

# Recent developments and future directions in hyperspectral data classification

Antonio J. Plaza<sup>a</sup>

<sup>a</sup>Department of Technology of Computers and Communications  
University of Extremadura, Avda. de la Universidad s/n  
E-10071 Cáceres, Spain

## ABSTRACT

Imaging spectroscopy, also known as hyperspectral imaging, has been transformed in less than 30 years from being a sparse research tool into a commodity product available to a broad user community. As a result, there is an emerging need for standardized data processing techniques, able to take into account the special properties of hyperspectral data and to take advantage of latest-generation sensor instruments and computing environments. The goal of this paper is to provide a seminal view on recent advances in techniques for hyperspectral data classification. Our main focus is on the design of techniques able to deal with the high-dimensional nature of the data, and to integrate the spatial and spectral information. The performance of the proposed techniques is evaluated in different analysis scenarios, including land-cover classification, urban mapping and spectral unmixing. To satisfy time-critical constraints in many remote sensing applications, parallel implementations for some of the discussed algorithms are also developed. Combined, these parts provide a snapshot of the state-of-the-art in those areas, and offer a thoughtful perspective on the potential and emerging challenges in the design of robust hyperspectral data classification algorithms.

**Keywords:** Hyperspectral data classification, mathematical morphology, support vector machines, intelligent training sample selection algorithms, spectral mixture analysis, parallel processing.

## 1. INTRODUCTION

Hyperspectral imaging, is concerned with the measurement, analysis, and interpretation of spectra acquired from a given scene (or specific object) at a short, medium or long distance by an airborne or satellite sensor.<sup>1</sup> The concept of hyperspectral imaging originated at NASA's Jet Propulsion Laboratory in California with the development of the Airborne Visible Infra-Red Imaging Spectrometer (AVIRIS), able to cover the wavelength region from 0.4 to 2.5  $\mu\text{m}$  using more than two hundred spectral channels, at nominal spectral resolution of 10 nm.<sup>2</sup> As a result, each pixel vector collected by a hyperspectral instrument can be seen as a *spectral signature* or *fingerprint* of the underlying materials within the pixel.

The special characteristics of hyperspectral datasets pose different processing problems,<sup>3</sup> which must be necessarily tackled under specific mathematical formalisms, such as classification, segmentation, image coding or spectral mixture analysis.<sup>4</sup> These problems also require specific dedicated processing software and hardware platforms. In most studies, techniques are divided into full-pixel and mixed-pixel techniques, where each pixel vector defines a *spectral signature* or *fingerprint* that uniquely characterizes the underlying materials at each site in a scene.<sup>5</sup> Mostly based on previous efforts in multispectral imaging, full-pixel techniques assume that each pixel vector measures the response of one single underlying material. Often, however, this is not a realistic assumption. If the spatial resolution of the sensor is not fine enough to separate different pure signature classes at a macroscopic level, these can jointly occupy a single pixel, and the resulting spectral signature will be a composite of the individual pure spectra, called *endmembers* in hyperspectral terminology.<sup>6</sup> Mixed pixels can also result when distinct materials are combined into a homogeneous or intimate mixture, which occurs independently of the spatial resolution of the sensor. To address these issues, spectral unmixing approaches have been developed under the assumption that each pixel vector measures the response of multiple underlying materials.<sup>7</sup>

---

Send correspondence to Antonio J. Plaza:

E-mail: aplaza@unex.es; Telephone: +34 927 257195 (Ext. 571); URL: <http://www.umbc.edu/rssi/pl/people/aplaza>

Our main goal in this paper is to provide a seminal view on recent advances in techniques for full-pixel and mixed-pixel classification of hyperspectral images, taking into account both the spectral and spatial properties of the data. Due to the small number of training samples and the high number of features available in remote sensing applications, reliable estimation of statistical class parameters is a challenging goal.<sup>4</sup> As a result, with a limited training set, classification accuracy tends to decrease as the number of features increases. This is known as the Hughes effect. Furthermore, high-dimensional spaces are mostly empty, thus making density estimation more difficult. One possible approach to handle the problem of dimensionality is to consider the geometrical properties rather than the statistical properties of the classes. In this regard, it is important to develop techniques able to select the most highly informative training samples from the available training set.<sup>8</sup> The good classification performance already demonstrated by techniques such as kernel methods and support vector machines (SVMs) in remote sensing applications,<sup>9</sup> using spectral signatures as input features, is further increased in this work using intelligent training sample selection algorithms.

It should be noted that most available hyperspectral data processing techniques focused on analyzing the data without incorporating information on the spatially adjacent data, i.e., hyperspectral data are usually not treated as images, but as unordered listings of spectral measurements with no particular spatial arrangement. In certain applications, however, the incorporation of spatial and spectral information is mandatory to achieve sufficiently accurate mapping and/or classification results.<sup>10</sup> To address the need for developments able to exploit *a priori* information about the spatial arrangement of the objects in the scene in order to complement spectral information, this paper further explores additional techniques for spatial-spectral data processing. In particular, mathematical morphology<sup>11</sup> is a well-consolidated image processing technique that provides a remarkable framework to achieve the desired integration. In previous work, morphological processing has been used to extract information about the size, shape and the orientation of structures in single-band remote sensing images.<sup>12</sup> Here, we revisit and improve morphological approaches which deal with the full spectral information available in the data, including classification techniques based on extended morphological profiles and spectral unmixing techniques able to extract image endmembers using spatial and spectral information simultaneously, and to exploit the inherent convexity of mixed pixels in the unmixing stage.

While integrated spatial-spectral developments such as those addressed above hold great promise for hyperspectral data processing, they also introduce new computational challenges. With the recent explosion in the amount and complexity of hyperspectral data, parallel processing *hardware* has soon become a requirement in many remote sensing missions, especially with the advent of low-cost systems such as commodity clusters of computers.<sup>13</sup> In order to address this relevant issue, this paper also explores the development of parallel processing support for some of the most representative data classification algorithms discussed in the paper.

The remainder of the paper is organized as follows. Section 2 explores techniques for supervised classification of hyperspectral data using limited training samples. Section 3 focuses on the development of unsupervised spatial-spectral analysis techniques, including morphological techniques for full-pixel classification and spatial-spectral mixture analysis. Section 4 develops parallel versions of some of the algorithms considered in Section 2 and Section 3. Section 5 illustrates the performance of the proposed techniques (in different application scenarios) using real hyperspectral data sets, and also discusses the possibility of obtaining processing results quickly enough for practical use utilizing the parallel Thunderhead supercomputer at NASA's Goddard Space Flight Center as a baseline computing platform. Finally, section 6 concludes the paper with some remarks and hints at plausible future research lines.

## 2. SUPERVISED CLASSIFICATION USING LIMITED TRAINING SAMPLES

Supervised classification is one of the most commonly undertaken analyses of remotely sensed hyperspectral data. The output of a supervised classification is effectively a thematic map that provides a snapshot representation of the spatial distribution of a particular theme of interest such as land cover. Recent research has indicated the considerable potential of SVM-based approaches for the supervised classification of remotely sensed hyperspectral data.<sup>14</sup> Comparative studies have shown that classification by a SVM can be more accurate than techniques such as neural networks, decision trees and probabilistic classifiers such as maximum likelihood classification.<sup>9</sup> SVMs were designed for binary classification but various methods exist to extend the binary approach to multiclass classification, such as the *one versus the rest* and the *one versus one* strategies.<sup>16</sup>

In essence, the SVM classification is based on the notion of fitting an optimal separating hyperplane between classes by focusing on the training samples that lie at the edge of the class distributions, the support vectors. All of the other training samples are effectively discarded as they do not contribute to the estimation of hyperplane location. In this way not only is an optimal hyperplane fitted, in the sense that it is expected to have a large degree of generalizability, but also a high accuracy may be obtained with the use of a small training set. Recently, innovative kernel-based algorithms with enhanced properties have been developed, such as semi-supervised or *transductive* SVMs (TSVMs) learning procedures,<sup>15</sup> which are used to exploit both labeled and unlabeled pixels in the training stage, or contextual SVMs,<sup>14</sup> in which spatial and spectral information is incorporated by means of the use of proper kernel functions. The capability of semi-supervised SVMs to capture the intrinsic information present in the unlabeled data can further mitigate the Hughes phenomenon, and contextual SVMs can address the problems related to the non-stationary behavior of the spectral signatures of classes in the spatial domain.

In this section, we first recall the basic formulation of standard SVMs and then introduce two novel automatic algorithms for intelligent selection of training samples, which are aimed at reducing the number of training samples required to produce an accurate full-pixel classification result.

## 2.1 SVM formulation and use of different kernel functions

The SVM was first investigated as a binary classifier.<sup>16</sup> Given a training set  $S = \{(\Phi(\mathbf{x})_i, y_i) \mid i \in [1, n]\}$  projected into a Hilbert space  $\mathcal{H}$  by some mapping  $\Phi$ , the SVM separates the data by an optimal hyperplane  $H_p$  that maximizes the margin. Allowing some training errors,  $H_p$  is found by jointly maximizing the margin  $\|\mathbf{w}\|$  and minimizing the sum of errors  $\sum_{i=1}^n \xi_i$ . The convex optimization problem is solved by considering the dual optimization through the use of Lagrange multipliers:

$$\begin{aligned} \max_{\alpha} \quad & \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \langle \Phi(\mathbf{x})_i, \Phi(\mathbf{x})_j \rangle_{\mathcal{H}}, \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C \quad \forall i \in [1, n], \quad \sum_{i=1}^n \alpha_i y_i = 0. \end{aligned} \tag{1}$$

Through the use of a kernel function,  $k$ , it is possible to compute implicitly the inner product in  $\mathcal{H}$  in the original space:  $\langle \Phi(\mathbf{x})_i, \Phi(\mathbf{x})_j \rangle_{\mathcal{H}} = k(\mathbf{x}_i, \mathbf{x}_j)$ . SVM used with a kernel function is a non-linear classifier, where the non-linear ability is included in the kernel. The decision rule is finally  $y_u = \text{sgn}(\sum_{i=1}^n y_i \alpha_i k(\mathbf{x}_u, \mathbf{x}_i) + b)$ . The use of different kernel functions leads to different SVM configurations. The most used kernels are the polynomial kernel,  $k_{poly}(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + \theta)^d$  and the Gaussian kernel,  $k_{gauss}(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \|\mathbf{x} - \mathbf{z}\|^2)$ . When some *a-priori* are known, it is possible to include them into the kernel, to improve the classification. A spectral angle distance (SAD)-based kernel was recently introduced as follows:  $k_{SAD}(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \alpha(\mathbf{x}, \mathbf{z})^2)$ , where  $\alpha(\mathbf{x}, \mathbf{z}) = \arccos\left(\frac{\langle \mathbf{x}, \mathbf{z} \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{z}\|}\right)$ .

## 2.2 Intelligent training sample selection algorithms

This subsection explores the issue of how to select a set of training samples for SVM classifiers. Traditionally, training samples have been selected randomly from available labeled samples. However, it has been shown in previous work that not only the size but also the composition of the training set has a significant impact on the final classification results. In particular, a challenging aspect in the design of SVM classifiers for hyperspectral imagery is to reduce the need for large training sets. Previous studies on this topic have investigated a range of issues, including the use of feature selection and feature extraction methods to reduce the dimensionality of the input data, the use of unlabeled and semi-labeled training samples, the accommodation of spatial dependence in the data to define an efficient sampling design, or the use of statistics derived on other locations.<sup>8</sup> In this work, we hypothesize that effective supervised classifiers demands intelligent training sample selection algorithms able to seek for the most informative training samples, thus optimizing the compromise between estimation accuracy (to be maximized) and ground-truth knowledge (to be minimized). Since training samples can be either pure or mixed in nature, we develop two unsupervised algorithms which account for the spectral purity of training samples in the process of selecting those samples for training purposes. Training sets obtained using these algorithms will be compared to randomly selected training sets in SVM-based classification experiments conducted in Section 5.

### 2.2.1 Selection of pure training samples

In order to extract the purest pixel vectors in the data set as training samples, we use a modified version of the pixel purity index (PPI) algorithm available commercially from Research Systems ENVI.<sup>17</sup> This algorithm automatically searches for the purest spectral signatures which are assumed to be the vertices of a convex hull. The algorithm proceeds by generating a large number of random,  $N$ -dimensional unit vectors called ‘skewers’ through the dataset. 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. 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 final endmembers.

The inputs to the algorithm are a hyperspectral data cube  $\mathbf{f}$  with  $N$  spectral bands, the number of training samples to be extracted,  $t$ , and the number of random skewers to be generated during the process,  $k$ . The output of the algorithm is a set of  $t$  training samples  $\{\mathbf{t}_i\}_{i=1}^t$ . The algorithm can be summarized by the following steps:

1. *Skewer generation.* Produce a set of  $k$  randomly generated unit vectors  $\{\mathbf{skewer}_j\}_{j=1}^k$ .
2. *Extreme projections.* For each  $\mathbf{skewer}_j$ , all sample pixel vectors  $\mathbf{f}(x,y)$  in the original data set, where  $(x,y)$  denotes the spatial coordinates, are projected onto  $\mathbf{skewer}_j$  via dot products of  $|\mathbf{f}(x,y) \cdot \mathbf{skewer}_j|$  to find sample vectors at its extreme (maximum and minimum) projections, thus forming an extrema set for  $\mathbf{skewer}_j$  which is denoted by  $S_{extrema}(\mathbf{skewer}_j)$ . Despite the fact that a different  $\mathbf{skewer}_j$  would generate a different extrema set  $S_{extrema}(\mathbf{skewer}_j)$ , it is very likely that some sample vectors may appear in more than one extrema set. In order to deal with this situation, we define an indicator function of a set  $S$ , denoted by  $I_S(\mathbf{x})$ , to denote membership of a vector element  $\mathbf{x}$  to that particular set as follows:

$$I_S(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in S \\ 0 & \text{if } \mathbf{x} \notin S \end{cases} \quad (2)$$

3. *Calculation of PPI scores.* Using the indicator function above, we calculate the PPI score associated to each sample pixel vector  $\mathbf{f}(x,y)$  in the input scene (i.e., the number of times that given pixel has been selected as extreme in step (2) using the following equation:

$$\text{PPI}(x,y) = \sum_{j=1}^k I_{S_{extrema}(\mathbf{skewer}_j)}(\mathbf{f}(x,y)) \quad (3)$$

4. *Selection of training samples.* Find the  $t$  pixel vectors with the highest scores of  $\text{PPI}(x,y)$  and use them to form a training set  $\{\mathbf{t}_i\}_{i=1}^t$ .

### 2.2.2 Selection of border training samples

As an alternative to the algorithm developed in the previous subsection, we describe below an automatic algorithm that iteratively seeks for border training samples. The separation of a training set into border and non-border patterns was first explored by Foody,<sup>8</sup> who expressed *borderness* as the difference between the two smallest distances measured for each training pattern. Here, membership was indicated by the Mahalanobis distance, which provides a measure of the typicality of a pattern to a certain class. A border-training pattern is expected to be almost as close to its actual class of membership as it is to any other class. Therefore, the difference in the Mahalanobis distances between the two most likely classes of membership would be small for a border pattern. This focus on the vicinity of the hyperplanes that can optimally separate the classes is similar to aspects of Lee and Landgrebe’s decision boundary feature extraction.<sup>18</sup>

Here, we develop an automatic algorithm inspired by the concept proposed by Foody, but further adapted to a mixed pixel interpretation scenario. The concept implemented by this algorithm can actually be viewed as the opposite to that used by convex geometry-based endmember extraction methods such as PPI or Winter’s N-FINDR algorithm.<sup>19</sup> Since the PPI does not produce a set of spectrally distinct endmembers but a set of pure pixels which are actually endmember candidates that may contain several instances of the same endmember, we

use the N-FINDR for initialization of our algorithm (allowing N-FINDR estimate how many endmembers are in the input hyperspectral data). The border training sample selection algorithm can be summarized by the following steps:

1. Label the set of spectral endmembers  $\{\mathbf{e}_i\}_{i=1}^p$  produced by the N-FINDR algorithm as class *core* patterns.
2. Apply a spectral screening algorithm to identify the sample pixel vectors within a small spectral angle  $\theta$  from any of the  $p$  core classes above, denoted by  $\{\mathbf{r}_j\}_{j=1}^q$  with  $q \geq p$ .
3. Associate each signature of the set  $\{\mathbf{r}_j\}_{j=1}^q$  to one of the available pure classes  $\{\mathbf{e}_i\}_{i=1}^p$  by computing  $\mathbf{r}_j^{(i)} = \operatorname{argmin}_i \{\operatorname{SAD}(\mathbf{r}_j, \mathbf{e}_i)\}$  for all  $j = 1, 2, \dots, q$ , where the notation of  $\mathbf{r}_j^{(i)}$  indicates that the SAD between  $\mathbf{r}_j$  and  $\mathbf{e}_i$  is the minimum, i.e.,  $\mathbf{e}_i$  is the most spectrally similar endmember to  $\mathbf{r}_j$ .
4. Let  $\mathbf{r}_{j,k}^{(i)} \subseteq \{\mathbf{r}_j\}_{j=1}^q$  be the  $k$ -th sample associated with class  $\mathbf{e}_i$ , and let  $|\mathbf{r}_{j,k}^{(i)}|$  be the cardinality of the set  $\{\mathbf{r}_{j,k}^{(i)}\}$ , composed of all the samples in  $\{\mathbf{r}_j\}_{j=1}^q$  associated with  $\mathbf{e}_i$ .
5. For each sample pixel vector  $\mathbf{f}(x, y)$ , compute the Mahalanobis distance from each pure class  $\mathbf{e}_i$  as follows:  $\operatorname{MD}(\mathbf{f}(x, y), \mathbf{e}_i) = (\mathbf{f}(x, y) - \boldsymbol{\mu}_i)^T \mathbf{K}_i^{-1} (\mathbf{f}(x, y) - \boldsymbol{\mu}_i)$ , where  $\mathbf{K}_i$  is the sample covariance matrix of the class given by  $\mathbf{e}_i$ , and  $\boldsymbol{\mu}_i$  is the mean for that class, given by  $\boldsymbol{\mu}_i = \frac{1}{|\mathbf{r}_{j,k}^{(i)}|} \sum_{k=1}^{|\mathbf{r}_{j,k}^{(i)}|} \mathbf{r}_{j,k}^{(i)}$ .
6. Compute a borderiness score  $\operatorname{BS}(x, y)$  for each pixel vector  $\mathbf{f}(x, y)$  by finding the two most likely classes of membership for the pixel, say,  $\mathbf{e}_i$  and  $\mathbf{e}_j$  with  $1 \leq i \leq p$  and  $1 \leq j \leq p$ , and then calculating:
$$\operatorname{BS}(x, y) = |\operatorname{MD}(\mathbf{f}(x, y), \mathbf{e}_i) - \operatorname{MD}(\mathbf{f}(x, y), \mathbf{e}_j)|. \quad (4)$$
7. Find the  $t$  pixel vectors with the highest scores of  $\operatorname{BS}(x, y)$  and use them to form a training set  $\{\mathbf{t}_i\}_{i=1}^t$ .

### 3. UNSUPERVISED SPATIAL-SPECTRAL UNMIXING

While our focus in the previous subsection has been on full-pixel classification approaches, spectral unmixing has been an alluring exploitation goal since the earliest days of imaging spectroscopy. No matter the available spatial resolution, in natural environments, the spectral signature for a nominal pixel is invariably a mixture of the signatures of the various materials found within the spatial extent of the ground instantaneous field view of the sensor. In hyperspectral imaging, the actual unmixing to determine apparent pixel fractions is a simple and straightforward numerical process. However, it is the derivation and validation of the correct suite of endmembers that has remained a challenging and elusive goal for the past twenty years. Several approaches have been developed for deriving spectral endmembers in the literature, including the PPI,<sup>17</sup> the N-FINDR,<sup>19</sup> or the iterative error analysis (IEA) algorithm.<sup>20</sup> Although these methods have been used successfully in many applications, they are exclusively based on the spectral information of the data. As a result, further work remains to be done in the area of exploiting the joint spatial and spectral nature of the data in spectral unmixing applications. In this section, we develop spatial-spectral techniques for unsupervised spectral unmixing using mathematical morphology concepts. The section is structured as follows. First, we describe how basic morphological operations have been extended to hyperspectral image processing. Then, we develop an unsupervised algorithm for endmember extraction and morphological spectral unmixing which uses both the spatial and the spectral information in complementary fashion.

#### 3.1 Extended mathematical morphology

Mathematical morphology is a theory for spatial structure analysis that was established by introducing fundamental operators applied to two sets. A set is processed by another one having a carefully selected shape and size, known as the structuring element (SE).<sup>11</sup> The two basic operations of mathematical morphology are erosion and dilation. These operators can be described (in the context of greyscale morphology) by viewing the image data as an imaginary topographic relief in which the brighter the gray tone, the higher the corresponding elevation. With the above assumption in mind, morphological operations can be interpreted as the result of

sliding a SE over the topographical relief, so that the SE defines the new (dilated or eroded) scene based on its spatial properties such as height or width.

Extension of morphological operators to multichannel data such as hyperspectral imagery with hundreds of spectral channels is not straightforward. A simple approach consists in applying grayscale morphology techniques to each channel separately, an approach that has been called *marginal morphology* in the literature. However, the marginal approach is often unacceptable in remote sensing applications because, when morphological techniques are applied independently to each image channel, analysis techniques are subject to the well-known problem of *false colors*; that is, it is very likely that new spectral constituents (not present in the original hyperspectral image) may be created as a result of processing the channels separately. An alternative (and perhaps more appropriate) way to approach the problem of multichannel morphology is to treat the data at each pixel as a vector. Unfortunately, there is no unambiguous means of defining the minimum and maximum values between two vectors of more than one dimension, and thus it is important to define an appropriate arrangement of vectors in the selected vector space. Here, we develop an application-driven vector ordering technique<sup>21</sup> based on a spectral purity-based criterion, where each pixel vector is ordered according to its spectral distance to other neighboring pixel vectors in the  $N$ -dimensional data set  $\mathbf{f}$ . More specifically, we adopt a distance-based technique which utilizes a cumulative distance between one particular pixel vector  $\mathbf{f}(x, y)$ , where  $(x, y)$  indicates the spatial coordinates, and all the pixel vectors in the spatial neighborhood given by a SE denoted by  $K$  as follows:

$$C_K(\mathbf{f}(x, y)) = \sum_{(s,t) \in K} \text{SAD}(\mathbf{f}(x, y), \mathbf{f}(s, t)), \quad (5)$$

where SAD is the spectral angle distance.<sup>3</sup> As a result,  $C_K(\mathbf{f}(x, y))$  is given by the sum of SAD scores between  $\mathbf{f}(x, y)$  and every other pixel vector in the  $K$ -neighborhood. At this point, we need to be able to define a maximum and a minimum given an arbitrary set of vectors  $\mathbf{S} = \{\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_p\}$ , where  $k$  is the number of vectors in the set. This can be done by computing  $C_K(\mathbf{S}) = \{C_K(\mathbf{v}_1), C_K(\mathbf{v}_2), \dots, C_K(\mathbf{v}_k)\}$  and selecting  $\mathbf{v}_i$  such that  $C_K(\mathbf{v}_i)$  is the minimum of  $C_K(\mathbf{S})$ , with  $1 \leq i \leq k$ . In similar fashion, we can select  $\mathbf{v}_j$  such that  $C_K(\mathbf{v}_j)$  is the maximum of  $C_K(\mathbf{S})$ , with  $1 \leq j \leq p$ . Based on the definitions above, the extended erosion  $\mathbf{f} \ominus K$  consists of selecting the  $K$ -neighborhood pixel vector that produces the minimum  $C_K$  value as follows:<sup>21</sup>

$$(\mathbf{f} \ominus K)(x, y) = \operatorname{argmin}_{(s,t) \in K} \{C_K(\mathbf{f}(x + s, y + t))\}. \quad (6)$$

On the other hand, the extended dilation  $\mathbf{f} \oplus K$  selects the  $K$ -neighborhood pixel that produces the maximum value for  $C_K$  as follows:<sup>21</sup>

$$(\mathbf{f} \oplus K)(x, y) = \operatorname{argmax}_{(s,t) \in K} \{C_K(\mathbf{f}(x - s, y - t))\}. \quad (7)$$

In order to avoid changing the size and shape of the features in the image, a desirable feature for spatial-spectral filtering, extended morphological opening and closing operations have also been defined, respectively, as follows:  $(\mathbf{f} \circ K)(x, y) = [(\mathbf{f} \ominus K) \oplus K](x, y)$ , i.e., erosion followed by dilation, and  $(\mathbf{f} \bullet K)(x, y) = [(\mathbf{f} \oplus K) \ominus K](x, y)$ , i.e., dilation followed by erosion.

### 3.2 Spatial-spectral endmember extraction and pixel unmixing

In the following, we develop a novel approach for spatial-spectral endmember extraction and spectral unmixing. The algorithm makes use of extended morphological operations to integrate the spatial and the spectral information. It should be noted that this spectral morphology approach is different to other morphological techniques used in the past for classification purposes, since here we wish to take into account the full (multi-band) spectral information *as a whole* (without any previous feature extraction) in order to account for subtle spectral differences in the endmember searching and spectral unmixing processes. A step-by-step description of the proposed endmember extraction algorithm is given below. The algorithm, based on the previously developed automated morphological endmember extraction (AMEE) algorithm,<sup>6</sup> is called I-AMEE to account for the new iterative nature of the proposed algorithm, which allows propagation of pure pixels between subsequent iterations, as opposed to the previous version of the algorithm. The algorithm described below also incorporates a spatial-spectral unmixing stage which was not present in the previous version. The inputs to I-AMEE are the full

hyperspectral data cube  $\mathbf{f}$ , a structuring element  $K$  (used to define the spatial context around each image pixel), a maximum number of algorithm iterations  $I_{max}$ , and a maximum number of endmembers to be extracted,  $p$ . The output is an endmember set,  $\{\mathbf{e}_i\}_{i=1}^q$ , with  $q \leq p$ . A step-by-step description of the algorithm follows:

1. Set  $i = 1$  and initialize a morphological eccentricity index  $MEI(x, y) = 0$  for each pixel  $\mathbf{f}(x, y)$ .
2. Move  $K$  through all the pixels of the input data, defining a local spatial search area around each pixel  $\mathbf{f}(x, y)$ , and calculate the maximum and minimum pixel vectors at each  $K$ -neighborhood using extended opening and closing operations, respectively, and update the MEI at each spatial location  $(x, y)$  using:

$$MEI(x, y) = SAD[(\mathbf{f} \circ K)(x, y), (\mathbf{f} \bullet K)(x, y)]. \quad (8)$$

3. Set  $i = i + 1$ . If  $i = I_{max}$ , then go to step 4. Otherwise, replace the original image with its dilation using  $K$  as follows:  $\mathbf{f} = \mathbf{f} \oplus K$ . This represents an optimization of the algorithm that propagates only the purest pixels at the local neighborhood to the following algorithm iteration. Then, go to step 2.
4. Select the set of  $p$  pixel vectors with higher associated MEI scores (called endmember candidates) and form a unique spectral set of  $\{\mathbf{e}_i\}_{i=1}^q$  pixels, with  $q \leq p$ , by calculating the SAD for all pixel vector pairs.
5. Finally, a spatial-spectral unmixing procedure is accomplished by considering a spatial neighborhood (defined by  $K$ ) around each mixed pixel. This method is similar to traditional approaches, in the sense that it makes use of the standard (fully constrained or unconstrained) least squares-based technique to estimate abundance fractions.<sup>3</sup> But it differs from traditional methods in the fact that the endmember set used for each pixel is adaptively calculated based on the spatial context. The method is based on the following sequence of steps:
  - (a) Before unmixing a certain pixel, say  $\mathbf{f}(x, y)$ , a weight is assigned to all the pixels in the  $K$ -neighborhood centered at spatial coordinates  $(x, y)$ . This is done by first calculating, for each pixel vector in the  $K$ -neighborhood, the SAD to each one of the endmembers in the set  $\{\mathbf{e}_i\}_{i=1}^q$ , and labeling each pixel as an instance of a certain endmember (candidate) by using the minimum SAD score.
  - (b) Then, a weight is assigned to each endmember candidate (the weight is inversely proportional to the minimum SAD score reported for that candidate, which indicates higher degree of similarity to its corresponding endmember).
  - (c) Finally, all endmember candidates in the  $K$ -neighborhood are sorted by weight, and only those with associated weights above a certain tolerance threshold are incorporated to the ‘local’ endmember set which is finally used to unmix the pixel  $\mathbf{f}(x, y)$ . In other words, the abundance estimation is still performed by using a (fully constrained or unconstrained) least squares technique, but the actual composition of the endmember set used to perform the least squares estimation is adaptively selected (for each particular pixel vector) depending on the spatial-spectral context around the pixel, as opposed to traditional unmixing approaches, in which the entire set of spectral endmembers is always used to unmix every pixel in the input scene.

#### 4. PARALLEL IMPLEMENTATIONS

This section develops parallel processing frameworks for implementation of some of the hyperspectral processing algorithms described in Sections 2 and 3. Our design of parallel algorithms was driven by the desire to maximize speedup while also considering productivity, with the ultimate goal of reusing most of the code for the sequential algorithms in the parallel implementations. For addressing speedup, or more specifically, to balance the load across the available processors, we used a standard master-slave decomposition framework.<sup>13</sup> In this approach, a particular processor of the underlying parallel computing platform is designated as the *master*, whose job is to decompose the problem and build a list of tasks that will be assigned to the *workers*. The master processor then waits to receive back the completed tasks from each worker processor, gathers the partial results from them and provides the final result. Under the above framework, the best situation arises when a problem can be decomposed into a non-trivial number of independent sub-problems. In the jargon of parallel computation, this is called an *embarrassingly parallel* problem.

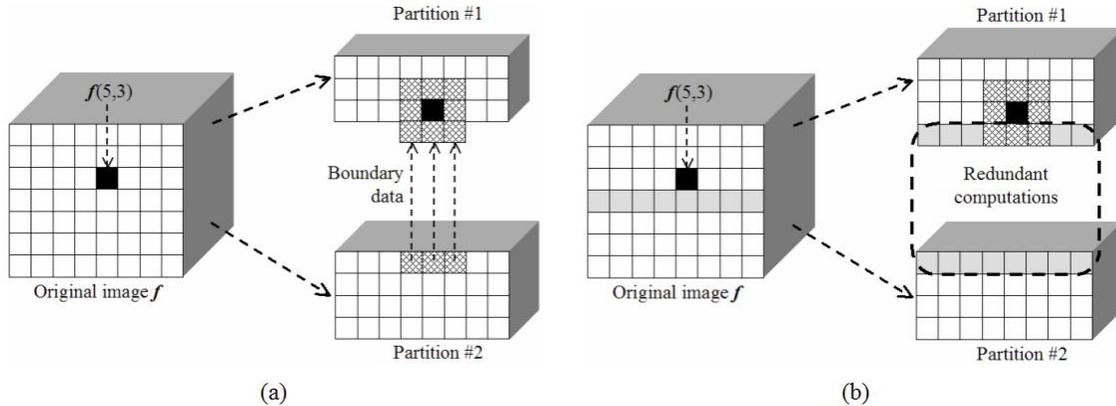


Figure 1. Parallel morphological processing of hyperspectral images: (a)  $3 \times 3$  SE computation split among two processing elements. (b) Introduction of an overlap border (redundant computations) to minimize inter-processor communication in a  $3 \times 3$  SE computation.

#### 4.1 Parallel implementation of supervised SVM classification

As stated in Section 2, a standard approach in SVM-based classification using a certain kernel is to decompose the multiple class problem into multiple two-class problems.<sup>9</sup> In order to develop our parallel version, we use *pairwise classification* to create  $\frac{S(S-1)}{2}$  separate pairs of classifiers. Then, we use a voting strategy which is based on building  $S$  pairs of classifiers, each of which with the pair of classes consisting of class  $i$ , and the combined class of  $1, 2, \dots, i-1, i+1, \dots, S$ . A simple *naive* approach to parallelization is to run groups of  $P$  pairs of problems distributed across the set of available  $P$  processors. However, typically the number of training vectors for each pair is not the same, resulting in the fact that the processor handling the largest number training vectors will always be the last to finish, leaving other processors idle. To address this issue, the master processor first builds a list of tasks, given by the set of all  $\frac{S(S-1)}{2}$  pairs of classes, and then assigns a particular processor to each of the first  $P-1$  tasks. When a worker processor completes its calculations, the master sends that worker the next task on the list. Using this simple approach, the worker processors are always busy.

#### 4.2 Parallel implementation of unsupervised morphological classification

Morphological operations rely on a SE-based window-moving strategy that introduces additional considerations for parallelization. In previous work, the term parallelizable spatial/spectral partition (PSSP) was defined as a partition of the input data in the spatial domain that can be processed in parallel without the need for additional inter-processor communications.<sup>13</sup> Therefore, a PSSP may be seen as a chunk of data that can be processed independently, using smaller SEs. The generalized description of a PSSP given above allows us to maximize code reusability since each PSSP can be analyzed independently at each processing unit. At the same time, communication overhead is minimized. In order to avoid inter-processor communication when the structuring element computation is split among processing nodes, as shown in Fig. 1(a), a spatial-domain partitioning module has been implemented in the master so that the partitioning of the data into PSSPs introduces an overlap border which avoids accesses outside the local domain of each partition. In this case, the computations for the pixel vector at spatial coordinates (5,3) in the original image, denoted by  $f(5,3)$ , will need to originate from two processing elements since this pixel becomes a border pixel after spatial-domain partitioning. As a result, a communication overhead involving three  $N$ -dimensional pixel vectors (located in partition #2) is required in order to complete the SE-based computation for the pixel vector  $f(5,3)$  in partition #1 (the same comment applies to all pixels in the third row of partition #1). However, if an overlap border is carefully added to partition #1 (consisting of the entire first row of pixels allocated to partition #2), as illustrated in Fig. 1(b), then boundary data no longer need to be exchanged between neighboring processors in order to complete all calculations at partition #1, and therefore this data partition can be processed locally without the need for inter-processor communications. With the above ideas in mind, we provide below a step-by-step description of our parallel version of the I-AMEE algorithm:

1. The master processor partitions the data into  $P$  spatial-domain partitions (with their overlap borders to avoid inter-processor communications), denoted by  $\{\text{PSSP}_i\}_{i=1}^P$ , and distributes the partitions among all the processors in the system.
2. Using parameters  $I_{max}$  (maximum number of iterations) and  $p$  (maximum number of endmembers to be extracted), each worker executes (in parallel) steps (1)–(4) of the sequential I-AMEE algorithm for its corresponding  $\text{PSSP}_i$ , thus obtaining a MEI score for each pixel in the local partition and obtaining a local set of *unique* endmembers.
3. The master gathers all the local endmember sets provided by the workers and forms a *global* set of endmembers  $\{\mathbf{e}_i\}_{i=1}^q$ , with  $q \leq p$ , by calculating the SAD for all possible endmember pairs in parallel.
4. Using the sets  $\{\text{PSSP}_i\}_{i=1}^P$  and  $\{\mathbf{e}_i\}_{i=1}^q$ , perform step (5) of the sequential I-AMEE algorithm (spatial-spectral unmixing) locally at each worker. This is an *embarrassingly parallel* computation, since the calculations can be performed independently at each worker. In the end, the master gathers the information provided by the workers and forms a final output.

To conclude this section, we emphasize that all the training sample selection algorithms described in subsection 2.2 have been implemented in parallel using the concept of PSSP described above, resulting in embarrassingly parallel implementations in most cases due to the regularity in the computations involved in those algorithms. Performance data for all the sequential and parallel algorithms described in this paper, in the context of two realistic hyperspectral image analysis applications, are given in the following section.

## 5. EXPERIMENTAL RESULTS

This section provides an assessment of the effectiveness of the proposed classification algorithms in the analysis of real hyperspectral data sets. Two different application domains: 1) land-cover classification of agricultural fields; and 2) sub-pixel mapping of geological features, are used to provide a detailed algorithm cross-validation. The section is organized as follows. First, we provide an overview of the hyperspectral image data sets used in this study. A detailed quantitative analysis of the proposed classification algorithms in the context of two highly representative application domains follows. The section concludes by reporting performance results for parallel versions of selected algorithms.

### 5.1 Hyperspectral image data sets

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

#### 5.1.1 AVIRIS hyperspectral data over Indiana's Indian Pines region

This 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  $1939 \times 677$  pixels and 204 spectral bands in the wavelength range 0.4–2.5  $\mu\text{m}$  (523 MB in size). 20 AVIRIS bands were removed from the original scene prior to analysis due to low SNR in those bands. The AVIRIS Indian Pines data set represents a very challenging classification problem dominated by similar spectral classes and mixed pixels. Specifically, the primary crops of the area, mainly corn and soybeans, were very early in their growth cycle with only about 5% canopy cover. This fact makes most of the scene pixels highly mixed in nature. Discriminating among the major crops under this circumstances can be very difficult, a fact that has made this scene an extensively used benchmark to validate classification accuracy of hyperspectral imaging algorithms. For illustrative purposes, Fig. 2(a) shows the spectral band at 587 nm of the original scene and Fig. 2(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>).

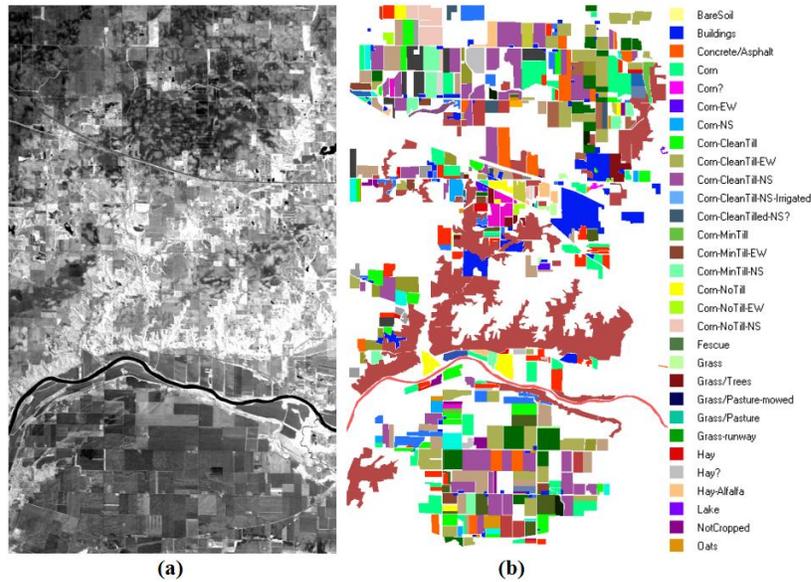


Figure 2. (a) Spectral band at 587 nm wavelength of an AVIRIS scene comprising agricultural and forest features at Indian Pines region. (b) Ground-truth map with 30 mutually-exclusive land-cover classes.

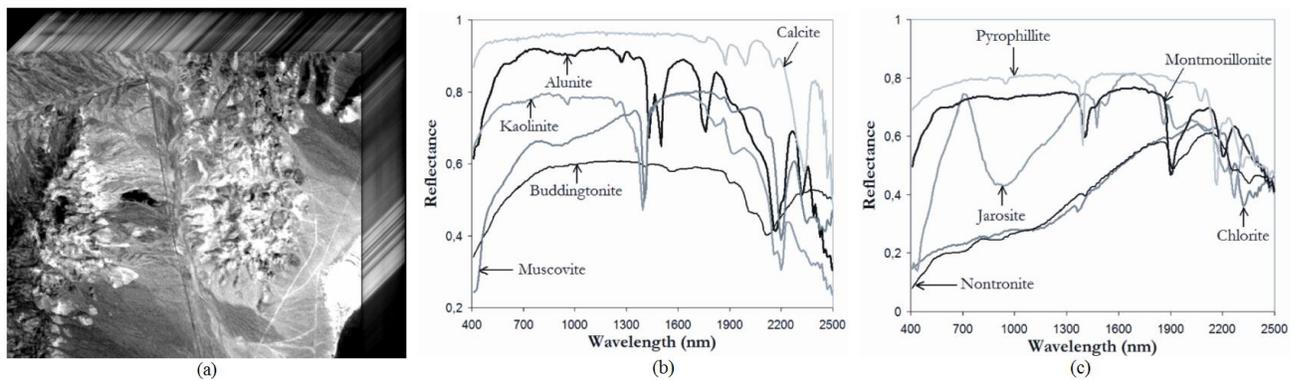


Figure 3. (a) AVIRIS scene over Cuprite mining district. (b-c) Ground-truth mineral spectra provided by USGS.

### 5.1.2 AVIRIS hyperspectral data over Cuprite mining district, Nevada

Another AVIRIS scene collected over the Cuprite mining district in Nevada was also used in experiments to evaluate the proposed algorithms in the context of a mineral mapping application. The data set (available from <http://aviris.jpl.nasa.gov/html/aviris.freedata.html>) consists of  $614 \times 512$  pixels and 204 bands in the wavelength range  $0.4\text{--}2.5 \mu\text{m}$  (125 MB in size). 20 AVIRIS bands were removed from the original scene prior to analysis due to low SNR in those bands. As opposed to the previously considered data sets, the Cuprite data are atmospherically corrected and available in reflectance units (not in at-sensor radiance), thus allowing direct comparison of pixel vectors to ground spectral signatures. The Cuprite site has been extensively mapped by the U.S. Geological Survey (USGS) in the last twenty years, and there is extensive ground-truth information available, including a library of mineral signatures collected on the field (see <http://speclab.cr.usgs.gov/spectral-lib.html>). Fig. 3(a) shows the AVIRIS Cuprite data cube using the spectral band at 587 nm wavelength of the scene. The spectra of USGS ground minerals: alunite, buddingtonite, calcite, kaolinite, muscovite [Fig. 3(b)], chlorite, jarosite, montmorillonite, nontronite, pyrophyllite [Fig. 3(c)] are also displayed. These selected spectral signatures will be used in this work to evaluate endmember extraction accuracy.

Table 1. Overall classification accuracies (in percentage) after applying polynomial, Gaussian and SAD-based kernels to the AVIRIS Indian Pines data set, using different training sample selection algorithms.

Training set size (%)		1%	2%	4%	6%	8%	10%	20%
$k_{poly}$	Random training samples	82.33	82.94	83.21	83.82	85.34	86.12	86.52
	Pure training samples	81.23	82.06	82.80	83.00	84.03	84.45	85.57
	Border training samples	83.44	84.23	84.45	84.96	86.27	87.44	89.96
$k_{gauss}$	Random training samples	87.94	88.23	88.78	88.96	89.45	89.48	90.77
	Pure training samples	86.53	87.02	87.64	87.93	88.12	88.26	88.55
	Border training samples	89.45	90.25	91.24	92.08	92.93	93.04	93.67
$k_{SAD}$	Random training samples	85.90	86.22	86.49	87.03	87.56	88.09	88.72
	Pure training samples	85.12	85.67	86.08	86.45	86.97	87.13	87.81
	Border training samples	86.05	86.93	87.57	88.12	89.30	90.12	90.57

## 5.2 Experimental assessment of algorithms

### 5.2.1 Experiment 1: Land-cover classification in agriculture

In this subsection, we discuss the performance of the proposed SVM-based classification algorithms in the context of an agricultural mapping application. For that purpose, we use the AVIRIS Indian Pines data set in Fig. 2. In order to validate the classification accuracy in this case study, we resort to ground-truth measurements collected by Purdue University over the Indian Pines region [see Fig. 2(b)]. Three types of kernels were used in experiments: polynomial ( $k_{poly}$ ), Gaussian ( $k_{gauss}$ ), and SAD-based ( $k_{SAD}$ ). Small training sets, composed of 1%, 2%, 4%, 6%, 8%, 10% and 20% of the ground-truth pixels available per class, were extracted using the two training sample selection algorithms described in subsection 2.2, and also using a random selection procedure. The SVMs were trained with each of these training subsets and then evaluated with the remaining test set. Each experiment was repeated five times, and the mean accuracy values were reported. Table 1 summarizes the overall classification results obtained using the three considered kernels and training sample selection algorithms.

From Table 1, it can be seen that SVMs generalize quite well: with only 1% of training pixels per class, almost 90% overall classification accuracy is reached by all kernels when trained using border training samples. In all cases, classification accuracies decreased when random and pure samples were used for the training sate. This confirms the fact that kernel-based methods in general and SVMs in particular are less affected by the Hughes phenomenon. It is also clear from Table 1 that the classification accuracy is generally correlated with the training set size. However, when border training samples were used, higher classification accuracies were achieved with less training samples. The above results indicate the importance of including mixed pixels at the border of class boundaries in the training set, as these border patterns are most efficient to determine the hyperplane between two classes. Finally, it can be seen in Table 1 that the best classification scores were generally achieved for the  $k_{gauss}$  kernel, in which the overall accuracy obtained with 1% of the training pixels per class is only 4.22% lower than the overall accuracy obtained with 20% of the training pixels per class (extracted using border training sample selection). On the other hand, the  $k_{SAD}$  kernel gives slightly degraded classification results. However, with accuracies above 85% in a challenging classification problem, this kernel also provides promising results. Finally, the  $k_{poly}$  kernel needs more training samples than the two other kernels to perform appropriately, as can be seen from the relatively poor results obtained by this kernel for a very limited number of training samples.

### 5.2.2 Experiment 2: Mapping of geological features

In this second experiment, we have conducted a cross-validation of spectral unmixing algorithms in the context of a mineral mapping application, using the well-known AVIRIS Cuprite data set for demonstration purposes. It should be noted that ground-truth information for this scene is only available in the form of a collection of USGS mineral signatures, and therefore the proposed I-AMEE algorithm is evaluated here in terms of its ability to extract a suite of spectral endmembers which are similar, spectrally, to USGS reference signatures. For comparative purposes, we also used the PPI, N-FINDR and IEA endmember extraction algorithms in our cross-validation. In addition, four linear spectral unmixing algorithms were used to derive fractional abundances. These include the standard fully constrained (abundance sum-to-one and non-negativity) and unconstrained least-squares based methods, and also two novel methods: spatial-spectral fully constrained and spatial-spectral unconstrained unmixing, presented in this work as the last step of the I-AMEE algorithm (but which can also use endmembers derived by other methods as input).

Table 2. SAD spectral similarity scores among selected USGS mineral spectra and the endmembers produced by different algorithms. The number of iterations executed by the I-AMEE algorithm are shown in the parentheses.

	PPI	N-FINDR	IEA	I-AMEE(1)	I-AMEE(3)	I-AMEE(5)
Alunite	0.084	0.081	0.084	0.084	0.076	0.063
Buddingtonite	0.106	0.084	0.112	0.094	0.091	0.084
Calcite	0.105	0.105	0.093	0.110	0.095	0.090
Chlorite	0.125	0.136	0.096	0.096	0.088	0.088
Kaolinite	0.136	0.152	0.134	0.134	0.134	0.134
Jarosite	0.112	0.102	0.112	0.108	0.102	0.089
Montmorillonite	0.106	0.089	0.120	0.096	0.089	0.082
Muscovite	0.108	0.094	0.105	0.106	0.091	0.077
Nontronite	0.102	0.099	0.099	0.099	0.078	0.078
Pyrophyllite	0.094	0.090	0.112	0.090	0.084	0.080

Prior to a full examination and discussion of results, it is important to outline parameter values used for the considered endmember extraction algorithms, bearing in mind that all of them require that the number of endmembers to be extracted,  $p$ , be set as an input parameter. In this work, we estimate the number of endmembers in the AVIRIS Cuprite data using the virtual dimensionality (VD) concept, which has been shown in previous work to be an effective approach for this purpose.<sup>3</sup> The VD estimated a total of  $p = 16$  endmembers, which is consistent with the available ground-truth for the scene. We remark that the N-FINDR algorithm includes an optional choice of setting a maximum number of endmembers to be identified, or allowing the algorithm to decide how many endmembers are present in the scene. In our experiments with this software, the number of endmembers to be extracted was left to the discretion of the algorithm. Since the algorithm also produced  $p = 16$  endmembers without using prior information, we believe that our dimensionality choice was indeed appropriate for this scene. As for the PPI algorithm, we used the version available in Research Systems ENVI 4.0 software, which is based on a semi-automatic procedure that first generates a large number of random vectors with unit norm ( $k = 10^4$  in experiments) and then projects the pixel vectors in the data to those random vectors, using ENVI's  $n$ -dimensional visualization tool to manually select a final endmember set. For the I-AMEE algorithm, we used a fixed SE with size of  $3 \times 3$  pixels and tested several values of the parameter  $I_{max}$ . Although in practice it is very difficult to fully optimize every method, we have used our previous experience with these methods to select parameters which are reasonably close to optimal for the test data. Also, the parameter values above are in agreement with those used before in previous studies.<sup>6</sup>

Table 2 tabulates the SAD scores obtained after comparing some selected USGS library spectra with the corresponding endmembers extracted by the four considered algorithms. The smaller the SAD values across the ten minerals considered, the better the results. As shown in the table the SAD spectral similarity scores obtained for the I-AMEE improved significantly as the number of iterations (and therefore the spatial context around each pixel) was increased in size. This demonstrated the importance of considering not only spectral but also spatial information in the selection of image endmembers.

To conclude this subsection, it is important to note that the linear mixture model is generally not flexible enough to accommodate the full range of spectral variability throughout real-world landscapes. In order to accurately characterize the data, it might be necessary to account for multiple scattering effects, which overestimate the results obtained from a linear scattering model. In order to calibrate the performance of linear spectral unmixing adopted in this work for estimating sub-pixel fractional abundances in the AVIRIS Cuprite data, we also unmix the data using the fully constrained and unconstrained versions of standard spectral unmixing<sup>3</sup> and the proposed spatial-spectral unmixing algorithm. Interestingly, our experiments revealed that the correspondent endmember fractional abundance maps derived by using unconstrained and fully constrained spatial-spectral unmixing were in visual agreement for all endmember extraction methods tested. In addition, negative and/or unrealistic abundance fractions (which usually indicate a bad fit of the model and reveal inappropriate endmember selections) were very rarely found by the unconstrained spatial-spectral unmixing, in particular, when the I-AMEE algorithm was used for the endmember selection stage. Quite opposite, the abundance maps derived by unconstrained linear spectral unmixing showed a much more significant fraction of negative abundances, in particular, when endmembers were derived using the spectral information only. Having those circumstances in mind, the results obtained in this section indicate that the linear mixture model, improved by the integration of spatial and spectral information, was able to provide a relatively good characterization of complex mixtures in the Cuprite mining district. However, further experimentation with nonlinear mixing models is highly desirable.

Table 3. Processing time (in seconds) on the Thunderhead cluster for the parallel SVM classifier using the AVIRIS Indian Pines scene. The times reported include those for training (finding the support vectors) using border training samples, prediction (applying the support vectors from every pair to get a pair prediction vote), classification (totaling the votes), and the total processing time. The speedups with regards to sequential execution of the algorithm in one processor are also reported.

Number of processors	1	4	16	64	256	512
Time border training	10023	4840	1076	309	332	379
Time prediction	12123	5680	1263	362	389	444
Time classification	53	50	47	49	52	54
Time total	22199	10570	2386	720	773	877
Speedup	1.0	2.1	9.3	30.8	28.7	25.3

Table 4. Processing times (in seconds) on the Thunderhead cluster for the parallel version of the I-AMEE algorithm using the AVIRIS Cuprite scene. The times reported include those for endmember extraction, spatial-spectral unmixing, and the total processing time. The speedups with regards to sequential execution in one processor are also reported.

Number of processors	1	4	16	64	256	512
Time endmember extraction	6625	2859	650	156	42	22
Time unmixing	2840	1226	279	64	18	11
Time total	9465	4085	929	220	60	33
Speedup	1.0	2.3	10.1	43.1	157.7	356.34

### 5.2.3 Experiment 3: Parallelization of hyperspectral data classification algorithms

In order to illustrate the performance of parallel SVM classifiers, we report results for the approach which resulted in better classification accuracies in Table 1, i.e., the SVM with Gaussian kernel, trained using border training samples. Table 3 reports parallel performance results of the proposed parallel SVM approach on the Thunderhead Beowulf cluster at NASA's Goddard Space Flight Center in Maryland. Thunderhead is 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. The nodes are interconnected using Myrinet 2000 connectivity, and the operating system is Linux Fedora Core. Our parallel algorithm was developed in the C++ programming language, using calls to message passing interface (MPICH was the message-passing library used).

As shown by Table 3, the time for extracting a suitable set of border training samples in parallel can be substantial. Specifically, the most significant times reported in the table correspond to the training stage, which includes finding a set of border training samples (10% of available ground-truth per class in this example) and then moving all the training vectors from the master to every worker at the beginning of the training phase (time for training). In contrast, the times reported for the classification phase after gathering the votes from each of the pair results are much lower. This stage is performed sequentially on a single processor. As shown in Table 3, the speedup (performance gain with regards to executing a sequential version of the algorithm using one processor) starts growing linearly and then saturates around 30 for 64 processors and slightly declines afterwards. The decline is due to the limited communication bandwidth among the processors, i.e., as the number of processors increases there will be more data collision and thereby delays. Here, the saturation is likely due to variations in processing times in the training phase which depend on the number of training vectors for each pair classification.

In order to illustrate the performance of parallel morphological algorithms, Table 4 shows the total time spent on Thunderhead by a parallel version of our spatial-spectral endmember extraction and unmixing algorithm using a case study where the algorithm performs  $I_{max} = 5$  iterations, reported to produce the most accurate endmember extraction results in Table 2. From the results addressed in Table 4, we can conclude that our parallel algorithm scaled very well, even for a large number of processors, resulting in a total processing time of 33 seconds for the full AVIRIS Cuprite image when 512 processors were used, as opposed to 2 hours and 37 minutes of computation when a single processor was used. Contrary to common perception that spatial-spectral techniques are too computationally demanding for practical use, these results indicate that such combined approaches may indeed be very appealing for parallel design and implementation, not only due to the window-based nature of such algorithms, but also because they can greatly reduce sequential computations at the master node and involve only minimal communication between the parallel tasks, namely, at the beginning and ending of such tasks.

## 6. CONCLUSIONS AND FUTURE PERSPECTIVES

The introduction of the concept of hyperspectral imaging has significantly influenced the evolution of remotely sensed data classification techniques. With hundreds of spectral channels now available, the sampled pixel spectra contain enough detail to allow spectroscopic principles to be applied for image understanding. The array of analytical techniques regularly used in hyperspectral data classification encompasses very different mathematical formalisms, some of them already exploited in other fields such as multispectral imaging. However, the special characteristics of hyperspectral images pose new processing problems, not to be found in other types of remotely sensed data:

- The high-dimensional nature of hyperspectral data introduces important limitations in supervised, full-pixel classifiers, such as the limited availability of training samples or the inherently complex structure of the data.
- There is a need to integrate the spatial and spectral information to take advantage of the complementarities that both sources of information can provide, in particular, for unsupervised mixed-pixel classifiers.
- There is a need to develop cost-effective parallel implementations of hyperspectral data classification algorithms, able to speed up algorithm performance and to satisfy the extremely high computational requirements of time-critical remote sensing applications.

In this paper, we have taken a necessary first step towards the understanding and assimilation of the above aspects in the design of innovative hyperspectral data classification techniques. In particular, new trends in algorithm design (such as the joint use of spatial and spectral information or the appropriate exploitation of limited training samples) have been specifically addressed. Parallel processing support for some of the proposed algorithms has also been developed and discussed. Performance data reported in two highly representative application examples reflect the versatility that currently exists in the design of approaches undertaken to solve real problems. Hyperspectral data classification exemplifies a subject area that has drawn together an eclectic collection of participants, but increasingly this is the nature of many endeavors at the cutting edge of science and technology. In this regard, the collection of data classification techniques presented in this work reflects the increasing sophistication of a field that is rapidly maturing at the intersection of many different disciplines.

To conclude this paper, we provide our vision on the future potential of hyperspectral data classification techniques such as those discussed in this work. As demonstrated by our experimental results, techniques are rapidly changing from *hard* classifiers to *soft* classifiers, such as different types of SVMs discussed in this paper. This is because these classifiers are better suited to cope with the extremely high dimensionality of the data (which will continue increasing in future years as ultraspectral imaging represents the new frontier of remote sensing data analysis), and also with the limited availability of training samples. In this regard, we anticipate that the full adaptation of *soft* classifiers to mixed-pixel classification problems (e.g., via multi-regression and robust training sample selection algorithms) may push the frontiers of hyperspectral data classification to new application domains. Further developments on the joint exploitation of the spatial and the spectral information in the input data are also needed to complement initial approximations to the problem of interpreting the data in unsupervised fashion, such as the morphological approaches described in this work, thus being able to cope with the dramatically enhanced spatial and spectral capabilities expected in the design of future imaging spectrometers. Advances in high performance computing environments including clusters of computers and distributed grids, as well as specialized hardware modules such as field programmable gate arrays (FPGAs) or graphics processing units (GPUs), will also be crucial to help increase algorithm efficiency and meet timeliness needs in many remote sensing applications. As a result, we anticipate that the future potential of hyperspectral data classification methods will be largely defined by their suitability for being efficiently implemented in parallel.

## ACKNOWLEDGMENTS

The author would like to gratefully thank J. A. Benediktsson, J. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton and G. Trianni for their contributions to a review paper on recent advances in techniques for hyperspectral image processing. Their outstanding contributions inspired some of the developments presented in this work.

## REFERENCES

1. A. F. H. Goetz, G. Vane, J. E. Solomon, and B. N. Rock, "Imaging spectrometry for earth remote sensing," *Science* **228**, pp. 1147–1153, 1985.
2. R. O. Green, "Imaging spectroscopy and the airborne visible-infrared imaging spectrometer (AVIRIS)," *Remote Sensing of Environment* **65**, pp. 227–248, 1998.
3. C.-I. Chang, *Hyperspectral imaging: techniques for spectral detection and classification*, Kluwer Academic and Plenum Publishers, New York, 2003.
4. D. A. Landgrebe, *Signal theory methods in multispectral remote sensing*, John Wiley and Sons, Hoboken, NJ, 2003.
5. J. A. Richards, "Analysis of remotely sensed data: the formative decades and the future," *IEEE Trans. Geoscience and Remote Sensing* **43**, pp. 422–432, 2005.
6. A. Plaza, P. Martinez, R. Perez, and J. Plaza, "A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data," *IEEE Trans. Geoscience and Remote Sensing* **42**, pp. 650–663, 2004.
7. J. B. Adams, M. O. Smith, and P. E. Johnson, "Spectral mixture modeling: a new analysis of rock and soil types at the viking lander 1 site," *Journal of Geophysical Research* **91**, pp. 8098–8112, 1986.
8. G. M. Foody and A. Mathur, "Toward intelligent training of supervised image classifications: directing training data acquisition for svm classification," *Remote Sensing of Environment* **93**, pp. 107–117, 2004.
9. J. A. Gualtieri and R. F. Crompt, "Support vector machines for hyperspectral remote sensing classification," in *Advances in Computer-Assisted Recognition*, R. J. Mericsko, ed., *Proceedings of SPIE* **3584**, pp. 221–232, 1999.
10. P. Gamba, F. Dell'Acqua, A. Ferrari, J. A. Palmason, and J. A. Benediktsson, "Exploiting spectral and spatial information in hyperspectral urban data with high resolution," *IEEE Geoscience and Remote Sensing Letters* **1**, pp. 322–326, 2004.
11. P. Soille, *Morphological image analysis: principles and applications*, Springer-Verlag, Berlin, 2003.
12. J. A. Benediktsson, J. A. Palmason, and J. R. Sveinsson, "Classification of hyperspectral data from urban areas based on extended morphological profiles," *IEEE Trans. Geoscience and Remote Sensing* **42**, pp. 480–491, 2005.
13. A. Plaza, D. Valencia, J. Plaza, and P. Martinez, "Commodity cluster-based parallel processing of hyperspectral imagery," *Journal of Parallel and Distributed Computing* **66**, pp. 345–358, 2006.
14. G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters* **3**, pp. 93–97, 2006.
15. L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive svm for the semisupervised classification of remote sensing images," *IEEE Trans. Geoscience and Remote Sensing* **44**, pp. 3363–3373, 2006.
16. K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Networks* **12**, pp. 181–202, 2001.
17. J. W. Boardman, "Automating spectral unmixing of aviris data using convex geometry concepts," in *Summaries of Airborne Earth Science Workshop*, R. O. Green, ed., *JPL Publication* **93-26**, pp. 111–114, 1993.
18. C. Lee and D. A. Landgrebe, "Decision boundary feature extraction for neural networks," *IEEE Trans. Neural Networks* **8**, pp. 75–83, 1997.
19. M. E. Winter, "Algorithm for fast autonomous spectral endmember determination in hyperspectral data," in *Imaging Spectrometry V*, M. R. Descour and S. S. Shen, eds., *Proceedings of SPIE* **3753**, pp. 266–275, 1999.
20. K. Staenz, R. Neville, J. Levesque, T. Szeredi, V. Singhroy, G. Borstad, and P. Hauff, "Evaluation of CASI and SFSI hyperspectral data for environmental and geological applications: two case studies," *Canadian Journal of Remote Sensing* **25**, pp. 311–322, 1999.
21. A. Plaza, P. Martinez, J. Plaza, and R. Perez, "Dimensionality reduction and classification of hyperspectral image data using sequences of extended morphological transformations," *IEEE Trans. Geoscience and Remote Sensing* **43**, pp. 466–479, 2005.