. For illustrative purposes, Table 7.4 displays the execution times in seconds measured for different processors in this application case study.

To conclude this subsection, we evaluate the data compression capabilities of the proposed P-MORPHCOMP method, which comprises the combination of

P-MORPHEE for endmember extraction followed by P-MORPHSU for abundance estimation. In order to explore the quality of the compressed images produced by the proposed compression method, Table 7.5 reports the SAD scores among the USGS reference signatures in Fig. 7.7(b) and the endmembers extracted by the best endmember extraction combination reported in Table 7.3 (P-PCT+P-FINDR) from the resulting images after data compression (the lowest the scores, the highest the similarity), using compression ratios of 20:1, 40:1 and 80:1 (given by different tested values of input parameter q). As expected, the highest-quality endmembers were extracted from the original data set. As the compression ratio was increased, the quality of extracted endmembers was decreased. For illustrative purposes, we have also included the results provided by two standard methods in our comparison, i.e., the 3-D SPIHT method [17] and the wavelet-based JPEG2000 multi-component method [18]). The JPEG2000 implementation used for our experiments was the one available in kakadu software (<http://www.kakadusoftware.com>). Both techniques are 3-D compression algorithms that treat the hyperspectral data as a 3-D volume, where the spectral information is the third dimension. Results in Table 7.5 show that such 3-D techniques, which enjoy great success in classical image processing, may not necessarily find equal success in hyperspectral image compression. Specifically, for the same compression ratio, a 3-D lossy compression may result in significant loss of spectral information which can be preserved much better, in turn, by a spectral unmixing-oriented lossy compression algorithm such as the proposed P-MORPHCOMP.

7.5.4 Summary

In this section we have thoroughly analyzed the performance of a suite of parallel algorithms for spatial-spectral processing of hyperspectral images in the context of two different application areas. Here, we intend to provide a quick reference of the main observations and lessons learned after each experiment. For that purpose, Table 7.6 summarizes the outcome of processing experiments conducted using parallel algorithms, including the algorithm combinations that performed best in each area, the best sequential and parallel processing times measured, and the use of spatial-spectral or just spectral information in the design of parallel algorithms. As Table 7.6 shows, the incorporation of spatial information to the traditionally spectrally-guided approach used in hyperspectral imaging can be very beneficial from the viewpoint of both algorithm accuracy and parallel algorithm design. We believe that the compendium of parallel spatial-spectral processing techniques and their detailed cross-validation in the context of real application domains, summarized on Table 7.6, may help hyperspectral image analysts and practitioners in this field in the task of selecting advanced data processing techniques and strategies for specific applications. Our experimental assessment of parallel algorithms also revealed important considerations which have not been previously addressed in the hyperspectral imaging literature to the authors' best knowledge:

Table 7.6. Summary of experiments using parallel spatial-spectral techniques

	Dimensionality reduction	Unsupervised classification	Endmember extraction	Abundance estimation	Data compression
Algorithms compared	P-PCT P-MORPH	P-ISODATA P-WSHED	P-PPI,P-FINDR P-MORPH	P-LSU P-MORPHSU	SPIHT,JPEG 2000 P-MORPHCOMP
Best method:	P-MORPH	P-WSHED	P-FINDR	P-MORPHSU	P-MORPHCOMP
Spatial info?	Yes	Yes	No	Yes	Yes
Time (seq.)	9935	60356	695	2308	3375
Time (par.)	45	355	7	10	15
Speedup	220.7	170.1	99.2	230.8	225.0

- Contrary to the common perception that spatial-spectral algorithms involve more complex operations than traditional, spectral-based techniques, results in this chapter indicate that spatial-spectral techniques, when carefully designed and implemented, can indeed be more *pleasingly parallel* than spectral-based techniques, mainly because they can reduce sequential computations at the master and only involve minimal communication between the parallel tasks, namely, at the beginning and ending of such tasks.
- Another important issue confirmed by experimental results is that the performance of spatial-spectral techniques is directly linked to the application domain. These techniques generally performed accurately in applications involving data sets with high spatial auto-correlation. On the other hand, applications in which the relevant image features are not spatially correlated generally benefit from the use of more spectrally-guided approaches.
- A relevant observation is that abundance estimation can be greatly improved by the incorporation of spatial context into the estimation. Standard techniques for fractional abundance determination in the literature have only resorted to the spectral-based techniques, and therefore the use of spatial-spectral information in the unmixing process (after a set of endmembers has been extracted) is a novel contribution first presented in this chapter.
- It is also important to emphasize that when the application considered requires high-quality preservation of the rich spectral information present in the original hyperspectral image, compression techniques based on spectral unmixing concepts are generally more effective than traditional 3-D compression techniques which have enjoyed great success in the image processing literature but may not be as successful in exploitation-based hyperspectral data compression.
- As a final major remark, this chapter has shown that spatial-spectral techniques are very appealing for the design of efficient parallel implementations, thus allowing their rapid execution in commodity cluster-based parallel systems.

7.6 Conclusions and Future Research

In this chapter, we have discussed the role of joint spatial-spectral information (via specialized morphological processing) in the analysis of hyperspectral images.

Specifically, we have explored the performance of five types of spatial-spectral algorithms for dimensionality reduction, unsupervised classification, endmember extraction, abundance estimation and data compression in the context of two different application domains, i.e., land-cover classification in agricultural applications, and mapping of geological features. Our experimental assessment of parallel spatial-spectral algorithms revealed important considerations about the properties and nature of such algorithms. On the other hand, performance results measured on the Thunderhead system at NASA's Goddard Space Flight Center indicate that our parallel implementations were able to provide adequate results in both the quality of the solutions and the time to obtain them, in particular, when they are implemented on commodity Beowulf clusters. The compendium of parallel spatial-spectral techniques presented in this chapter reflects the increasing sophistication of a field that is rapidly maturing at the intersection of many different disciplines, including image and signal processing, sensor design and instrumentation, parallel and distributed computing, and environmental applications. As future work, we plan to implement the full suite of parallel spatial-spectral algorithms discussed in this chapter on alternative high performance computing architectures, such as Grid computing environments and specialized hardware platforms, including field programmable gate arrays (FPGAs) and general-purpose graphic processing units (GPUs). These platforms may allow us to fully accomplish the goal of real-time processing of hyperspectral image data, with potential applications in on-board hyperspectral image data compression and analysis.

Acknowledgement

The author would like to acknowledge J. Dorband, J.C. Tilton and J.A. Gualtieri for their support and collaboration in the experiments developed on the NASA Thunderhead Beowulf cluster. The author also gratefully thanks A. Paz for generating some of the illustrations displayed in this chapter.

References

1. Goetz, A.F.H., Vane, G., Solomon, J.E., Rock, B.N.: Imaging spectrometry for Earth remote sensing. *Science* 22, 1147–1153 (1985)
2. Green, R.O., et al.: Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sensing of Environment* 65, 227–248 (1998)
3. Richards, J.A., Xia, J.: *Remote sensing digital image analysis: an introduction*, 4th edn. Springer, Heidelberg (2006)
4. Chang, C.I.: *Hyperspectral imaging: techniques for spectral detection and classification*. Kluwer, New York (2003)
5. Richards, J.A.: Analysis of remotely sensed data: the formative decades and the future. *IEEE Trans. Geoscience and Remote Sensing* 43, 422–432 (2005)
6. Jimenez, L.O., Rivera-Medina, J.L., Rodriguez-Diaz, E., Arzuaga-Cruz, E., Ramirez-Velez, M.: Integration of spatial and spectral information by means of unsupervised extraction and classification for homogenous objects applied to multispectral and hyperspectral data. *IEEE Trans. Geoscience and Remote Sensing* 43, 844–851 (2005)

7. Gamba, P., Dell'Acqua, F., Ferrari, A., Palmason, J.A., Benediktsson, J.A., Arnason, J.: Exploiting spectral and spatial information in hyperspectral urban data with high resolution. *IEEE Geoscience and Remote Sensing Letters* 1, 322–326 (2004)
8. Chanussot, J., Benediktsson, J.A., Fauvel, M.: Classification of remote sensing images from urban areas using a fuzzy probabilistic model. *IEEE Geoscience and Remote Sensing Letters* 3, 40–44 (2006)
9. Landgrebe, D.A.: *Signal theory methods in multispectral remote sensing*. Wiley, Hoboken (2003)
10. Plaza, A., Martinez, P., Perez, R., Plaza, J.: Spatial/spectral endmember extraction by multi-dimensional morphological operations. *IEEE Trans. Geoscience and Remote Sensing* 40, 2025–2041 (2002)
11. Plaza, A., Martinez, P., Plaza, J., Perez, R.: Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations. *IEEE Trans. Geoscience and Remote Sensing* 43, 466–479 (2005)
12. Soille, P.: *Morphological image analysis: principles and applications*. Springer, Heidelberg (2003)
13. Serra, J.: *Image analysis and mathematical morphology*. Academic, New York (1982)
14. Boardman, J.: Automating spectral unmixing of aviris data using convex geometry concepts. In: Green, R.O. (ed.) *Summaries of Airborne Earth Science Workshop*, pp. 111–114. JPL Publication 93-26, Pasadena, CA (1993)
15. Winter, M.: Algorithm for fast autonomous spectral endmember determination in hyperspectral data. In: Descour, M.R., Shen, S.S. (eds.) *Imaging Spectrometry V*, Proceedings of SPIE, pp. 266–275 (1999)
16. Adams, J.B., Smith, M.O., Johnson, P.E.: Spectral mixture modeling: a new analysis of rock and soil types at the Viking Lander 1 site. *Journal of Geophysical Research* 91, 8098–8112 (1986)
17. Said, A., Pearlman, W.A.: A New, fast, and efficient image codec based on set partitioning in hierarchical trees. *IEEE Trans. Circuits and Systems* 6, 243–350 (1996)
18. Taubman, D.S., Marcellin, M.W.: *JPEG2000: Image compression fundamentals, standards and practice*. Kluwer, Boston (2002)
19. Plaza, A., Chang, C.I.: *High performance computing in remote sensing*. Chapman & Hall/CRC Press, Boca Raton (2007)
20. Comer, M., Delp, E.: Morphological operations for color image processing. *Journal of Electronic Imaging* 8, 279–289 (1999)
21. Plaza, A., Martinez, P., Perez, R., Plaza, J.: A new approach to mixed pixel classification of hyperspectral imagery based on extended morphological profiles. *Pattern Recognition* 37, 1097–1116 (2004)
22. Plaza, A., Valencia, D., Plaza, J., Martinez, P.: Commodity cluster-based parallel processing of hyperspectral imagery. *Journal of Parallel and Distributed Computing* 66, 345–358 (2006)
23. Seinstra, F.J., Koelma, D.: User transparency: a fully sequential programming model for efficient data parallel image processing. *Concurrency and Computation: Practice and Experience* 16, 611–644 (2004)
24. Plaza, A., Plaza, J., Valencia, D.: Impact of platform heterogeneity on the design of parallel algorithms for morphological processing of high-dimensional image data. *Journal of Supercomputing* 40, 81–107 (2007)

25. Plaza, A., Valencia, D., Plaza, J., Chang, C.I.: Parallel implementation of endmember extraction algorithms from hyperspectral data. *IEEE Geoscience and Remote Sensing Letters* 3, 334–338 (2006)
26. Benediktsson, J.A., Palmason, J.A., Sveinsson, J.R.: Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Trans. Geoscience and Remote Sensing* 42, 480–491 (2005)
27. Beucher, S.: Watershed, hierarchical segmentation and waterfall algorithm. In: Dougherty, E. (ed.) *Mathematical morphology and its applications to image processing*. Kluwer, Boston (1994)
28. Malpica, N., Ortuno, J.E., Santos, A.: A multichannel watershed-based algorithm for supervised texture segmentation. *Pattern Recognition Letters* 24, 1545–1554 (2003)
29. Moga, A.N., Gabbouj, M.: Parallel marker-based image segmentation with watershed transformation. *Journal of Parallel and Distributed Computing* 51, 27–45 (1998)
30. Plaza, A., Martinez, P., Perez, R., Plaza, J.: A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data. *IEEE Trans. Geoscience and Remote Sensing* 42, 650–663 (2004)
31. Plaza, A., Chang, C.I.: Impact of initialization on design of endmember extraction algorithms. *IEEE Trans. Geoscience and Remote Sensing* 44, 3397–3407 (2006)
32. Graña, M., Hernandez, C., Gallego, J.: A single individual evolutionary strategy for endmember search in hyperspectral images. *Information Sciences* 161, 181–197 (2004)
33. Ramakrishna, B., Plaza, A., Chang, C.I., Ren, H., Du, Q., Chang, C.C.: Spectral/spatial hyperspectral image compression. In: Storer, J.A., Motta, G. (eds.) *Hyperspectral data compression*. Springer, Heidelberg (2006)
34. Valencia, D., Plaza, A.: FPGA-based compression of hyperspectral imagery using spectral unmixing and the pixel purity index algorithm. *LNCS*, vol. 3993, pp. 24–31. Springer, Heidelberg (2006)
35. Achalakul, T., Taylor, S.: A distributed spectral-screening PCT algorithm. *Journal of Parallel and Distributed Computing* 63, 373–384 (2003)
36. Dhodhi, M.K., Saghri, J.A., Ahmad, I., Ul-Mustafa, R.: D-ISODATA: A distributed algorithm for unsupervised classification of remotely sensed data on networks of workstations. *Journal of Parallel and Distributed Computing* 59, 280–301 (1999)
37. Du, Q., Chang, C.I.: Estimation of number of spectrally distinct signal sources in hyperspectral imagery. *IEEE Trans. Geoscience and Remote Sensing* 42, 608–619 (2004)