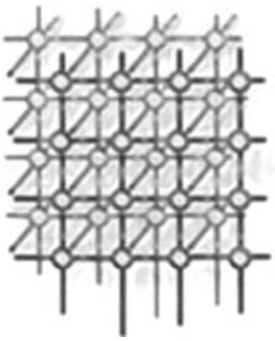
CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2008; **20**:1539–1572
Published online 28 April 2008 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe.1291



Parallel processing of remotely sensed hyperspectral imagery: full-pixel versus mixed-pixel classification

Antonio J. Plaza*, †

Department of Technology of Computers and Communications, University of Extremadura, Avda. de la Universidad s/n, E-10071 Càceres, Spain

SUMMARY

The rapid development of space and computer technologies allows for the possibility to store huge amounts of remotely sensed image data, collected using airborne and satellite instruments. In particular, NASA is continuously gathering high-dimensional image data with Earth observing hyperspectral sensors such as the Jet Propulsion Laboratory's airborne visible–infrared imaging spectrometer (AVIRIS), which measures reflected radiation in hundreds of narrow spectral bands at different wavelength channels for the same area on the surface of the Earth. The development of fast techniques for transforming massive amounts of hyperspectral data into scientific understanding is critical for space-based Earth science and planetary exploration. Despite the growing interest in hyperspectral imaging research, only a few efforts have been devoted to the design of parallel implementations in the literature, and detailed comparisons of standardized parallel hyperspectral algorithms are currently unavailable. This paper compares several existing and new parallel processing techniques for pure and mixed-pixel classification in hyperspectral imagery. The distinction of pure versus mixed-pixel analysis is linked to the considered application domain, and results from the very rich spectral information available from hyperspectral instruments. In some cases, such information allows image analysts to overcome the constraints imposed by limited spatial resolution. In most cases, however, the spectral bands collected by hyperspectral instruments have high statistical correlation, and efficient parallel techniques are required to reduce the dimensionality of the data while retaining the spectral information that allows for the separation of the classes. In order to address this issue, this paper also develops a new parallel feature extraction algorithm that integrates the spatial and spectral information. The proposed technique is evaluated (from the viewpoint of both classification accuracy and parallel performance) and compared with other parallel techniques for dimensionality reduction and classification in the context of three representative application case

*Correspondence to: Antonio J. Plaza, Department of Technology of Computers and Communications, University of Extremadura, Avda. de la Universidad s/n, E-10071 Càceres, Spain.
†E-mail: aplaza@unex.es

Contract/grant sponsor: European Commission; contract/grant number: MRTN-CT-2006-035927
Contract/grant sponsor: Spanish Ministry of Education and Science; contract/grant number: PR2003-0360

studies: urban characterization, land-cover classification in agriculture, and mapping of geological features, using AVIRIS data sets with detailed ground-truth. Parallel performance is assessed using Thunderhead, a massively parallel Beowulf cluster at NASA's Goddard Space Flight Center. The detailed cross-validation of parallel algorithms conducted in this work may specifically help image analysts in selection of parallel algorithms for specific applications. Copyright © 2008 John Wiley & Sons, Ltd.

Received 28 November 2006; Revised 8 November 2007; Accepted 21 November 2007

KEY WORDS: parallel image processing; hyperspectral imaging; cluster computing; parallel processing; load balance; morphological processing; spatial/spectral integration; hyperspectral applications

1. INTRODUCTION

Hyperspectral imaging (also known as imaging spectroscopy) is an emerging technique that has gained tremendous popularity in many research areas, most notably, in remotely sensed satellite imaging and aerial reconnaissance [1]. This technique 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. Recent advances in sensor technology have led to the development of advanced hyperspectral instruments capable of collecting image data, using hundreds of contiguous spectral channels, over the same area on the surface of the Earth. The concept of hyperspectral imaging is linked to one of NASA's premier instruments for Earth exploration, the Jet Propulsion Laboratory's Airborne Visible–Infrared Imaging Spectrometer (AVIRIS) system [2]. As shown in Figure 1, this imager measures reflected radiation in the wavelength region from 0.4 to 2.5 μm using 224 spectral channels, at a nominal spectral resolution of 10 nm. The result is an 'image cube' in which each pixel is given by a vector of values that can be interpreted as a representative *spectral signature* [3] for the observed material.

The incorporation of latest-generation sensors such as AVIRIS onto airborne and satellite platforms is now producing a nearly continual stream of high-dimensional data, and this explosion in the amount of collected information has rapidly introduced new processing challenges. For instance, it is estimated that NASA collects and sends to Earth more than 850 GB of hyperspectral data every day. In particular, the price paid for the wealth of spatial and spectral information available from hyperspectral sensors is the enormous amounts of data that they generate. As a result, the development of techniques able to transform a large amount of hyperspectral data into scientific understanding quickly enough for practical use is critical for Earth science and planetary exploration [4].

In recent years, several efforts have been directed towards the incorporation of massively parallel computing models into remote sensing applications [5,6], especially with the advent of relatively cheap Beowulf clusters. For instance, the concept of Beowulf cluster originated at the Center of Excellence in Space and Data Information Sciences (CESDIS), located at NASA's Goddard Space Flight Center in Maryland [7]. The goal of building Beowulf clusters in the context of remote sensing applications was mainly to create a set of cost-effective parallel computing systems from commodity components to satisfy specific computational requirements for the Earth and space sciences community. The purpose of commodity cluster computing is to maximize the performance-to-cost ratio of computing by using low-cost commodity components and free-source Linux and

DOI: 10.1002/cpe

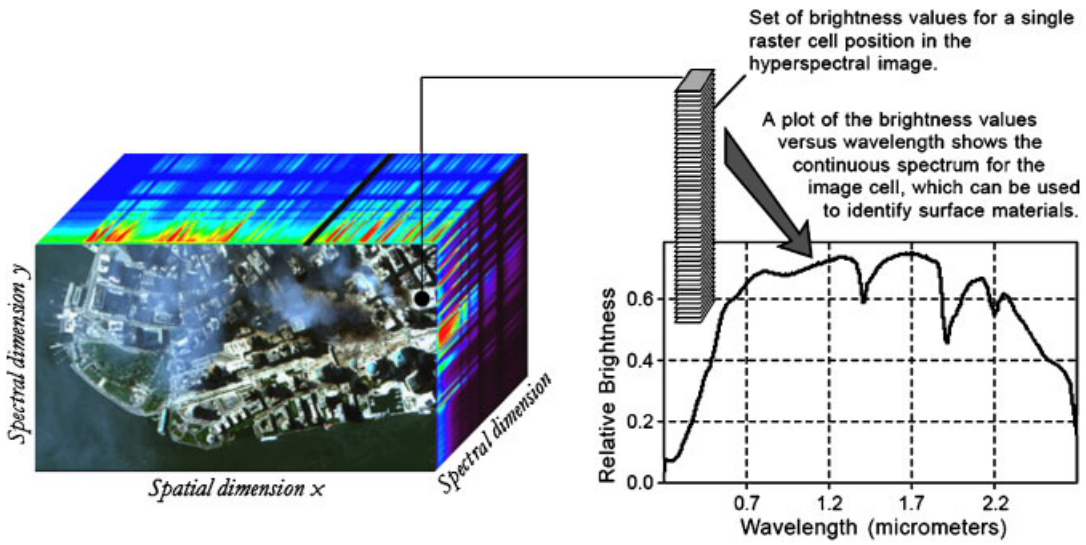


Figure 1. The concept of hyperspectral imaging illustrated using Jet Propulsion Laboratory's AVIRIS system.

GNU (Gnu's not Unix) software to assemble a system with computational speed in the order of hundreds of Gflops, and memory capacity sufficient for storing and processing very large data sets.

It is important to emphasize that, although the number of spectral bands recorded by hyperspectral imagers is already in the order of several hundreds (this figure will soon be increased to several thousands with the development of various *ultraspectral* imagers), the application of cluster computing techniques to hyperspectral image analysis is not simply intended to process all available spectral bands in the input data. For instance, many hyperspectral imaging applications only make use of a limited number of spectral bands, e.g. in mineral analyses, the spectral features in the wavelength region from 2 to 2.5 μm are more representative for mineral characterization than features in other regions. Most importantly, many of the spectral bands recorded by hyperspectral instruments have high statistical correlation. As a result, previous efforts have demonstrated that high-dimensional data spaces are mostly empty, indicating that the data structure involved exists primarily in a subspace [8]. Despite the ever-increasing availability of a low-cost parallel computing infrastructure, there is still a need for feature extraction techniques that can reduce the dimensionality of the data to the right subspace without losing the original information that allows for the separation of classes. For that purpose, several dimensionality reduction techniques have been widely used in order to bring the data from a high-order dimension to a low-order dimension, thus overcoming the 'curse' of dimensionality [9].

One of the most widely used dimension reduction techniques in remote sensing is the principal component transform (PCT) [8], which computes orthogonal projections that maximize the amount of data variance, and yields a data set in a new uncorrelated coordinate system. This transform is often used as a preprocessing step for classification. Unfortunately, information content in hyperspectral images does not always match such projections [10]. This rotational transform is characterized by its global nature, and as a result, it might not preserve all the information useful

DOI: 10.1002/cpe

to obtain a good classification. 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. As a result, feature extraction is carried out without incorporating information on the spatially adjacent data, i.e. the data are not managed as an image but as a disarranged listing of spectral measurements in which the spatial coordinates can be randomly permuted without affecting the final analysis. In turn, one of the distinguishing properties of hyperspectral data (as collected by available imaging spectrometers) is the rich spectral information coupled with a two-dimensional (spatial) pictorial representation amenable to image interpretation. Subsequently, there is a need for computationally efficient techniques able to incorporate both the spatial and spectral information when extracting relevant features for the understanding of hyperspectral image data.

In this regard, the wealth of spatial and spectral information provided by latest-generation hyperspectral instruments has opened ground-breaking perspectives in many different application domains. For instance, in the context of image segmentation/classification, there are several applications that demand accurate labeling of the individual pixels in a scene [8]. Examples range from urban area mapping, which generally requires sufficient spatial resolution to distinguish between small spectral classes (such as trees in a park or cars on a street), to other complex land-cover classification scenarios such as crop classification in agriculture. In some cases (e.g. when the available spatial resolution is not fine enough to separate agricultural crops that are very early in their growth cycle), sub-pixel classification techniques are required to accurately model pixels that are essentially mixed in nature. Fortunately, the extremely rich spectral information available from hyperspectral instruments allows processing techniques to overcome the limitations imposed by limited spatial resolution in certain application domains. As a result, classification labels in hyperspectral imaging can be *hard* (if we assume that the pixel is 'pure' and therefore can be associated to a single class) or *soft* (if we assume that the pixel is 'mixed' or composed of multiple underlying substances residing at a sub-pixel level). The former case is generally referred to as 'full-pixel classification' in the literature [8], whereas for the latter case the term 'mixed-pixel classification' is commonly employed [3]. It should be noted that a comparison of parallel techniques for hyperspectral image processing from the viewpoint of both full- and mixed-pixel classification has never been conducted in the past. In addition, very few research efforts have addressed the combined use of spatial and spectral information in the design of parallel techniques for feature extraction from hyperspectral images.

In this paper, we take a necessary first step towards the development and comparison of parallel techniques for feature extraction and full-/mixed-pixel classification of hyperspectral images. One of the main contributions of this paper is the design and implementation of a new parallel morphological feature extraction algorithm that integrates the spatial and spectral information in the hyperspectral data as opposed to currently available techniques such as PCT, which only make use of the spectral information. A detailed comparison of this spatial/spectral feature extraction technique to other parallel dimensionality reduction and classification algorithms, from the viewpoint of their capacity to model both pure and mixed pixels, is also conducted. The exhaustive cost–performance assessment of the proposed feature extraction method, in combination with other parallel techniques for dimensionality reduction and classification of hyperspectral images, is an entirely new contribution of this work. Such cross-validation of parallel algorithms has been conducted in the context of three highly representative applications (i.e. urban area characterization, land-cover classification, and mapping of geological features), thus

DOI: 10.1002/cpe

providing a detailed interpretative discussion along with application-oriented insights that have the potential to assist hyperspectral image analysts in the selection of parallel algorithms for specific applications.

The remainder of the paper is organized as follows. Section 2 describes the data partitioning strategies adopted for parallelization. Section 3 describes several parallel hyperspectral image processing algorithms, including techniques for dimensionality reduction/feature extraction, pure pixel selection, and mixed-pixel characterization. Section 4 includes a detailed survey on the analysis accuracy and parallel performance of the discussed algorithms in the context of three different application domains, and further analyzes the scalability of parallel algorithms on Thunderhead, a massively parallel Beowulf cluster at NASA's Goddard Space Flight Center in Maryland. Variables such as the impact of inter-processor communication and coordination, load balance, and speedup ratios are thoroughly investigated. This section also provides a detailed interpretative discussion of the results obtained from the viewpoint of classification accuracy, parallel performance, and suitability to the considered application domains. Finally, Section 5 concludes with some remarks and hints at plausible future research.

2. DATA PARTITIONING STRATEGIES

As mentioned in the previous section, most available techniques for classification of hyperspectral images focus on analyzing the data based on the properties of *spectral signatures* (see Figure 1), i.e. these techniques use the information provided by the pixel vector *as a whole*, potentially complementing this source of information (spectral) 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. In particular, it has been shown in the literature that domain decomposition techniques provide great flexibility and scalability for parallel image processing [11,12]. In hyperspectral imaging, two types of partitioning can be exploited: spectral-domain partitioning and spatial-domain partitioning [6]. Spectral-domain partitioning subdivides the volume into sub-volumes made up of contiguous spectral bands [see Figure 2(a)], and assigns

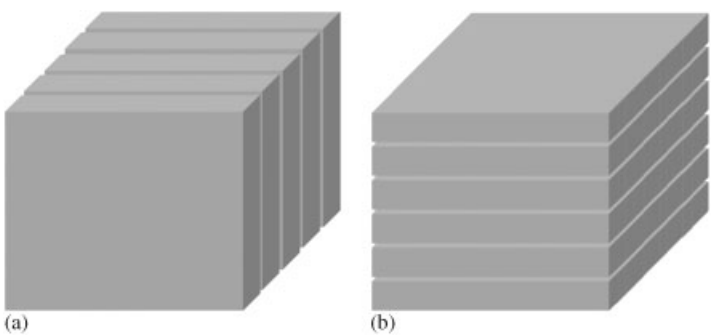


Figure 2. (a) Spectral-domain and (b) spatial-domain partitioning of a hyperspectral image cube.

DOI: 10.1002/cpe

one or more sub-volumes to each processor. With this model, each pixel vector may be split among several processors and the communication cost associated with the computations based on spectral signatures would be increased [13].

In order to exploit parallelism as much as possible, we have adopted a spatial-domain partitioning approach [see Figure 2(b)], in which the data are partitioned in slabs that retain the full spectral information associated with each pixel vector. There are several reasons that justify our decision to incorporate spatial-domain partitioning techniques in our application:

1. 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 [13].
2. A second major reason 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.
3. 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 [6].

As will be shown in the following section, all the parallel algorithms developed in this study have been designed from the assumption that each pixel vector is uniquely represented by its associated spectral signature. Therefore, the introduction of a spectral domain-based decomposition approach would require additional strategies to combine the partial results from several processing elements. A final consideration regarding data partitioning for kernel-based image processing techniques is noteworthy. In this type of image processing algorithms, a sliding kernel is moved throughout the spatial domain of the image. The goal is to define a local neighborhood around each image pixel, so that a local processing is performed with the set of pixels falling into the spatial context defined by the kernel. In this case, additional inter-processor communications are required when the window-based computation needs to be split among several different processing nodes due to boundary effects, as illustrated in Figure 3(a) using a spatial kernel with 3×3 pixels in size. In the example, the computations for a certain pixel, i.e. the pixel at spatial coordinates (5, 3) in the original image—denoted by $\mathbf{f}(5, 3)$—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 kernel-based computation for the pixel $\mathbf{f}(5, 3)$ in partition #1. However, if an overlap border is added to partition #1 (consisting of the entire first row of pixels allocated to partition #2), as illustrated in Figure 3(b), then there is no need to exchange boundary data between neighboring processors in order to complete all calculations at partition #1. 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 kernel 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

DOI: 10.1002/cpe

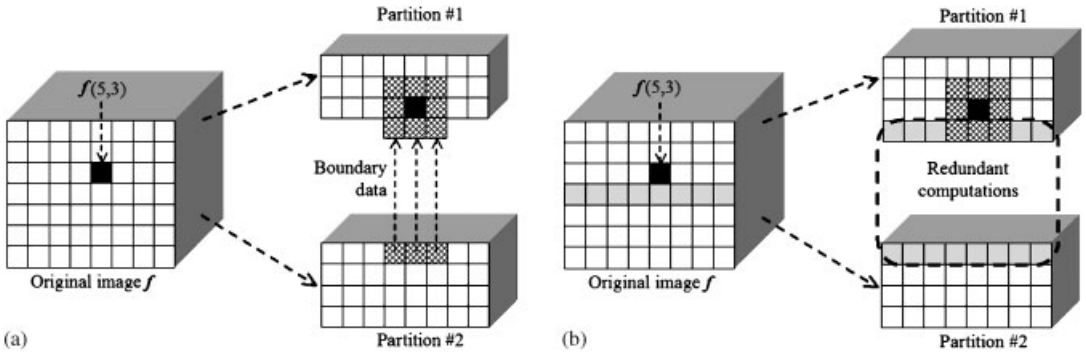


Figure 3. (a) 3×3 kernel computation split among two processing elements and (b) introduction of an overlap border to minimize inter-processor communication in a 3×3 kernel computation.

been recently explored in [14] concluding that, for small kernels, a redundant-computation strategy is often preferred.

3. PARALLEL HYPERSPECTRAL IMAGING ALGORITHMS

This section describes several parallel algorithms for full- and mixed-pixel classification of hyperspectral images. At this point, we reiterate the distinction between full-pixel techniques and mixed-pixel techniques. The underlying assumption governing full-pixel techniques is that each pixel vector measures the response of one predominantly underlying material at each site in a scene. In contrast, the underlying assumption governing mixed-pixel techniques is that each pixel vector measures the response of multiple underlying materials at each site. As a result, a hyperspectral image is often a combination of the two situations, i.e. a few pixels in a scene may be pure materials but many others are actually mixtures of materials. Three types of parallel algorithms are considered in this section: (1) dimensionality reduction and feature extraction; (2) pure pixel classification; and (3) spectral unmixing.

3.1. Parallel algorithms for dimensionality reduction and feature extraction

In this category, we first outline a parallel version of the PCT transform for spectral-based feature extraction in hyperspectral data. Then, we introduce a new parallel algorithm for morphological feature extraction that integrates the spatial and spectral information in a simultaneous fashion.

3.1.1. Parallel implementation of PCT (P-PCT)

The PCT transform has been often used to summarize and decorrelate the information in hyperspectral images by reducing redundancy and packing the residual information into a small set of images, termed principal components [8]. PCT is a highly compute-intensive algorithm amenable to

DOI: 10.1002/cpe

parallel implementation. On the basis of previous parallelization efforts adopted for this transform [15,16], we adopt a standard master–slave decomposition technique, where the master coordinates the actions of the workers, gathers the partial results from them, and provides the final result. The input to the parallel algorithm is an N-dimensional hyperspectral image cube $\mathbf{f}$ and the output is a transformed image cube $\mathbf{g}$. The steps implemented by the master–slave parallel algorithm can be summarized as follows:

1. The master divides the original image cube $\mathbf{f}$ into K spatial-domain partitions, where K is the number of workers in the system. Then, the master sends a partition, consisting of a slab of pixel vectors [see Figure 2(b)], to each of the workers. Since our target platform in this work is assumed to be a massively parallel homogeneous cluster, the input image is partitioned into slabs with approximately the same number of (spatially contiguous) pixel vector rows. In case an exact division of the input data volume into equally sized slabs is not possible, then one of the processors (other than the master) will receive the remaining pixel vector rows until the entire original data cube is distributed among the workers.
2. The mean vector $\mathbf{m}$ is computed concurrently, and each worker computes a local mean component using the pixel vectors in its local partition as follows:

$$\mathbf{m}_k = \frac{1}{P-1} \sum_{x=1}^{k_x} \sum_{y=1}^{k_y} \mathbf{f}(x, y)$$

   where P is the total number of pixel vectors in the scene, and k_x and k_y, respectively, denote the total number of pixel vector rows and columns in the local, spatial-domain data partition allocated to processor k, with $k = \{1, \ldots, K\}$. Each worker then returns its local component to the master, which forms the mean vector $\mathbf{m}$ by adding up the components provided by the workers and broadcasts $\mathbf{m}$ to the workers.
3. The covariance matrix is computed concurrently, and each worker first subtracts the mean vector $\mathbf{m}$ from each pixel vector $\mathbf{f}(x, y)$ in its local partition and then obtains a local covariance component by computing the cross-product of each unbiased pixel vector with its own transpose as follows:

$$\frac{1}{P-1} \sum_{x=1}^{k_x} \sum_{y=1}^{k_y} \mathbf{f}(x, y) \cdot \mathbf{f}(x, y)^{\mathrm{T}}$$

   thus producing a matrix for each local pixel vector. Each worker then sums up all matrices and returns the result to the master, which forms the covariance matrix by adding up the local covariance components provided by all the workers.
4. The master computes the eigenvalues and the corresponding eigenvectors using the covariance matrix. In this work we use the Jacobi method [17], which computes all eigenvalues at once. Then, a transformation matrix $\mathbf{T}$ is formed by sorting the eigenvectors according to their corresponding eigenvalues, which provide a measure of their variances. As a result, the spectral content is forced into the front components. Since the degree of data dependency of the calculation is high and its complexity is related to the number of spectral bands rather than the image size, this step is done sequentially at the master. Once this step is finalized, the matrix $\mathbf{T}$ is broadcast to all the workers.

DOI: 10.1002/cpe

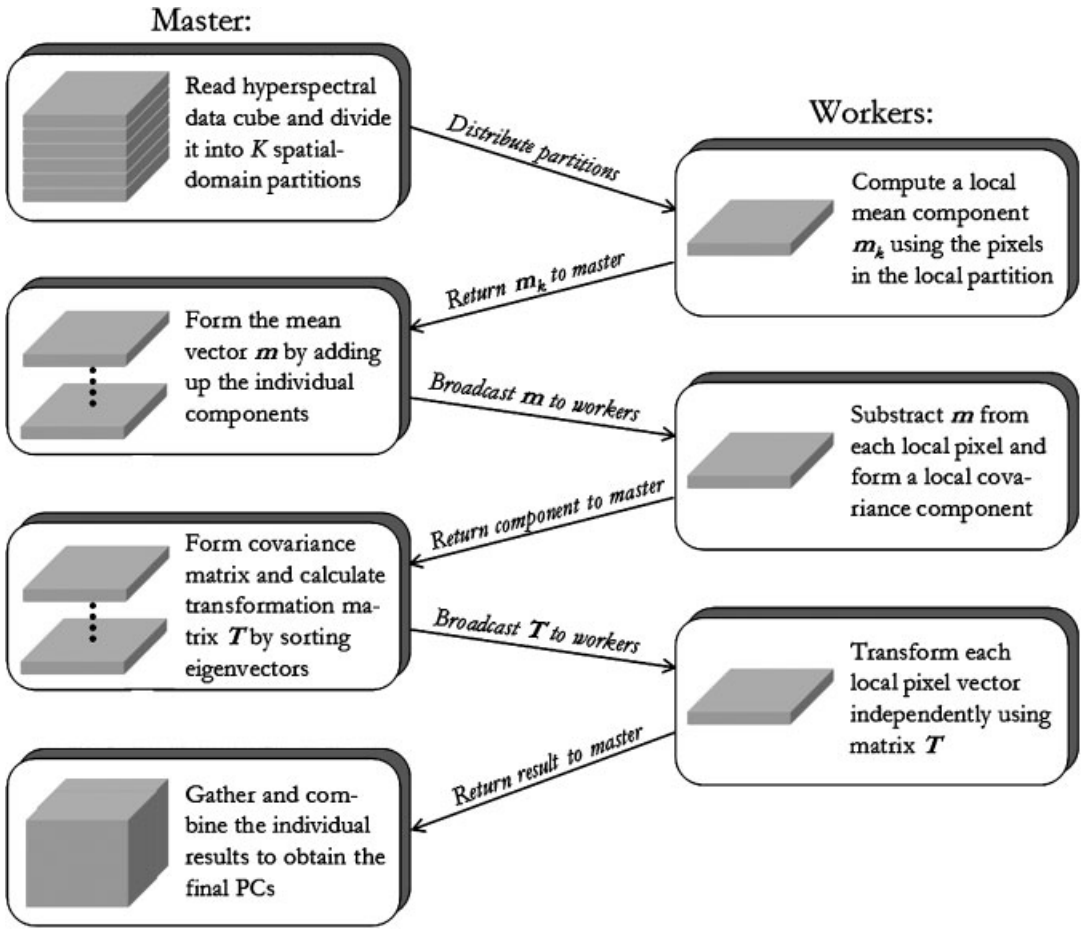


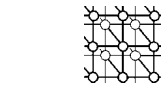
Figure 4. Flowchart illustrating the different steps involved in the P-PCT algorithm.

5. Transform each pixel vector in the original hyperspectral image independently using $\mathbf{g}(x, y) = \mathbf{T} \cdot \mathbf{f}(x, y)$. This step is done in parallel, where all workers transform their respective portions of data concurrently and the master gathers the individual results. For illustrative purposes, Figure 4 shows a flowchart summarizing the different steps involved in the P-PCT algorithm.

3.1.2. Parallel morphological feature extraction (P-MORPH)

A second parallel algorithm is introduced in this subsection to naturally integrate the spatial and spectral information present in the input hyperspectral data, as an alternative to P-PCT, which relies on the spectral information alone. Mathematical morphology [18] is a non-linear image

DOI: 10.1002/cpe



processing technique that provides a remarkable framework to achieve the desired integration. This approach relies on the definition of a so-called structuring element (SE), which acts as a probe for extracting or suppressing specific image structures. The SE can be ultimately seen as a kernel that is moved through the spatial domain of the image defining a spatial context around each pixel. Although mathematical morphology was originally defined for binary images, and then extended to grayscale images [19], we have extended the concept of morphological filtering to hyperspectral images by defining a cumulative distance between each pixel vector, $\mathbf{f}(x, y)$, and all pixel vectors in the spatial neighborhood of $\mathbf{f}(x, y)$ defined by a certain SE (denoted by B) as follows [10]: $D_B[\mathbf{f}(x, y)] = \sum_s \sum_t \text{SAD}[\mathbf{f}(x, y), \mathbf{f}(s, t)]$, where $(s, t) \in Z^2(B)$ and SAD is the spectral angle distance [3,8], defined by $\text{SAD}[\mathbf{f}(x, y), \mathbf{f}(s, t)] = \cos^{-1}[\mathbf{f}(x, y) \cdot \mathbf{f}(s, t)/\|\mathbf{f}(x, y)\| \cdot \|\mathbf{f}(s, t)\|]$. This metric offers important advantages for hyperspectral image processing, such as robust performance in the presence of atmospheric and illumination interferers.

Based on the distance metric above, two extended morphological operations (erosion and dilation) are defined and used, respectively, to extract the most highly pure and the most highly mixed pixel in the B-given spatial neighborhood as follows [15]: $(\mathbf{f} \otimes B)(x, y) = \arg\min_{(s,t)\in Z^2(B)}\{D_B[\mathbf{f}(x + s, y + t)]\}$ and $(\mathbf{f} \oplus B)(x, y) = \arg\max_{(s,t)\in Z^2(B)}\{D_B[\mathbf{f}(x + s, y + t)]\}$. For illustrative purposes, Figure 5 shows a graphical illustration of these two basic operators using a toy example in which a synthetic hyperspectral image is used for demonstration. As can be seen from Figure 5, morphological dilation expands the spatial regions made up of pure pixel vector in accordance with the spatial neighborhood defined by a 3×3 SE, whereas morphological erosion expands the regions made up of highly mixed-pixel vectors in accordance with the same spatial neighborhood.

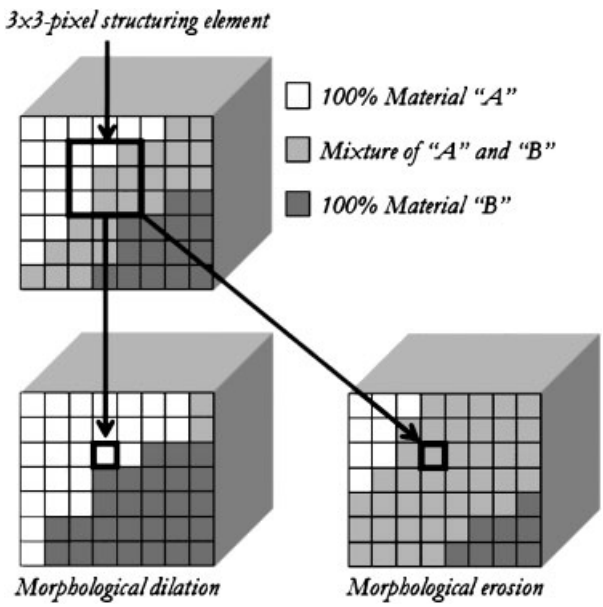


Figure 5. Graphical interpretation of extended morphological dilation/erosion operations.

DOI: 10.1002/cpe

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: $(\mathbf{f} \circ B)(x, y) = [(\mathbf{f} \otimes B) \oplus B](x, y)$, i.e. erosion followed by dilation, and $(\mathbf{f} \bullet B)(x, y) = [(\mathbf{f} \oplus B) \otimes B](x, y)$, i.e. dilation followed by erosion.

In order to perform spatial/spectral feature extraction, sequences of extended opening by reconstruction operations [20] with SE's of varying width (called morphological profiles) have been applied in the past to characterize image structures in grayscale remotely sensed image data. In this work, we extend this concept to spatial/spectral feature extraction by developing a new parallel method for morphological feature extraction in hyperspectral images. The inputs to the parallel algorithm are an N-dimensional hyperspectral image cube, $\mathbf{f}$, a maximum number of filtering iterations, t, and an SE B with a constant size of 3×3 pixels. The output is a transformed image cube, $\mathbf{g}$:

1. The master divides the original image cube $\mathbf{f}$ into K spatial-domain partitions, where K is the number of workers in the system. In order to avoid excessive inter-processor communication during morphological processing, the partitions are formed using overlap borders, as described in Figure 3(b).
2. Each worker performs spatial/spectral filtering on its local partition using the following steps [10]:
   (a) Compute an extended opening by reconstruction for each local pixel $\mathbf{f}(x, y)$ as follows: $(\mathbf{f} \circ B)^t(x, y) = \min_{t \geq 1}\{\delta_B^t(\mathbf{f} \circ B|\mathbf{f})(x, y)\}$, where $\delta_B^t(\mathbf{f} \circ B|\mathbf{f})(x, y) = \overbrace{\delta_B \delta_B \cdots \delta_B}^{t \text{ times}}(\mathbf{f} \circ B|\mathbf{f})(x, y)$ and $\delta_B(\mathbf{f} \circ B|\mathbf{f})(x, y) = \min\{[(\mathbf{f} \circ B) \oplus B](x, y), \mathbf{f}(x, y)\}$.
   (b) Compute an extended closing by reconstruction for each local pixel $\mathbf{f}(x, y)$ as follows: $(\mathbf{f} \bullet B)^t(x, y) = \max_{t \geq 1}\{\varepsilon_B^t(\mathbf{f} \bullet B|\mathbf{f})(x, y)\}$, where $\varepsilon_B^t(\mathbf{f} \bullet B|\mathbf{f})(x, y) = \overbrace{\delta_B \delta_B \cdots \delta_B}^{t \text{ times}}(\mathbf{f} \bullet B|\mathbf{f})(x, y)$ and $\varepsilon_B(\mathbf{f} \bullet B|\mathbf{f})(x, y) = \min\{[(\mathbf{f} \bullet B) \otimes B](x, y), \mathbf{f}(x, y)\}$.
   (c) Compute the derivative of the extended opening profile as follows: $\mathbf{p}_k^{\circ}(x, y) = \{\text{SAD}[(\mathbf{f} \circ B)^{\lambda}(x, y), (\mathbf{f} \circ B)^{\lambda-1}(x, y)]\}$, with $\lambda = \{1, 2, \ldots, t\}$. Here, $\mathbf{f}(x, y) = (\mathbf{f} \circ B)^0(x, y)$ for $\lambda = 0$ by the definition of opening by reconstruction [18].
   (d) Compute the derivative of the extended closing profile as follows: $\mathbf{p}_k^{\bullet}(x, y) = \{\text{SAD}[(\mathbf{f} \bullet B)^{\lambda}(x, y), (\mathbf{f} \bullet B)^{\lambda-1}(x, y)]\}$, with $\lambda = \{1, 2, \ldots, t\}$. Here, $\mathbf{f}(x, y) = (\mathbf{f} \bullet B)^0(x, y)$ for $\lambda = 0$ by the definition of closing by reconstruction [18].
   (e) Form a $(2t - 1)$-dimensional morphological profile for each local pixel $\mathbf{f}(x, y)$ as follows: $\text{MP} = \{p_k^{\circ}(x, y), \mathbf{p}_k^{\bullet}(x, y)\}$. 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 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 end of the parallel process. However, this approach is subject to a redundant-computation overhead introduced by the overlap borders used by the proposed data partitioning strategy.

DOI: 10.1002/cpe

3.2. Parallel algorithm for full-pixel classification

The second category of parallel algorithms considered in this work (specifically addressing the problem of full-pixel classification) is represented by a parallel version of ISODATA [8], one of the most widely used unsupervised clustering/classification algorithms in the literature. A standard parallel implementation of this algorithm is D-ISODATA [21], that has been used as a baseline for the development of our parallel algorithm. Here, we provide a modification (called P-ISODATA) that has been specifically adapted to a hyperspectral image classification scenario by making use of SAD (instead of the Euclidean distance) as a similarity measure to cluster data elements into different classes. The inputs to our parallel algorithm are an N-dimensional data cube, $\mathbf{f}$, a maximum number of clusters to be found, p, and a convergence threshold, t_c. The output is a two-dimensional image containing a classification label for each pixel $\mathbf{f}(x, y)$ in the original image. A master–slave parallel implementation of the algorithm can be summarized as follows:

1. Optionally, apply a dimension reduction technique to reduce the dimensionality of the input data.
2. Divide the original image cube $\mathbf{f}$ into K equally sized slabs so that there is no overlapping among the different spatial-domain partitions. Send each partition, consisting of a set of pixel vectors, to a worker, along with a set of p randomly selected pixel vectors assumed to be initial class centroids.
3. Each worker labels each pixel $\mathbf{f}(x, y)$ in the corresponding partition. Let us denote by n_{ij} the number of pixels belonging to the jth cluster of the ith worker, and by $\mathbf{f}_k^j(x, y)$ the kth pixel of the jth cluster. Then, to minimize inter-processor communication, each worker sends the number of pixels belonging to each cluster and the summation of feature vectors of all pixels belonging to each cluster, that is, $\mathbf{Y}_{ij} = [\sum_{k=1}^{n_{ij}} \mathbf{f}_k^1(x, y), \ldots, \sum_{k=1}^{n_{ij}} \mathbf{f}_k^p(x, y)]$, to the master processor.
4. The master collects all the information provided by the workers and combines it to obtain a new centroid for each cluster j using $c_j = (\sum_{i=1}^{p_j} \mathbf{Y}_{ij} / \sum_{i=1}^{p_j} \mathbf{n}_{ij})$, where p_j is the number of pixels in the jth cluster.
5. The master then compares the current centroids and the new centroids by calculating the SAD distance between them. If the angle between them is less than the convergence threshold t_c, then convergence occurs and the master informs all workers about the current status of convergence. Otherwise, steps 2–4 are repeated until the convergence status is true.
6. Following convergence, each worker i computes the summation of the SAD distance of all pixels within a cluster j from its centroid $\mathbf{c}_j$. Each worker then sends this information to the master, which combines it to obtain the deviation of pixels within each cluster j. It should be noted that this information is only exchanged when convergence has occurred (instead of doing so every time new centroids are calculated) in order to minimize inter-processor communication.
7. The master now decides between splitting or merging the resulting clusters, based on parameters such as the inter-cluster distances (separation), the intra-cluster distances (compactness), the ratio of standard deviations along different axes for each cluster, and the number of pixels per cluster. The above procedure is repeated until no further cluster is eligible for the split and merges operation. When the stopping criterion is satisfied, each worker sends the label

DOI: 10.1002/cpe

(cluster number) associated with each local pixel to the master, which combines all the individual results and forms a final, two-dimensional classification image in which each pixel vector has an associated (hard) class label.

3.3. Parallel algorithms for mixed-pixel classification

The third category of parallel algorithms considered in this work (specifically addressing the problem of mixed-pixel classification) comprises two types of algorithms. The first type is concerned with the automatic identification of pure spectral signatures (called spectral *endmembers* in hyperspectral analysis terminology [22]). Here, we develop parallel versions of two consolidated algorithms for this purpose: the pixel purity index (PPI) algorithm [23], available in Kodak's Research Systems ENVI (one of the most widely used commercial remote sensing software packages in the hyperspectral imaging community), and the N-FINDR algorithm [24], distributed by Technical Research Associates, Inc. (see http://www.tracam.com for details). Once a suite of spectral endmembers has been extracted from the data, a second type of inversion algorithms can be used to estimate the fractional *abundances* of each of the endmembers at mixed pixels, i.e. to determine the sub-pixel contribution of each endmember participating in a mixture. Here, we develop a parallel version of a commonly used technique for abundance estimation in the hyperspectral analysis literature, i.e. the fully constrained linear spectral unmixing (LSU) model [3,25].

3.3.1. Parallel pixel purity index (P-PPI)

The PPI is one of the most widely used endmember extraction algorithms in the remote sensing community due to its availability in well-established commercial software. The algorithm was designed to search for a set of vertices of a convex hull in a given data set that are assumed to represent the purest signatures available in the data. It first reduces the dimensionality of the input data and then proceeds by generating a large number of random, N-dimensional unit vectors called 'skewers' through the data set. Every data point is projected onto each skewer, and the data points that correspond to extrema in the direction of a skewer are identified and placed on a list (see Figure 6). As more skewers are generated the list grows, and the number of times a given pixel is placed on this list is also tallied. The pixels with the highest tallies are considered the purest ones.

A parallel version of the PPI algorithm has been recently proposed [26] and is used here for illustrative purposes. The inputs to the parallel algorithm are an N-dimensional image cube, $\mathbf{f}$, a number of skewers to be generated, s, a number of endmembers to be extracted, p, and a threshold value, t_{PPI}. The output is a set of p endmembers, denoted by $\{\mathbf{e}_i\}_{i=1}^{p}$. A step-by-step description of the parallel algorithm follows:

1. Apply a dimension reduction technique to reduce the dimensionality of the input data from N to p.
2. The master generates a set of s randomly generated unit N-dimensional vectors called 'skewers,' denoted by $\{\mathbf{skewer}_j\}_{j=1}^{s}$, and broadcasts the entire set of generated skewers to all the workers.

DOI: 10.1002/cpe

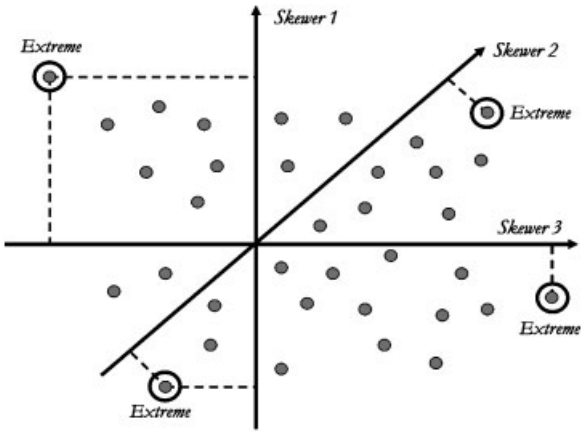


Figure 6. Toy example illustrating the performance of the PPI algorithm in two dimensions.

3. For each $\mathbf{skewer}_j$, each worker projects all the pixel vectors at each local partition k, with $k = \{1, \ldots, K\}$, onto $\mathbf{skewer}_j$ to form an extrema set for that skewer denoted by $S^{(k)}(\mathbf{skewer}_j)$.
4. Define an indicator function of a set S by

$$I_S(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in S \\ 0 & \text{if } \mathbf{x} \notin S \end{cases}$$

and use it to calculate $N_{PPI}^{(k)}[\mathbf{f}(x, y)] = \sum_j I_{S^{(k)}(\mathbf{skewer}_j)}[\mathbf{f}(x, y)]$ for each pixel $\mathbf{f}(x, y)$ at the local partition. Select those pixels with $N_{PPI}^{(K)}[\mathbf{f}(x, y)] > t_{PPI}$, and send them to the master.
5. The master collects all the partial endmember sets provided by the workers and merges them together to form a final set of spectrally distinct endmembers, $\{\mathbf{e}_i\}_{i=1}^{p}$.

3.3.2. *Parallel N-FINDR (P-FINDR)*

The N-FINDR algorithm aims at identifying the set of pixels that define the simplex with the maximum volume, potentially inscribed within the data set. After a dimensionality reduction step (mandatory for this algorithm), a set of pixel vectors is first randomly selected, and their corresponding volume is calculated. For illustrative purposes, Figure 7(a) shows an example of the above situation, in which three randomly selected endmembers (represented as black circles) define a volume which can be used to expressed mixed pixels included in the volume (represented as dark gray circles) in terms of linear combinations of selected endmembers. As shown in Figure 7(a), many hyperspectral pixel vectors (represented as light gray circles) may remain unexplained after the initial random selection, including the most extreme pixels (represented as white circles). In order to refine the initial estimate, a trial volume is then calculated for every pixel in each endmember position by replacing that endmember and recalculating the volume. If the replacement results in a volume increase, the pixel replaces the endmember. This procedure is repeated until there are no

DOI: 10.1002/cpe

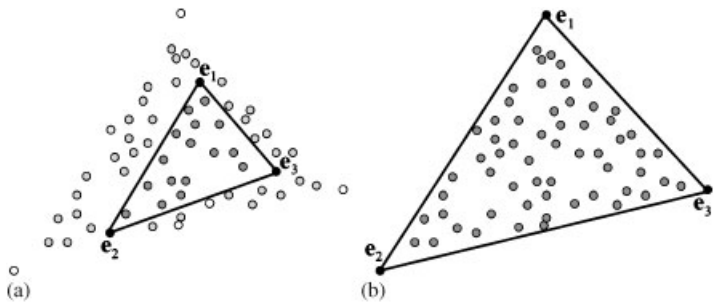


Figure 7. Toy example illustrating the concept of the N-FINDR endmember extraction algorithm in two dimensions: (a) initial volume estimation based on random endmembers and (b) final volume estimation and resulting endmembers.

replacements of endmembers left. As shown in Figure 7(b), the endmembers obtained at the end of this process will likely define a simplex that encloses most of the pixels in the input hyperspectral data set.

A master–slave parallel version of this endmember search process has been recently developed [26]. The inputs to the parallel algorithm (called P-FINDR) are N-dimensional image cube, $\mathbf{f}$, and the number of endmembers to be extracted, p. The output of the algorithm is a set of final endmembers, $\{\mathbf{e}_j\}_{j=1}^{p}$. A step-by-step description of the parallel algorithm follows:

1. Use a dimension reduction technique to reduce the dimensionality of the input data from N to p.
2. The master selects a random set of p initial pixels $\{\mathbf{e}_j^{(0)}\}_{j=1}^{p}$ and find $V(\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)})$, the volume of the simplex defined by $\{\mathbf{e}_j^{(0)}\}_{j=1}^{p}$, denoted by $S(\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)})$.
3. Calculate the volume of p simplexes, $V(\mathbf{f}(x, y), \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)}), \ldots, V(\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{f}(x, y))$ in parallel, each of which is formed by replacing one endmember $\mathbf{e}_j^{(0)}$ with the sample vector $\mathbf{f}(x, y)$. Each worker performs replacements locally, using all the pixels in its local partition.
4. If none of these p recalculated volumes is greater than $V(\mathbf{e}_1^{(0)}, \mathbf{e}_2^{(0)}, \ldots, \mathbf{e}_p^{(0)})$, then no endmember in $\{\mathbf{e}_j^{(0)}\}_{j=1}^{p}$ is replaced. Otherwise, the master replaces the endmember, which is absent in the largest volume among the p simplexes with the vector $\mathbf{g}(x, y)$. Let such an endmember be denoted by $\mathbf{e}_j^{(0)}$. A new set of endmembers is then produced by letting $\mathbf{e}_j^{(1)} = \mathbf{f}(x, y)$ and $\mathbf{e}_i^{(1)} = \mathbf{e}_i^{(0)}$ for $i \neq j$.
5. Repeat from step 2 until no replacements occur. The final combination gives the final set of endmembers, $\{\mathbf{e}_j\}_{j=1}^{p}$.

3.3.3. Parallel linear spectral unmixing (P-LSU)

A commonly adopted technique in the hyperspectral analysis literature for expressing mixed pixels as (linear) combinations of endmembers is the linear spectral unmixing (LSU) technique [3]. It can

DOI: 10.1002/cpe

be briefly described as follows. Suppose that p endmembers, $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p$, have been extracted from a hyperspectral image scene $\mathbf{f}$, and let $\mathbf{f}(x, y)$ be a mixed-pixel vector. LSU assumes that the spectral signature of $\mathbf{f}(x, y)$ can be represented by a linear mixture of $\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_p$ with abundance fractions specified by $a_1, a_2, \ldots, a_p$. As a result, we can model the spectral signature of an image pixel $\mathbf{f}(x, y)$ via a linear regression form: $\mathbf{f}(x, y) = \mathbf{e}_1 \cdot a_1 + \mathbf{e}_2 \cdot a_2 + \cdots + \mathbf{e}_p \cdot a_p$. Two constraints are often imposed on this model to produce physically meaningful solutions [25]. These are the abundance sum-to-one constraint, that is, $\sum_{i=1}^{p} a_i = 1$, and the abundance non-negativity constraint, that is, $a_i \geq 0$ for $1 \leq i \leq p$.

It should be noted that the process of LSU works on a pixel-by-pixel basis with no data dependencies involved; hence, a scalable parallel implementation can be simply developed. The inputs to our parallel version of LSU (called P-LSU) are an N-dimensional data cube, $\mathbf{f}$, and a set of spectral endmembers, $\{\mathbf{e}_j\}_{j=1}^{p}$. The output is a set of p endmember fractional abundances, $\{a_j(x, y)\}_{j=1}^{p}$, for each pixel vector $\mathbf{f}(x, y)$ in the original image. The parallel algorithm can be summarized by the following steps:

1. The master divides the original data cube $\mathbf{f}$ into K spatial-domain-based partitions, where K is the number of workers, and broadcasts the input endmember set $\{\mathbf{e}_j\}_{j=1}^{p}$ to all the workers.
2. For each pixel $\mathbf{f}(x, y)$ in the local partition, obtain a set of abundance fractions specified by $a_1(x, y), a_2(x, y), \ldots, a_p(x, y)$ using $\{\mathbf{e}_j\}_{j=1}^{p}$, so that $\mathbf{f}(x, y) = \mathbf{e}_1 \cdot a_1(x, y) + \mathbf{e}_2 \cdot a_2(x, y) + \cdots + \mathbf{e}_p \cdot a_p(x, y)$, taking into account the abundance sum-to-one and abundance non-negativity constraints [25].
3. The master collects all the individual sets of fractional abundances $\{a_j^{(k)}(x, y)\}_{j=1}^{p}$ calculated for the pixels at every individual partition k, with $k = \{1, \ldots, K\}$, and forms a final set of abundances designated by $\{a_j(x, y)\}_{j=1}^{p} = \bigcup_{k=1}^{K} \{a_j^{(k)}(x, y)\}_{j=1}^{p}$. Optionally, the master can also obtain a hard classification label to each pixel $\mathbf{f}(x, y)$ by associating it with a class given by the endmember with the highest fractional abundance score in that pixel. This is done by adopting a winner-take-all (WTA) criterion often used in neural networks [27], i.e. by comparing all estimated abundances, $a_1(x, y), a_2(x, y), \ldots, a_p(x, y)$, and finding the one with the maximum value, say $a_{j^*}(x, y)$, with $j^* = \arg\{\max_{1 \leq j \leq p}\{a_j(x, y)\}\}$.

4. EXPERIMENTAL RESULTS

This section provides an assessment of the effectiveness of the proposed parallel algorithms in the task of providing significant performance gains and without loss of accuracy in the analysis of real hyperspectral data sets. Three different application domains, namely, urban characterization, land-cover classification in agriculture, and 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 hyperspectral imaging techniques and algorithms. 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

DOI: 10.1002/cpe

context of three highly representative application domains follows. The section concludes with a summary and detailed discussion on the results obtained.

4.1. 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 the 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.

4.2. Hyperspectral data sets

Three 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.

4.2.1. AVIRIS hyperspectral data over the World Trade Center in New York City

The first scene used in experiments was obtained after an AVIRIS flight over the World Trade Center (WTC) area in New York City on September 16, 2001, just five days after the terrorist attacks that collapsed the two main towers and other buildings in the WTC complex. The selected data set comprises 614×512 pixels, 224 narrow spectral bands in the wavelength range 0.4–2.5 μm, and a total size of 140 MB. The spatial resolution is very fine as a consequence of the low altitude of the AVIRIS flight; with 1.7 m per pixel (most other AVIRIS data sets exhibit spatial resolutions of 20 m per pixel). The fine spatial resolution available in this application case study allowed us to use this scene as a benchmark for urban characterization studies, which often require very fine spatial and spectral detail to model complex image features. Figure 8(a) shows a false color composite of the data set selected for experiments using the 1682, 1107, and 655 nm channels, displayed as red, green, and blue, respectively (a picture of the full AVIRIS flight line collected over Manhattan is available online from the AVIRIS Web site at http://aviris.jpl.nasa.gov/ql/01qlook/f010916t01p01_r04.gql.jpg). A detail of the WTC area is shown in a red rectangle. As shown in Figure 8(a), vegetated areas

DOI: 10.1002/cpe

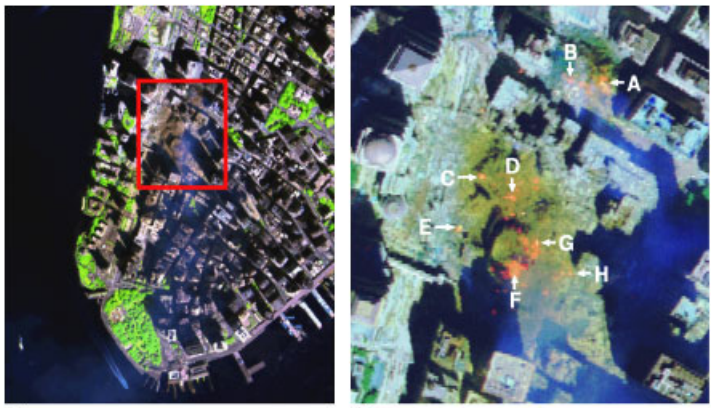


Figure 8. (a) False color composite of an AVIRIS hyperspectral image collected over lower Manhattan on September 16, 2001 and (b) location of thermal hot spots in the fires observed in World Trade Center area.

(in light gray) are located south in lower Manhattan, whereas burned areas appear dark gray. Figure 8(a) also shows smoke coming from the WTC area (in the rectangle) and going south to Battery Park in lower Manhattan.

At the time of data collection, a small U.S. Geological Survey (USGS) field crew visited lower Manhattan to collect spectral samples of dust and airfall debris deposits from several outdoor locations around the WTC area. These spectral samples were then mapped onto the AVIRIS data using the USGS Tetracorder method [28], supported by reflectance spectroscopy and chemical analyses in specialized laboratories. For illustrative purposes, Figure 8(b) (available from http://pubs.usgs.gov/of/2001/ofr-01-0429/hotspot.key.tgif.gif) shows a thermal map centered at the region where the buildings collapsed. The map shows the target locations of the thermal hot spots, shown as bright red, orange, and yellow spots on Figure 8(b). The temperatures range from 700°F (marked as 'F') to 1300°F (marked as 'G'). This thermal map, along with a dust/debris map produced by USGS (available online from the following URL: http://pubs.usgs.gov/of/2001/ofr-01-0429/wtc.Sept16.2001.usgs.r091011.plume1+bw.lowermhtn.tgif.gif), will be used in this work as ground-truth to validate the classification accuracy of parallel algorithms.

4.2.2. AVIRIS hyperspectral data over Indiana's Indian Pines region

A second hyperspectral scene was used to evaluate the performance of parallel algorithms in the context of land-cover classification applications. The scene was gathered by AVIRIS over the Indian Pines test site in Northwestern 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).

In this example, the spatial resolution was much coarser than in the AVIRIS WTC data set in Figure 8, with 20 m per pixels. As a result, the AVIRIS Indian Pines data set represents a very

DOI: 10.1002/cpe

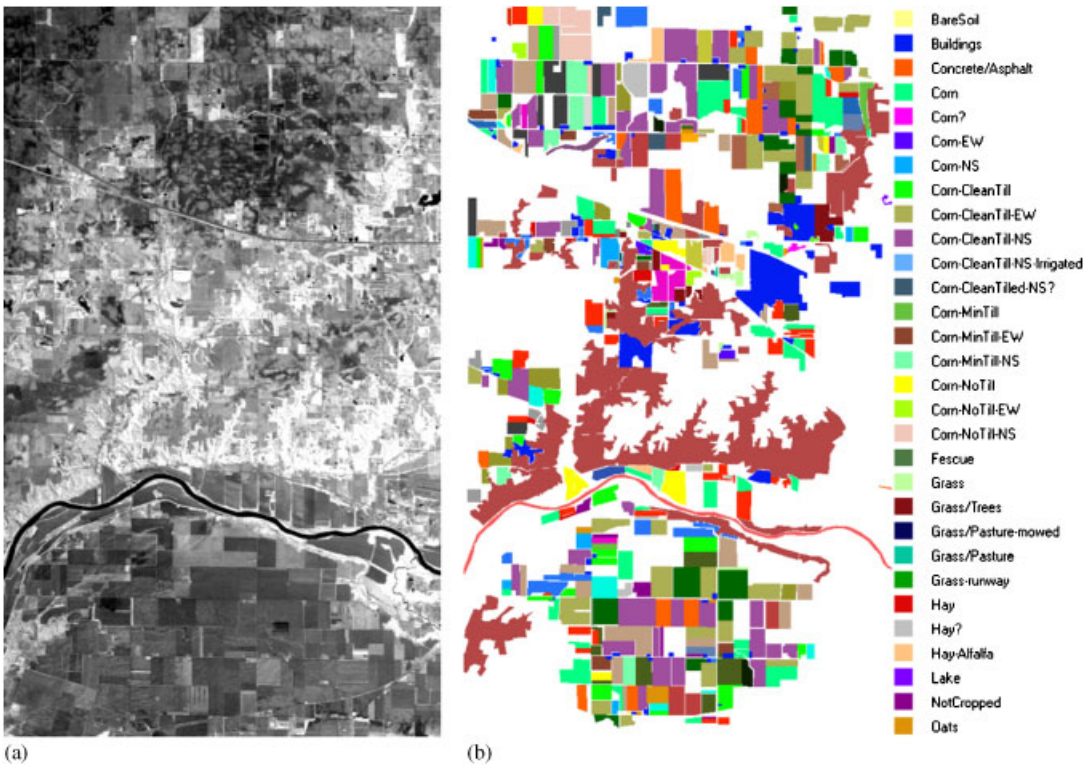


Figure 9. (a) Spectral band at 587 nm wavelength of an AVIRIS scene comprising agricultural and forest features at Indian Pines region and (b) ground-truth map with 30 mutually exclusive land-cover classes.

challenging classification problem dominated by similar spectral classes and mixed pixels. In particular, 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 [4,6,8,15]. For illustrative purposes, Figure 9(a) shows the spectral band at 587 nm of the original scene and Figure 9(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://dynamo.ecn.purdue.edu/~biehl/MultiSpec).

4.2.3. AVIRIS hyperspectral data over the Cuprite mining district in Nevada

Finally, an 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

DOI: 10.1002/cpe

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 0.4–2.5 μm. As opposed to the two previously considered data sets, the Cuprite scene is atmospherically corrected and available in reflectance units, thus allowing direct comparison of pixel vectors to ground spectral signatures. In this work, we have used a subset of the data, retaining the last 50 spectral bands covering the short-wave infrared (SWIR) region of the spectrum, where the differences between mineral features are clear. As a result, the image size in this example was about 30 MB. The Cuprite site has been extensively mapped by USGS in the last 20 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). Figure 10(a) shows the spectral band at 587 nm wavelength of the AVIRIS scene. The spectra of USGS ground minerals namely, alunite, buddingtonite, calcite, kaolinite, muscovite [Figure 10(b)], chlorite, jarosite, montmorillonite, nontronite, pyrophilite [Figure 10(c)], are also displayed. These selected spectral signatures will be used in this work to evaluate endmember extraction accuracy.

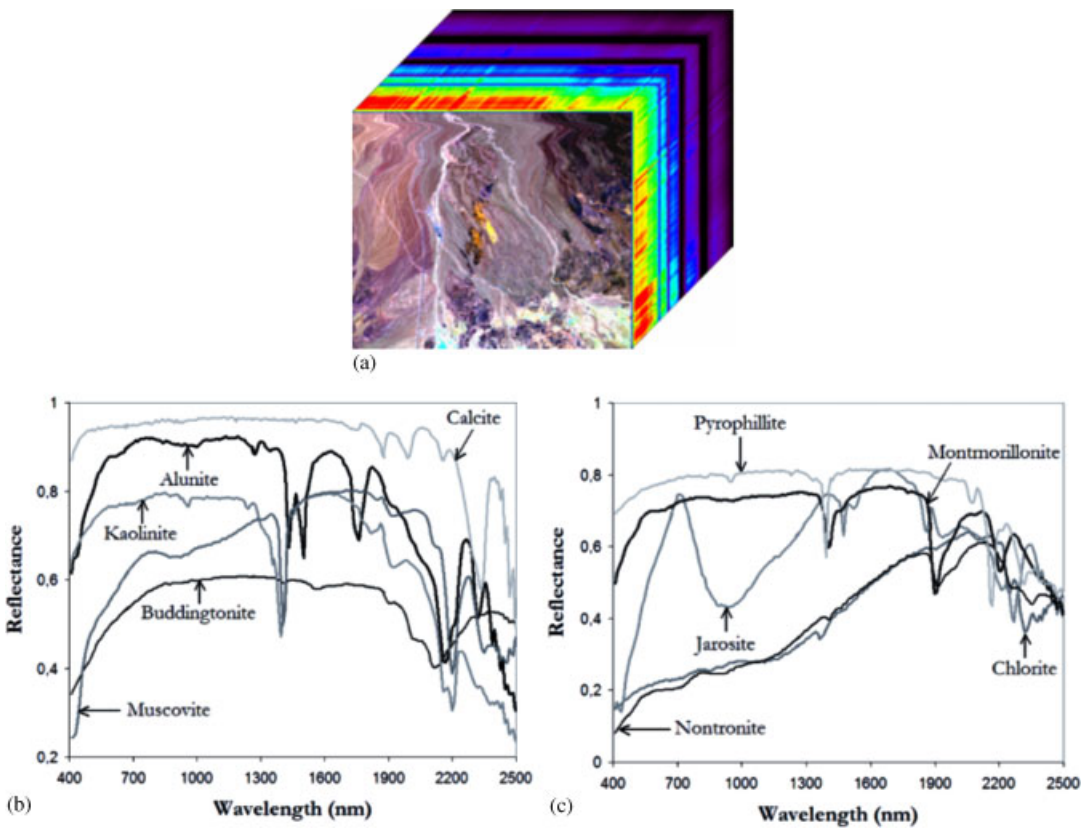


Figure 10. (a) AVIRIS scene over Cuprite mining district and (b–c) ground-truth mineral spectra provided by USGS.

DOI: 10.1002/cpe

4.3. Experimental assessment of parallel algorithms

4.3.1. Experiment 1: urban area characterization

In this subsection, we discuss the performance of the proposed parallel algorithms in the context of an urban application using the AVIRIS hyperspectral data set displayed in Figure 8(a). In order to validate the experimental results provided by those algorithms, we resort to ground-truth measurements collected by USGS over the WTC site. Specifically, we use the dust/debris map produced by the Tetracorder (http://pubs.usgs.gov/of/2001/ofr-01-0429/wtc.Sept16.2001.usgs.r091011.plume1+bw.lowermhtn.tgif.gif), as a reference map of materials in the WTC site. It should be noted that this map was rigorously produced by USGS using ground spectral samples collected on the field, and then using the Tetracorder algorithm to produce high-quality labels for high-resolution pixels in this scene. Additional information on the different classes available in our reference map and on the procedure for determining this reference information is available online from http://pubs.usgs.gov/of/2001/ofr-01-0429/dustplume.html.

Several different combinations were tested for the considered parallel full-pixel and mixed-pixel classification algorithms in the considered application domain. In order to assess the classification accuracy of full-pixel classification techniques, we selected the P-ISODATA as a representative case study and run the algorithm using three different configurations:

1. First, we run the P-ISODATA on the entire N-dimensional hyperspectral data cube.
2. Then, we run P-ISODATA on a p-dimensional, reduced data set obtained after applying the P-PCT algorithm to the original data cube. In this experiment, we set $p = 15$ and retained the first 15 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 [29], which has been demonstrated in previous work to be an accurate technique for estimating the number of endmembers in hyperspectral data [30].
3. Finally, we run the P-ISODATA on a p-dimensional, reduced data set obtained after applying the P-MORPH algorithm (with $t = 8$, resulting in $2t - 1 = 15$ components after morphological, spatial/spectral dimensionality reduction).

Similarly, in order to assess the classification accuracy of mixed-pixel classification techniques, we performed the following experiments:

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 p (with $p = 15$ based on the estimation provided by the VD), thus obtaining a set of 15 spectral endmembers in both cases. These endmembers were then fed to the P-LSU algorithm, which was used to generate a class label for each pixel using the WTA criterion.
2. Then, we repeated exactly the same experiment reported above, but this time we used P-MORPH instead of P-PCT to perform feature extraction. Here, we used $t = 8$, resulting in $2t - 1 = 15$ components, which is consistent with the dimensionality estimation provided by the VD concept.

In all cases, we carefully adjusted algorithm parameters based on our experience with those algorithms in different application domains [6,22,26]. With the above experimental settings in

DOI: 10.1002/cpe

mind, Table I reports the individual and overall classification accuracies (with regards to the USGS reference classifications) produced by the parallel algorithms after using different algorithm configurations. As shown in Table I, both the P-PPI and P-FINDR (combined with P-LSU) produced higher classification accuracies than those found by P-ISODATA. Interestingly, when P-ISODATA was combined with P-MORPH for dimensionality reduction, the classification accuracies increased significantly with respect to the cases in which the same algorithm was applied to the original image or to a reduced version of the image using P-PCT. This fact reveals that the incorporation of spatial information to the feature extraction stage may offer important advantages for classification, in particular, when the spatial information is an added value for the analysis, as it seems to be the case in the considered urban environment. A similar effect was observed for the P-PPI and P-FINDR

Table I. Individual (per-class) and overall percentages of correctly classified pixels in the AVIRIS WTC scene by different combinations of parallel algorithms.

	Full-pixel classification			Mixed-pixel classification			
	P-ISODATA (original)	P-PCT+ P-ISODATA	P-MORPH+ P-ISODATA	P-PCT+ P-PPI+ P-LSU	P-MORPH+ P-PPI+ P-LSU	P-PCT+ P-FINDR+ P-LSU	P-MORPH+ P-FINDR+ P-LSU
Processing time (s)	49 912	38 025+ 3340 = 41 365	40 239+ 3352 = 43 591	38 025+ 9456+ 2308 = 49 789	40 239+ 9456+ 2308 = 52 003	38 025+ 13 188+ 2308 = 53 521	40 239+ 13 188+ 2308 = 55 735
Ground-truth class							
Concrete (WTC01-37B)	77.21	71.23	85.42	90.34	96.32	95.61	95.93
Concrete (WTC01-37Am)	72.43	67.34	79.21	87.90	92.88	92.54	93.27
Cement (WTC01-37A)	61.05	65.28	77.43	70.06	90.23	88.45	91.05
Dust (WTC01-15)	58.72	62.05	75.20	68.15	82.34	88.93	90.48
Dust (WTC01-28)	58.03	61.97	77.89	67.00	89.23	91.02	90.92
Dust (WTC01-36)	69.09	65.25	84.43	75.78	88.45	90.23	91.42
Gypsum wall board	67.81	64.27	82.99	73.29	93.29	89.34	92.24
Hot spot 'A'	70.23	67.03	84.24	92.34	93.01	93.05	94.12
Hot spot 'B'	66.78	66.54	82.48	90.23	94.56	91.48	94.70
Hot spot 'C'	68.44	68.32	83.09	88.67	90.25	89.12	89.35
Hot spot 'D'	69.32	67.44	83.58	89.27	89.85	89.03	92.48
Hot spot 'E'	61.25	65.48	79.23	87.72	89.43	88.40	91.67
Hot spot 'F'	60.03	59.29	78.67	93.54	94.77	93.01	95.52
Hot spot 'G'	67.99	70.45	80.56	91.08	95.29	94.14	94.60
Overall accuracy	69.84 ± 9.59	66.97 ± 5.97	82.45 ± 5.11	89.93 ± 11.67	92.05 ± 6.99	91.04 ± 3.61	93.12 ± 3.29

Sequential execution times (in seconds), measured in a single node of NASA's Thunderhead cluster, are also reported for each algorithm combination.

DOI: 10.1002/cpe

algorithms, which resulted in higher classification accuracies when P-MORPH was used for feature extraction instead of P-PCT. We note that both PPI and N-FINDR include dimension reduction as the first algorithm step and therefore our parallel versions were not applied to the original data cube.

For illustrative purposes, Table I also reports the execution times (in seconds) measured for the different algorithms using a single processor of the Thunderhead parallel system. We emphasize that these times correspond to real sequential versions of the considered algorithms. As shown by the table, the two considered dimension reduction techniques were computationally expensive but led to significant reductions in the processing times of other algorithms (e.g. P-ISODATA). We note that, in this example, the number of bands was reduced from more than 200 to only 15. 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, Figure 11(a) plots their speedup factors as a function of the number of available processors on Thunderhead (only 256 processors were available to us at the time of experiments). Results in Figure 11(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, e.g. step 4 (eigenvector calculations) in the P-PCT or step 6 (split and merge) in the P-ISODATA. It should be noted that the complexity of the eigenvector calculations is related to the number of spectra used in the problem, whereas the split and merge are highly dependent on the inherent complexity of the input data. On the other hand, we can observe in Figure 11(a) that the P-PPI and P-FINDR algorithms resulted in closer-to-linear speedups.

It can also be seen from Figure 11(a) that both P-LSU and P-MORPH were the two algorithms that achieved better scalability on Thunderhead. These results come as no surprise since the algorithms were already identified as 'pleasingly parallel' (despite the use of overlap borders in the data partitioning procedure adopted for P-MORPH), and do not involve many data dependencies. Although for a high number of nodes the speedup graphs flatten out a little—mainly due to the relatively short execution times—the spatial/spectral feature extraction and mixed-pixel classification algorithms reveal themselves in this example as highly scalable approaches able to outperform their counterparts, i.e. P-ISODATA (for full-pixel classification) and P-PCT (for spectral-based feature extraction), respectively. This observation also resulted in faster execution times for these algorithms as reported in Figure 11(b), which also displays the average size of the partitions processed by the workers in each run.

In addition to scalability and parallel execution times, we have also evaluated the efficiency of the proposed parallel algorithms considering two important new topics: the computational load balance and the cost of communications. In order to measure load balance, Figure 11(c) shows the load balancing rates achieved by the parallel algorithms, defined as $D = R_{max}/R_{min}$ [31], where R_{max} and R_{min} are the maxima and minima processing times across all the processors, respectively. Therefore, perfect balance is achieved when $D = 1$. As shown in Figure 11(c), all parallel algorithms were able to provide values of D, which were close to 1 in most cases, with the exception of P-ISODATA and P-PCT. We experimentally tested that the most important contribution to the imbalance observed

DOI: 10.1002/cpe

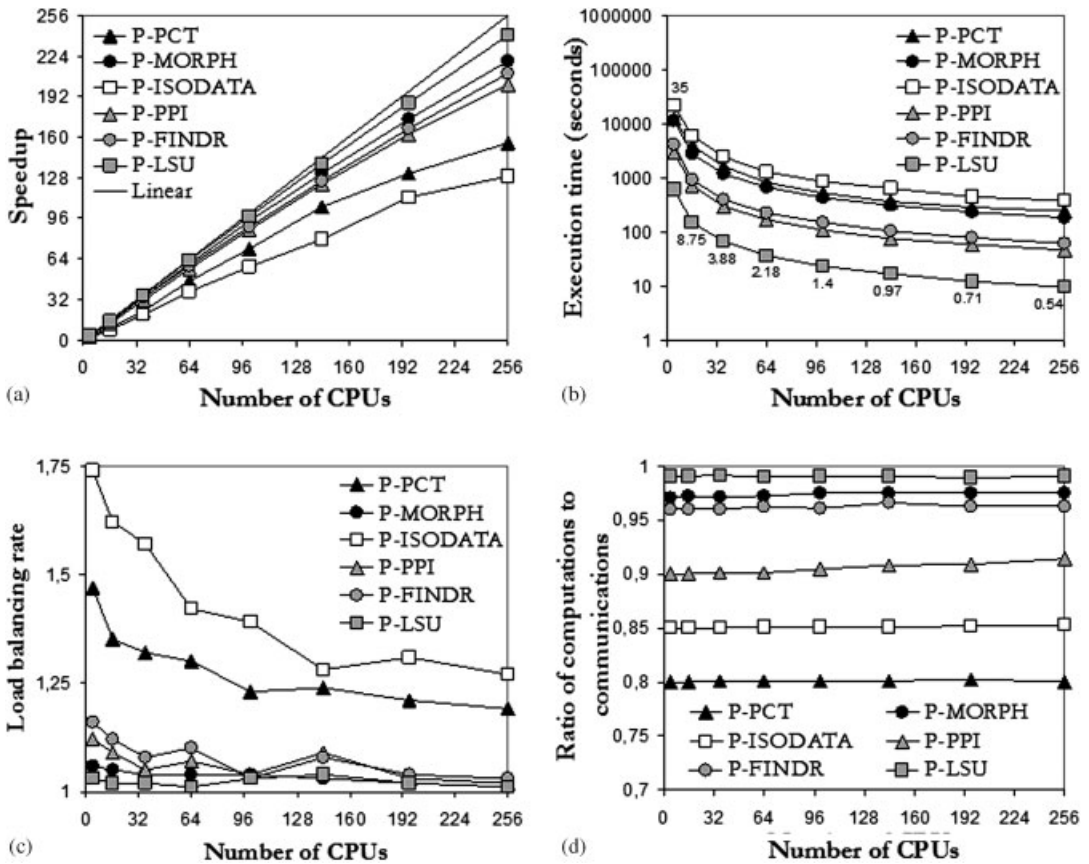


Figure 11. Performance results after processing the AVRIS WTC scene with the proposed parallel algorithms: (a) parallel algorithm scalability; (b) computation times as a function of the number of CPUs and the partition size; (c) load balancing rates; and (d) ratio of computations to communications.

in those algorithms originated from the high computational load of the master due to sequential computations. It is also clear from Figure 11(c) that load balance was much better for the P-MORPH and P-LSU. As a result, the combination of P-MORPH for feature extraction, followed by any of P-PPI or P-FINDR for endmember extraction, plus spectral unmixing via P-LSU, resulted in a well-balanced (and highly scalable) parallel combination, which outperformed any combination involving the P-ISODATA algorithm for full-pixel classification.

Finally, Figure 11(d) shows the ratio of computations to communications for all the parallel algorithms tested in this example. Two types of computation times were included in the ratio, i.e. sequential computations (those performed by the master node *with no other parallel tasks active in the system*) and parallel computations (those performed by the master node and/or the workers *in parallel*, including the times in which the workers remain idle). It can be seen from Figure 11(d) that both the P-PCT and P-ISODATA required to communicate more often than the

DOI: 10.1002/cpe

other parallel algorithms tested. These algorithms had comparatively more data dependencies, as shown by the algorithmic descriptions provided in Section 3. On the other hand, P-PPI required communicating more often than P-FINDR. Finally, both P-LSU and P-MORPH showed the highest ratio of computations to communications in Figure 11(d). Again, this comes as no surprise since these algorithms greatly reduced the number of data dependencies (in the latter algorithm, this is achieved at the expense of introducing a small amount of redundant computations which did not significantly affect the optimality of the measured computation to communication ratios).

Summarizing, experimental results in this subsection reveal that important improvements can be obtained from the viewpoint of computational performance by using parallel algorithms in the framework of the considered urban application. The measured execution times were deemed suitable for rapidly providing emergency response teams with information on the presence of fires and the distribution of debris and other materials in the dusts deposited around the WTC area. For instance, the P-MORPH + P-PPI + P-LSU combination was able to provide a highly accurate mapping in 4 min using 256 processors, as opposed to more than 14 h of computation required by the same combination on a single Thunderhead processor. Despite these encouraging results, further experimentation in other application areas (with lower spatial resolutions) is required to establish the statistics obtained in this case study.

4.3.2. Experiment 2: land-cover classification in agriculture

The second case study presented in this work intends to address an application example in which lower spatial resolution significantly complicated the classification problem due to the presence of highly mixed pixels. In order to evaluate the performance of the proposed parallel algorithms in a more challenging classification scenario, we have repeated exactly the same experiments conducted for the AVIRIS WTC scene with the AVIRIS Indian Pines scene, which is about four times larger in size and has lower spatial resolution (20-m pixels as opposed to 1.7-m pixels in the previous example). Table II reports the classification accuracies and sequential execution times obtained for some selected ground-truth classes (it should be noted that the overall accuracies refer to the entire set of 30 ground-truth classes available, not displayed for space considerations). Here, we set $p = 41$ based on the dimensionality estimation provided by the VD, and parameter t was set to $t = 21$, resulting in $2t - 1 = 41$ components. The fact that the VD estimated more than 30 classes for the AVIRIS Indian Pines scene was not surprising, since Figure 9(b) reveals that the 30 available ground-truth classes only cover about 60% of the entire scene.

As shown in Table II, the overall classification accuracies achieved by parallel algorithms decreased with regards to the previous example. Interestingly, the considered mixed-pixel classification algorithms performed much better than full-pixel classification algorithms in this example, indicating that pure pixel-based classification strategies cannot accurately separate highly similar spectral classes in this particular case study. In fact, many of the crops in the agricultural fields of the Indian Pines test site were very early in their growth cycle when the AVIRIS image was taken, resulting in soil-like mixed pixels, which was indeed the dominant endmember in many of the ground-truth classes displayed in Figure 9(b). As a result, the incorporation of spatial information was particularly useful to complement spectral information (less informative in this example due to pixel mixing), and hence the parallel algorithms using P-MORPH for feature extraction prior to classification resulted in much better classification accuracies, as reported in Table II.

DOI: 10.1002/cpe

Table II. Individual (per-class) and overall percentages of correctly classified pixels in the AVIRIS Indian Pines scene by different combinations of parallel algorithms.

	Full-pixel classification			Mixed-pixel classification			
	P-ISODATA (original)	P-PCT+ P-ISODATA	P-MORPH+ P-ISODATA	P-PCT+ P-PPI+ P-LSU	P-MORPH+ P-PPI+ P-LSU	P-PCT+ P-FINDR+ P-LSU	P-MORPH+ P-FINDR+ P-LSU
Processing time (s)	202 992	152 860+ 37 150 = 190 010	162 163+ 37 168 = 199 331	152 860+ 44 916+ 11 078 = 208 854	162 163+ 44 916+ 11 078 = 218 157	152 860+ 63 566+ 11 078 = 227 504	162 163+ 63 566+ 11 078 = 236 807
Ground-truth class							
Bare Soil	60.32	62.34	77.81	85.44	90.24	91.35	91.67
Buildings	52.45	52.89	65.23	79.23	84.31	82.56	88.23
Concrete/ asphalt	58.33	61.12	75.94	80.45	85.00	85.23	90.35
Corn	51.23	54.27	67.01	78.33	84.93	84.08	86.59
Fescue	56.03	55.03	69.23	79.05	86.20	84.75	86.04
Grass	62.37	60.48	73.91	83.44	89.88	86.23	87.24
Hay	55.15	53.49	72.16	82.67	88.04	81.23	85.43
Lake	58.23	59.25	73.45	82.94	87.76	82.67	87.95
Oats	53.12	56.78	69.32	80.58	85.23	83.03	90.66
Overall accuracy	55.48 ± 5.57	57.21 ± 4.72	72.26 ± 6.29	82.93 ± 3.55	87.91 ± 2.96	83.45 ± 5.06	88.05 ± 3.12

Sequential execution times (in seconds) measured in a single node of NASA's Thunderhead cluster are reported for each algorithm combination.

Finally, we experimentally observed that the computation times measured for the parallel algorithms in this example scaled linearly with regard to the times reported in the previous example. For instance, the measured execution time after applying the P-ISODATA to the entire AVIRIS Indian Pines scene was approximately four times higher than the time reported in Table I for the experiment with the AVIRIS WTC scene (approximately four times smaller in size). On the other hand, the number of components retained after dimensionality reduction in this example was 41 as opposed to 15 in the previous example, and therefore the speedup achieved by the sequential version of P-ISODATA applied to the data cube reduced by P-PCT was in the order of $224/41 = 5.4$ instead of $224/15 = 15.9$ observed for the same experiment in Table I. Most importantly, we experimentally tested that the speedups achieved by the parallel algorithms in this example, as well as the load balancing rates and computation to communication ratios, followed a very similar pattern than those reported in Figure 11 for the AVIRIS WTC scene (since the plots are very similar, we do not display them here for space considerations). Although our experimental results in this case study revealed that an increase in the volume of data to be processed by the parallel algorithms generally results in a linear increase in the parallel execution times (regardless of the complexity of the application environment), further experimentation using smaller hyperspectral scenes containing features without significant spatial correlation is still highly desirable in order to fully substantiate the context in which the above-mentioned remarks are valid.

DOI: 10.1002/cpe

4.3.3. Experiment 3: mineral mapping

In this final experiment, we have conducted a cross-validation of parallel endmember extraction algorithms in the context of a mineral mapping application, using the well-known AVIRIS Cuprite data set for demonstration purposes. Since we used only the last 50 spectral bands of the data, the total size of the image in this example is 30 MB. 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. 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 p (with $p = 15$), obtaining a set of 15 spectral endmembers in both cases. This value was obtained using the VD concept, which was applied to the 50-band data set to estimate its dimensionality.
2. Then, we repeated the previous experiment but this time using P-MORPH instead of P-PCT to perform feature extraction from the input 50-band scene. Here, we used $t = 8$, resulting in $2t - 1 = 15$ components, which is consistent with the dimensionality estimation provided by the VD concept.

Table III 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 with 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

Table III. 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-PPI	P-MORPH + P-PPI	P-PCT + P-FINDR	P-MORPH + P-FINDR
Time (s)				
Endmember	8150 + 2035 = 10 185	8619 + 2035 = 10 654	8150 + 2824 = 10 974	8619 + 2824 = 11 443
Alunite	**0.084**	0.099	0.112	0.099
Buddingtonite	0.106	0.106	**0.095**	0.106
Calcite	**0.105**	0.116	0.112	0.118
Kaolinite	**0.136**	0.151	0.151	0.151
Muscovite	0.108	0.108	**0.106**	0.112
Chlorite	0.125	0.136	**0.096**	0.136
Jarosite	0.112	0.115	**0.108**	0.115
Montmorillonite	0.106	0.106	**0.096**	0.106
Nontronite	0.102	0.106	**0.099**	0.108
Pyrophilite	0.094	0.098	**0.090**	0.098

DOI: 10.1002/cpe

smaller SAD values indicate higher spectral similarity. As shown in Table III, the P-PCT+P-PPI and P-PCT+P-FINDR result in the largest number of minimal SAD values (displayed in bold typeface) among all considered combinations. On the contrary, the combinations that used P-MORPH for feature extraction prior to endmember extraction generally produced endmembers that were less similar, spectrally, with regard 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. This results from the fact that geological features in the Cuprite mining district appear quite scattered, and exhibit little spatial correlation. Therefore, it is not surprising that the performance of parallel endmember extraction algorithms in this example was better when they were combined with P-PCT instead of P-MORPH. This behavior is completely contrary to that observed in experiments with hyperspectral data collected over urban and agricultural areas, in which higher spatial correlation of observed image features was apparent.

On the other hand, although the processing times reported in Table III for the different algorithm combinations tested were significantly smaller than those reported in the previous two examples, the smaller size of the hyperspectral data considered in this example introduced some scalability problems, as reported in Figure 12(a). Here, we can observe that neither the parallel dimensionality reduction algorithms nor the parallel endmember extraction algorithms could scale as effectively as in the two previous examples. This is because, given the relatively small size of the image considered in this example, an increase in the number of processors results in very small average partition sizes [see Figure 12(b)], and the ratio of computations to communications will also decrease, as it can be observed in Figure 12(d). It has been recently shown in [32] that a very likely explanation for the above situation results from the fact that execution times measured at the different processors may escalate non-linearly for medium-sized communication messages, thus introducing some fluctuations in load balance as observed in Figure 12(c). Although this situation seems not to be critical in this example [see Figure 12(b)], experimental results in this subsection reveal that a better model for I/O operations may be required in order to improve the scalability of parallel hyperspectral imaging algorithms on massively parallel platforms, in which very small-sized partitions will be produced when the input data are distributed among a very large number of processors.

To conclude this subsection, we would like to emphasize that the proposed combinations of parallel algorithms have also been evaluated in terms of their capacity to produce high-quality abundance estimations for geological features in the Cuprite mining district. This experiment has been performed by estimating the fractional abundance of endmembers provided by P-PPI or P-FINDR, using different variations of the proposed P-LSU algorithm. The two tested variations of P-LSU were fully constrained LSU (i.e. linear spectral unmixing with abundance sum-to-one and abundance non-negativity constraints), and unconstrained LSU. 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 obtain in real-world scenarios), our quantitative experiments demonstrated that the use of P-PCT for dimensionality reduction prior to endmember extraction generally resulted in very similar abundance fractions estimated by both fully constrained and unconstrained LSU. A common indicator of poor model fitting and/or inappropriate selection of endmembers is estimation of negative abundance fractions by unconstrained LSU. This situation was very rarely observed when P-PCT was used for dimensionality reduction. In contrast, a more significant fraction of negative abundances was obtained by the unconstrained LSU when P-MORPH was used for feature extraction prior to endmember extraction by either P-PPI or P-FINDR. This

DOI: 10.1002/cpe

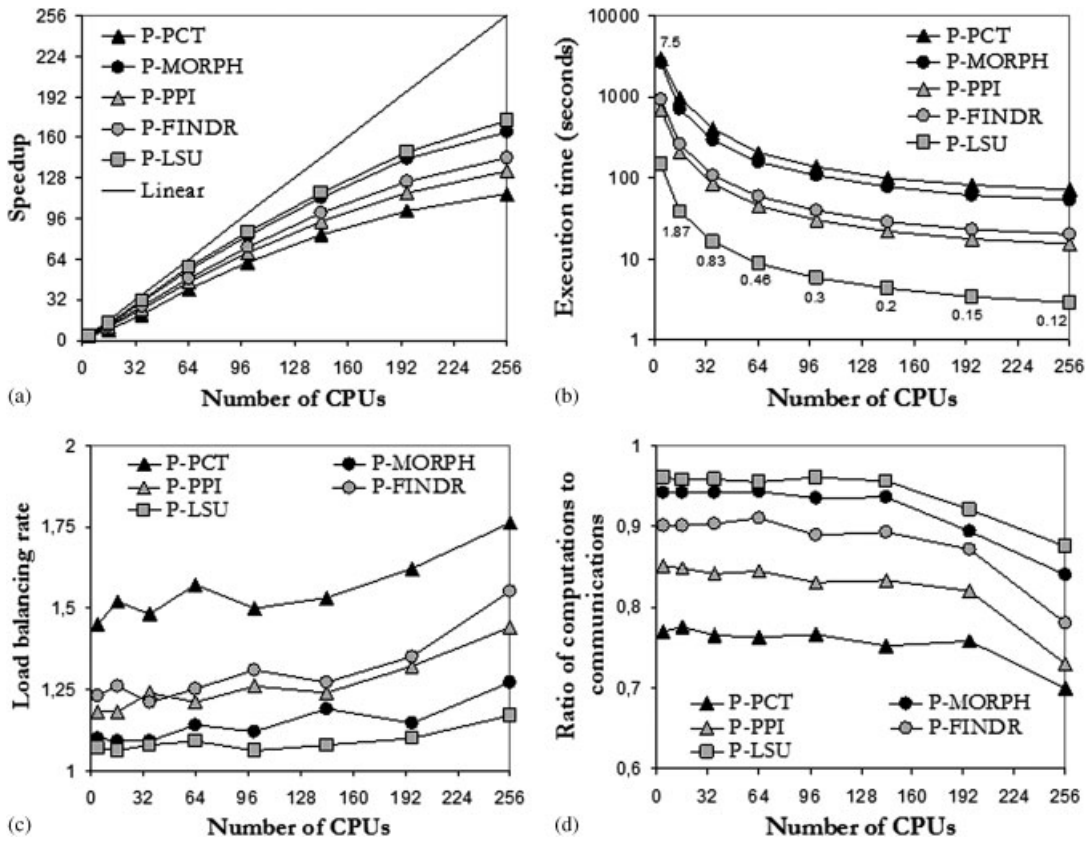


Figure 12. Performance results after processing AVIRIS Cuprite scene: (a) algorithm scalability; (b) computation times versus number of CPUs and partition sizes; (c) load balancing; and (d) ratio of computations to communications.

result is consistent with those found in our evaluation of endmember extraction accuracy, indicating that P-MORPH slightly underperforms in this particular application.

4.4. Discussion

In previous subsections, we have thoroughly analyzed the performance of a suite of parallel algorithms for hyperspectral image analysis in the context of three different application domains. As a follow-up to these analyses, in this subsection we first intend to provide a quick summary of the main observations and lessons learned after each conducted experiment. For that purpose, Table IV summarizes the outcome of processing experiments conducted using parallel algorithms, including relevant aspects such as the specific properties of each considered data set, the algorithm combinations that performed best and worst in each application domain, and also addressing the algorithm

DOI: 10.1002/cpe

Table IV. Summary of processing experiments using the proposed parallel techniques in different application areas.

	Urban area characterization	Land-cover classification in agriculture	Mapping of mineral features
Image size (MB)	140	574	30
Spatial resolution (meters per pixel)	1.7	20	20
Spectral resolution	224 bands (0.4–2.5 μm)	224 bands (0.4–2.5 μm)	50 bands (2.0–2.5 μm)
Data dimensionality	15 (estimated by VD)	41 (estimated by VD)	15 (estimated by VD)
Atmospheric correction	No (radiance data)	No (radiance data)	Yes (reflectance data)
Geometric correction	Yes (performed by JPL)	Yes (performed by JPL)	Yes (performed by JPL)
Ground-truth	USGS classification map	Purdue classification map	USGS signature library
Classifiers compared	Full- and mixed-pixel	Full- and mixed-pixel	Mixed-pixel
Best classification algorithm	P-MORPH + P-FINDR + P-LSU	P-MORPH + P-FINDR + P-LSU	P-PCT + P-FINDR
Second best algorithm	P-MORPH + P-PPI + P-LSU	P-MORPH + P-PPI + P-LSU	P-PCT + P-PPI
Worst classification algorithm	P-PCT + P-ISODATA + P-LSU	P-ISODATA on original image	P-MORPH + P-FINDR
Best parallel processing time (256 Thunderhead processors)	4 min (P-MORPH + P-PPI + P-LSU)	17 min (P-MORPH + P-PPI + P-LSU)	1.11 min (P-MORPH + P-PPI)
Second best processing time (256 Thunderhead processors)	4.1 min (P-MORPH + P-PPI + P-LSU)	17.5 min (P-MORPH + P-PPI + P-LSU)	1.18 min (P-MORPH + P-FINDR)
Worst parallel processing time (256 Thunderhead processors)	10.6 min (P-ISODATA on original image)	46.7 min (P-ISODATA on original image)	1.48 min (P-PCT + P-FINDR)

processing times measured on Thunderhead. It should be noted that, although we acknowledge the vast amount of recent literature devoted to the design of supervised classifiers for hyperspectral image analysis [8], our focus in this paper has been on unsupervised techniques. With the above considerations in mind, we believe that the compendium of parallel processing techniques and their detailed cross-validation in the context of real application domains, summarized in Table IV, 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 author's best knowledge:

- Contrary to the common perception that spatial/spectral and mixed-pixel hyperspectral image classification algorithms involve more complex operations than traditional, full-pixel classification techniques, results in this paper indicate that mixed-pixel techniques, when carefully designed and implemented, can indeed be more 'pleasingly parallel' than standard full-pixel classification techniques, mainly because they can reduce sequential computations at the

DOI: 10.1002/cpe

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 full- and mixed-pixel classification techniques is directly linked to the application domain. Spatial/spectral techniques generally performed accurately in applications involving data sets with high spatial resolution and features with spatial auto-correlation. On the other hand, applications in which the relevant image features are not correlated generally benefit from the use of more spectrally guided approaches.
- Although in previous work there has been a clear distinction between endmember extraction and classification algorithms, experimental results in this work reveal that endmember extraction algorithms can also be successfully used for classification purposes. In this work, this was achieved by applying a simple WTA criterion to the fractional abundances estimated using LSU.
- A final important observation from experiments is the fact that the performance of parallel hyperspectral algorithms strongly depends on the dimensionality of the input data set. For large and complex data cubes, the proposed parallel algorithms scaled very well, with speedup ratios close to optimal. However, the performance of parallel algorithms was progressively degraded as the size of the input data was decreased, a situation that is also expected to happen when large images are processed using a very large number of nodes in the underlying parallel platform. In this case, better communication models are required.

In order to address the above-mentioned concern, which may result in important scalability problems when implementing the proposed parallel algorithms on massively parallel platforms, we have conducted preliminary experiments and found that this situation requires an optimization of I/O and communications. In fact, the performance decrease experienced by all parallel algorithms can be explained by carefully analyzing the nature of our MPI-based parallel implementations, in which a master processor sends the partitions of the original image, using the `MPI_Scatter` operation, to a group of worker processors. The partitions are then processed in parallel and processing results are returned to the master using the `MPI_Gather` operation. Although fine tuning of the parallel algorithms for efficient performance in large-scale commodity clusters is out of the scope of this paper, the scalability problem mentioned above can be approached by resorting to a recently proposed model for many-to-one communication operations such as those involved in our `MPI_Gather` operations [32]. This model distinguishes between small, medium and large messages by introducing two threshold parameters, T_1 and T_2. For small messages (of size below T_1), the execution time is shown to respond linearly to the increase of message sizes in Beowulf clusters. For medium-sized messages (of size between T_1 and T_2), a very significant degradation in performance due to nonlinear effects is found empirically. Finally, for large messages (of size larger than T_2), the execution time resumes linear predictability. In our application we have experimentally tested that, when a small number of processors is used, the message size generally fits into the area of large messages. On the contrary, when the number of processors is increased, the message size may fit into the area of medium-sized messages, falling into a congestion region that ultimately increases the execution times of the single `MPI_Gather` operation.

In order to address this issue, we can redesign our parallel algorithms as follows. If we replace the single `MPI_Gather` operation used to gather medium-sized messages by an equivalent sequence

DOI: 10.1002/cpe

of `MPI_Gather` operations, each gathering messages with a size that fits the range of small messages, then each medium-sized partition can be communicated as a sequence of small-sized sub-partitions, thus avoiding the above-mentioned scalability problem. To achieve this goal, an option is to calculate the number of sub-partitions s of a partition of medium size, S, so that $S/s \leq T_1$ and $S/(s-1) \geq T_1$. An alternative option is to modify the system settings for MPI and the TCP/IP flags to allow a reservation mode to begin for smaller messages. If larger messages are required, then the non-buffered mode can be selected for MPI by setting the system buffer size to a smaller value. In this case, sporadic non-linear behavior may still be observed, but it is likely to occur at negligible probability [32]. Although we are planning on carefully exploring the above-mentioned issues in our future developments, access restrictions to configuration parameters and settings for the Thunderhead system, a U.S. Government facility, currently prevented us from implementing the above-mentioned optimizations in this work.

5. CONCLUSIONS AND FUTURE RESEARCH

In this paper, we have discussed the performance of several parallel algorithms for hyperspectral image analysis in the context of different application domains. Specifically, we have explored the performance of four types of hyperspectral imaging algorithms, i.e. dimensionality reduction, unsupervised classification, endmember extraction, and spectral unmixing, in the context of three different application domains, i.e. urban area characterization, land-cover classification in agricultural applications, and mapping of geological features. The cross-validation of parallel algorithms in different application domains conducted in this work represents an entirely novel contribution. Another important topic addressed by this paper is the design and implementation of a new parallel feature extraction algorithm that integrates the spatial and spectral information in the hyperspectral data using mathematical morphology concepts. A detailed analysis on the implications of using spatial/spectral feature extraction versus spectral-based dimensionality reduction prior to full-pixel and mixed-pixel classification has also been provided.

Our experimental assessment of parallel algorithms revealed important considerations about the properties and nature of such algorithms. For instance, the experiments in this work demonstrated that the use of redundant computations can significantly overcome the limitations introduced by parallel tasks that communicate very often. This introduces important considerations in parallel algorithm design, in particular, since parallel machines are now typically given a lot more memory per node. In this regard, performance results measured on the Thunderhead system at NASA's Goddard Space Flight Center not only indicate that parallel implementations of both full-pixel and mixed-pixel algorithms are able to provide adequate results in both the quality of the solutions and the time to obtain them but also reveal important aspects about the design of future parallel machines for executing this parallel algorithms (e.g. it is expected that future evolutions of the Thunderhead Beowulf cluster would even allocate more local memory resources per node). Although this type of parallel architecture has gained popularity in the last few years due to the chance of building a 'high performance system' at a reasonable cost, the proposed parallel implementations can be ported to any type of distributed memory system, including heterogeneous networks of workstations. Combining this readily available computational power with the new, *ultraspectral* capabilities of sensor instruments currently under development may introduce major changes in

DOI: 10.1002/cpe

the systems currently used by NASA and other agencies for exploiting Earth and planetary remotely sensed hyperspectral data. In this regard, the compendium of parallel techniques presented in this work 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 processing, and environmental applications.

As future work, we plan to implement the proposed parallel hyperspectral imaging algorithms on other high-performance parallel 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. We also envision closer multidisciplinary collaborations with environmental scientists to address global monitoring land services and security issues, which are the main concern of several on-going and planned remote sensing missions.

ACKNOWLEDGEMENTS

This research was supported by the European Commission through the project 'Hyperspectral Imaging Network' (contract no. MRTN-CT-2006-035927). Funding from the Spanish Ministry of Education and Science (Fellowship PR2003-0360), which allowed the author to conduct research as postdoctoral research associate at NASA's Goddard Space Flight Center and University of Maryland, is also gratefully acknowledged. The author would like to thank Drs John E. Dorband, James C. Tilton, and J. Anthony Gualtieri at NASA's Goddard Space Flight Center for their collaboration in the development of experiments on the Thunderhead cluster. Last but not least, the author gratefully thanks Dr Alejandro Curado from the Department of English at the University of Extremadura for his linguistic revision of the manuscript.

REFERENCES

1. Goetz AFH, Vane G, Solomon JE, Rock BN. Imaging spectrometry for Earth remote sensing. *Science* 1985; **228**: 1147–1153.
2. Monsch KA, Green RO, Eastwood ML, Sarture CM, Chrien TG, Aronsson M, Chippendale BJ, Faust JA, Pavri BE, Chovit CJ, Solis M, Olah MR, Williams O. Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)—a case study in the Landes Forest, South West France. *Remote Sensing of Environment* 1998; **65**:227–248.
3. Chang C-I. *Hyperspectral Imaging*: *Techniques for Spectral Detection and Classification*. Kluwer: New York, NY, 2003.
4. Plaza A, Chang C-I. *High Performance Computing in Remote Sensing*. Chapman & Hall/CRC Press: Boca Raton, FL, 2007.
5. Simpson JJ, McIntire TJ, Berg JS, Tsou YL. The parallel image processing environment (PIPE): Automated parallelization of satellite data analyses. *Concurrency and Computation*: *Practice and Experience* 2007; **19**:1–36.
6. Plaza A, Valencia D, Plaza J, Martínez P. Commodity cluster-based parallel processing of hyperspectral imagery. *Journal of Parallel and Distributed Computing* 2006; **66**:345–358.
7. Sterling T. Cluster computing. *Encyclopedia of Physical Science and Technology* (3rd edn), vol. 3. Academic Press: New York, 2002.
8. Richards JA, Jia X. *Remote Sensing Digital Image Analysis*: *An Introduction* (4th edn). Springer: Berlin, Germany, 2006.
9. Scott DW. The curse of dimensionality and dimension reduction. *Multivariate Density Estimation*: *Theory*, *Practice*, *and Visualization*, ch. 7. Wiley: New York, 1992; 195–217.
10. Plaza A, Martínez P, Plaza J, Pérez R. Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations. *IEEE Transactions on Geoscience and Remote Sensing* 2005; **43**:466–479.
11. Sano K, Kobayashi Y, Nakamura T. Differential coding scheme for efficient parallel image composition on a PC cluster system. *Parallel Computing* 2004; **30**:285–299.
12. Nicolescu C, Jonker P. A data and task parallel image processing environment. *Parallel Computing* 2002; **28**:945–965.

DOI: 10.1002/cpe

13. Seinstra FJ, Koelma D, Geusebroek JM. A software architecture for user transparent parallel image processing. *Parallel Computing* 2002; **28**:967–993.
14. 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* 2007; **40**:81–107.
15. El-Ghazawi T, Kaewpijit S, Le Moigne J. Parallel and adaptive reduction of hyperspectral data to intrinsic dimensionality. *Proceedings of the IEEE International Conference on Cluster Computing*, Newport Beach, CA, 2001; 102–109.
16. Achalakul T, Taylor S. A distributed spectral-screening PCT algorithm. *Journal of Parallel and Distributed Computing* 2003; **63**:373–384.
17. Diamantaras KI, Kung SY. *Principal Component Neural Networks*. Wiley: New York, NY, 1996.
18. Soille P. *Morphological Image Analysis*: *Principles and Applications* (2nd edn). Springer: Berlin, Germany, 2003.
19. Sternberg SR. Grayscale morphology. *Computer Vision*, *Graphics and Image Processing* 1986; **35**:333–355.
20. Benediktsson JA, Pesaresi M, Arnason K. Classification and feature extraction for remote sensing images from urban areas based on morphological transformations. *IEEE Transactions on Geoscience and Remote Sensing* 2003; **41**:1940–1949.
21. Dhodhi MK, Saghri JA, Ahmad I, Ul-Mustafa R. D-ISODATA: A distributed algorithm for unsupervised classification of remotely sensed data on network of workstations. *Journal of Parallel and Distributed Computing* 1999; **59**:280–301.
22. Plaza A, Martinez P, Perez RM, Plaza J. A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing* 2004; **42**(3):650–663.
23. Boardman JW, Kruse FA, Green RO. Mapping target signatures via partial unmixing of AVIRIS data. *Proceedings of the JPL Airborne Earth Science Workshop*, Pasadena, CA, 1995; 23–26.
24. Winter ME. N-FINDR: An algorithm for fast autonomous spectral endmember determination in hyperspectral data. *Proceedings of the SPIE—Image Spectrometry V*, Denver, CO, 1999; **3753**:266–277.
25. Heinz D, Chang CI. Fully constrained least squares linear mixture analysis for material quantification in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing* 2001; **39**:529–545.
26. Plaza A, Valencia D, Plaza J, Chang C-I. Parallel implementation of endmember extraction algorithms from hyperspectral data. *IEEE Geoscience and Remote Sensing Letters* 2006; **3**:334–338.
27. Simon AJ. A model for biological winner-take-all neural competition employing inhibitory modulation of NMDA-mediated excitatory gain. *Neurocomputing* 1999; **26**:587–592.
28. Clark RN, Swayze GA, Livo KE, Kokaly RF, Sutley SJ, Dalton JB, McDougal RR, Gent CA. Imaging spectroscopy: Earth and planetary remote sensing with the USGS Tetracorder and expert systems. *Journal of Geophysical Research* 2003; **108**:1–44.
29. Du Q, Chang C-I. Estimation of number of spectrally distinct signal sources in hyperspectral imagery. *IEEE Transactions on Geoscience and Remote Sensing* 2004; **42**:608–619.
30. Plaza A, Chang C-I. Impact of initialization on design of endmember extraction algorithms. *IEEE Transactions on Geoscience and Remote Sensing* 2006; **44**:3397–3407.
31. Martín MJ, Singh DE, Mouriño JC, Rivera FF, Doallo R, Bruguera JD. High performance air pollution modeling for a power plan environment. *Parallel Computing* 2003; **29**:1763–1790.
32. Lastovetsky A, O'Flynn M. A performance model of many-to-one collective communications for parallel computing. *Proceedings of the IEEE Parallel and Distributed processing Symposium* (*IPDPS*), Long Beach, CA, 2007; 475–483.

DOI: 10.1002/cpe