Contents lists available at ScienceDirect

Remote Sensing of Environment

journal homepage: www.elsevier.com/locate/rse

Recent advances in techniques for hyperspectral image processing

Antonio Plaza [a,*], Jon Atli Benediktsson [b], Joseph W. Boardman [c], Jason Brazile [d], Lorenzo Bruzzone [e], Gustavo Camps-Valls [f], Jocelyn Chanussot [g], Mathieu Fauvel [g,b], Paolo Gamba [h], Anthony Gualtieri [i,j], Mattia Marconcini [e], James C. Tilton [i], Giovanna Trianni [h]

[a] *Department of Technology of Computers and Communications, University of Extremadura, Cáceres, Spain*
[b] *Faculty of Electrical and Computer Engineering, University of Iceland, Reykjavik, Iceland*
[c] *Analytical Imaging and Geophysics, LLC Boulder, Colorado, USA*
[d] *Department of Geography, University of Zurich, Switzerland*
[e] *Department of Information Engineering and Computer Science, University of Trento, Italy*
[f] *Department of Electronics Engineering, University of Valencia, Spain*
[g] *GIPSA-Lab, Grenoble Institute of Technology, Grenoble, France*
[h] *Department of Electronics, University of Pavia, Italy*
[i] *Code 606.3, NASA's Goddard Space Flight Center, Greenbelt, Maryland, USA*
[j] *Global Science and Technology, Greenbelt, Maryland, USA*

ARTICLE INFO

Article history:
Received 21 July 2006
Received in revised form 25 June 2007
Accepted 27 July 2007

Keywords:
Classification
Hyperspectral imaging
Kernel methods
Support vector machines
Markov random fields
Mathematical morphology
Spatial/spectral processing
Spectral mixture analysis
Endmember extraction
Parallel processing

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. Currently, there is a need for standardized data processing techniques able to take into account the special properties of hyperspectral data. In this paper, we provide a seminal view on recent advances in techniques for hyperspectral image processing. 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. Performance of the discussed techniques is evaluated in different analysis scenarios. To satisfy time-critical constraints in specific applications, we also develop efficient parallel implementations of some of the discussed algorithms. Combined, these parts provide an excellent snapshot of the state-of-the-art in those areas, and offer a thoughtful perspective on future potentials and emerging challenges in the design of robust hyperspectral imaging algorithms.

1. Introduction

Imaging spectroscopy (Goetz et al., 1985), also known as 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. The concept of imaging spectroscopy originated in the 1980's, when A. F. H. Goetz and his colleagues at NASA's Jet Propulsion Laboratory began a revolution in remote sensing by developing new instruments such as the Airborne Imaging Spectrometer (AIS), then called AVIRIS, for Airborne Visible Infra-Red Imaging Spectrometer (Green, 1998). This system is now able to cover the wavelength region from 0.4 to 2.5 μm using more than two hundred spectral channels, at nominal spectral resolution of 10 nm.

The special characteristics of hyperspectral datasets pose different processing problems, which must be necessarily tackled under specific mathematical formalisms, such as classification and segmentation (Jia et al., 1999) or spectral mixture analysis (Adams et al., 1986; Smith et al., 1990a,b). For instance, several machine learning and image processing techniques have been applied to extract relevant information from hyperspectral data during the last decade (Varshney & Arora, 2004). Taxonomies of remote sensing data processing algorithms (including hyperspectral analysis methods) have been developed in the literature (King, 2003; Keshava & Mustard, 2002; Richards, 2005). It should be noted, however, 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 (Tadjudin & Landgrebe, 1998).

The importance of analyzing spatial and spectral patterns simultaneously has been identified as a desired goal by many scientists devoted to multidimensional data analysis. This type of

[*] Corresponding author. Tel.: +34 925257000x51662.
E-mail address: aplaza@unex.es (A. Plaza).

0034-4257/$ – see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.rse.2007.07.028

processing has been approached in the past from various points of view. For instance, several possibilities are discussed in (Landgrebe, 2003) for the refinement of results obtained by spectral-based techniques in multispectral imaging through a second step based on spatial context. Such contextual classification, extended also to hyperspectral images (Jimenez et al., 2005), accounts for the tendency of certain ground cover classes to occur more frequently in some contexts than in others. This approach consists of two parts: the definition of a pixel neighborhood (surrounding each pixel) and the performance of a local operation so that the pixel may be changed into the label mostly represented in the window that defines the neighborhood. This simple operation separates spatial from spectral information, and thus the two types of information are not treated simultaneously.

In certain applications, however, the integration of high spatial and spectral is mandatory to achieve sufficiently accurate mapping and/or detection results. For instance, urban area mapping requires sufficient spatial resolution to distinguish small spectral classes, such as trees in a park, or cars on a street (Gamba et al., 2004; Chanussot et al., 2006). This poses two main challenges:

(1) We need to manage very high-dimensional data volumes in which the spatial correlation between spectral responses of neighboring pixels can be potentially high. As a result, there is a need to incorporate the spatial arrangement of the data in the development of robust analysis techniques.

(2) Processing algorithms need to become more knowledge-based. With finer spatial resolutions, subtle details which can greatly improve scene interpretation may also be misleading in certain applications. This suggests that *a priori* knowledge about shape, texture, spatial relationships and patterns may be used to improve the characterization of single elements, as well as the whole scene.

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 another challenging goal (Foody & Mathur, 2004; Foody & Arora, 1996). 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 (Hughes, 1968). High-dimensional spaces have been demonstrated to be mostly empty (Jimenez & Landgrebe, 1998), thus making density estimation even more difficult.

One possible approach to handle the high-dimensional nature of hyperspectral data sets is to consider the geometrical properties rather than the statistical properties of the classes. The good classification performance demonstrated by support vector machines (SVMs) using spectral signatures as input features (Gualtieri & Cromp, 1998; Gualtieri et al., 1999; Watanachaturaporn et al., 2006) is further increased in this work by taking advantage of semi-supervised learning and contextual information. The latter is done through the combination of dedicated kernels to spectral and contextual information, while in the former the learning is provided with some supervised information in addition to the wealth of unlabeled data. Among the great many methods proposed in the literature for such approaches (Dempster77, Chapelle06), we focus on the transductive SVM for semi-supervised learning (Bruzzone et al., 2006), and in the composite kernels methodology (Camps-Valls et al., 2006) for contextual information integration.

Our main goal in this paper is to provide a seminal view on recent advances in techniques for hyperspectral image analysis which can successfully deal with the dimensionality problem and take into account both the spectral and spatial properties of the data. To address the need for knowledge-based 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 particularly explores the extension of mathematical morphology (Soille, 2003) to hyperspectral imagery for spatial/spectral data processing. 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 (Benediktsson et al., 2003). Here, we revisit and improve morphological techniques which deal with the full spectral information available in the data, including classification techniques based on extended morphological profiles (Benediktsson et al., 2005) and spectral unmixing techniques able to extract image endmembers using spatial and spectral information simultaneously (Plaza et al., 2002) and exploit the inherent convexity of mixed pixels in the unmixing stage.

In addition to mathematical morphology-based approaches, Markov random fields (Chellappa & Jain, 1993; Kasetkasem et al., 2005) can also be used to model the spatial neighborhood of a pixel as a spatially distributed random process, and attempt a regularization via the minimization of an energy function. In this work, we introduce a new Markov-based classification framework in which a neuro-fuzzy classifier is first used to perform classification in the spectral domain (taking also advantage of consolidated feature extraction techniques), and the resulting output is then fed to a spatial analysis stage combined with spectral re-classification.

Finally, we also explore the concept of hierarchical segmentation, which produces a set of several image segmentations of the same image at different levels of detail (Beaulieu & Goldberg, 1989; Tilton, 1998), in the context of hyperspectral image analysis. Although this technique has been applied in the past to multispectral data sets (Tilton et al., 2006), the application to hyperspectral imaging presented in this work represents a completely novel contribution. While integrated spatial/spectral developments 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 necessarily become a requirement in many remote sensing missions, especially with the advent of low-cost systems such as commodity clusters (Brazile et al., 2003; Plaza et al., 2006). In order to address this relevant issue, this work also explores the development of parallel processing support for some of the data processing algorithms discussed in the paper.

The remainder of the paper is organized as follows. Section 2 presents a set of hyperspectral data sets that will be used for illustrative purposes throughout the manuscript. Section 3 explores classification of hyperspectral data using kernel methods, introducing new approaches based on the standard SVM formulation and providing relevant processing examples. Section 4 focuses on integrated spatial/spectral data processing techniques, including morphological and Markov-based techniques for classification, hierarchical segmentation, and endmember extraction. Section 5 outlines the development of parallel versions of some of the algorithms considered in Sections 3 and 4. Section 6 summarizes the processing achievements presented throughout the paper and provides a short outlook on the future potential of the methods discussed. Finally, Section 7 concludes with some remarks.

2. Hyperspectral data sets

2.1. AVIRIS Indian Pines data set

The Indian Pines scene was gathered by the AVIRIS instrument in 1992. It consists of 145 × 145 pixels and 16 ground-truth classes, ranging from 20 to 2468 pixels in size. It was acquired over a mixed agricultural/forested region in NW Indiana. The data set represents a very challenging land-cover classification scenario, in which the primary crops of the area (mainly corn and soybeans) were very early in their growth cycle, with only about 5% canopy cover. Discriminating among the major crops under these circumstances can be very difficult (in particular, given the moderate spatial resolution of 20 m). The data is available online from http://dynamo.ecn.purdue.edu/biehl/MultiSpec. We removed 20 noisy bands covering the region of water absorption, and worked with 200 spectral bands.

2.2. ROSIS urban data over Pavia, Italy

These data were collected in 2003 by the ROSIS sensor, with spectral coverage ranging from 0.43 to 0.86 μm. The data is atmospherically corrected and has spatial resolution of 1.3-m/pixels with 115 spectral bands. Three subsets were considered:

(1) Subset #1. This subset, with 492 × 1096 pixels in size, was collected over Pavia city centre, Italy. It contains 102 spectral channels after removal of noisy bands [see Fig. 1(a) for a color composite]. Nine ground-truth classes were considered in experiments: Asphalt, Meadow, Gravel, Trees, Metal Sheet, Bare soil, Bitumen, Bricks and Shadow.

(2) Subset #2. This subset, with size of 610 × 340 pixels, is centered at University of Pavia [see Fig. 1(b) for a color composite]. Nine ground-truth classes were considered in experiments: Water, Trees, Grass, Parking lot, Bare Soil, Asphalt, Bitumen, Tiles and Shadow.

(3) Subset #3. The third scene comprises subset #2 plus an additional city area at the left of the imaged area (not displayed).

2.3. AVIRIS Cuprite data set

The AVIRIS Cuprite scene is available online from the AVIRIS website at from http://aviris.jpl.nasa.gov/html/aviris.freedata.html. We use reflectance data in order to relate our results to reference spectral libraries. The scene selected for experiments is the one labeled as f970619t01p02_r02_sc03.a.rfl. This scene comprises a relatively large area (614 × 512 pixels and 20-m pixels) and 224 spectral bands between 0.4 and 2.5 μm. Bands 1–3, 105–115 and 150–170 were removed prior to the analysis due to water absorption and low SNR in those bands. The site is well understood mineralogically (Clark et al., 1993), and has several exposed minerals of interest. Reference ground signatures of those minerals are available in the form of a U.S. Geological Survey library (http://speclab.cr.usgs.gov/spectral-lib.html). These signatures will be used to assess endmember signature purity in this work.

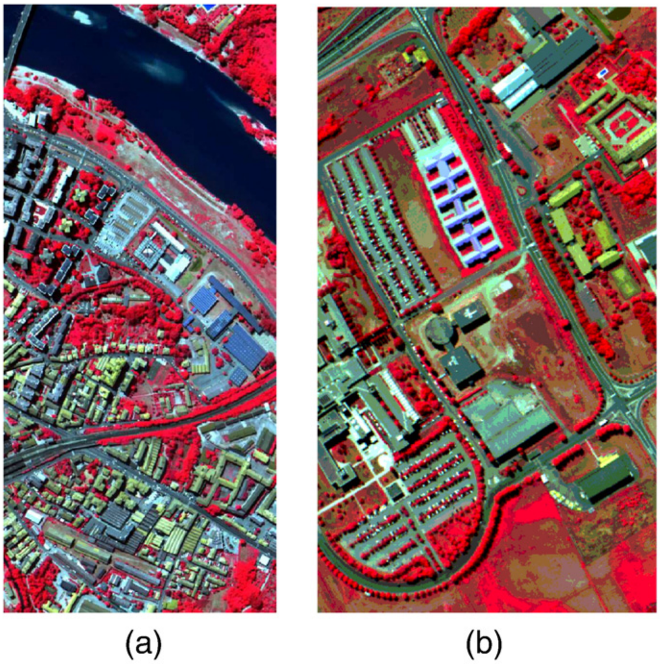

Fig. 1. ROSIS urban hyperspectral data: subset #1 (a) and subset #2 (b).

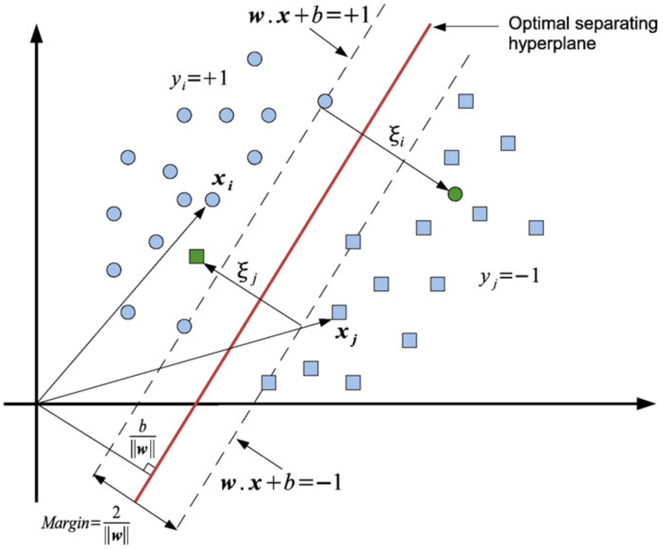

Fig. 2. Classification of the non-linearly separable case by SVMs.

3. Classification of hyperspectral data using kernel methods

This section first investigates the problem of local variation of spectral energy by introducing a family of scale-invariant kernels for SVM training. Then, ill-posed problems (induced by the limited amount of training samples) are addressed by the incorporation of a transductive approach to classification. Finally, a new family of kernels which includes contextual/textural information in the classification process is presented and discussed.

3.1. SVM formulation and the use of different kernel functions

The SVM was first investigated by (Boser et al., 1992; Cortes & Vapkik, 1995) as a binary classifier. Given a training set $S = \{(\Phi(\mathbf{x}))_i, y_i) | i \in 1, n]\}$ projected into an Hilbert space $\mathcal{H}$ by some mapping Φ, the SVM separates the data by an Optimal Hyperplane H_p that maximizes the margin, see Fig. 2. Allowing some training errors, H_p is found by jointly maximizing the margin $||\mathbf{w}||$ and minimizing the sum of errors $\Sigma_{i=1}^{n} \xi_i$ (Scholkopf & Smola, 2002). The convex optimization problem is solved by considering the dual optimization through the use of Lagrange multipliers:

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

Using kernel functions k it is possible to compute implicitly the inner product in $\mathcal{H}$ in the original space (Muller et al., 2001): $\langle \Phi(\mathbf{x})_i, \Phi(\mathbf{x})_j \rangle_{\mathcal{H}} = k(\mathbf{x}_i \mathbf{x}_j)$. 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 $y_u = sgn(\Sigma_{i=1}^{n} y_i \alpha_i k(\mathbf{x}_u, \mathbf{x}_i) + b)$. Different kernels leads to different SVMs. 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. (Mercier & Lennon, 2003) defined a spectral angle kernel $k_{SAM}(\mathbf{x}, \mathbf{z}) = \exp(-\gamma \alpha(\mathbf{x}, \mathbf{z})^2)$, $\alpha(\mathbf{x}, \mathbf{z}) = \arccos\left(\frac{\langle \mathbf{x}, \mathbf{z} \rangle}{||\mathbf{x}||.||\mathbf{z}||}\right)$ using the scale invariance property of the spectral data. Other information, such as texture or spatial context could also improve the kernel definitions, as will be shown in experiments. The extension of SVM to the multi-class cases is usually done by combining several binary classifiers. Two classical procedures are the *one versus the rest* and the *one versus one* (Scholkopf & Smola, 2002). Using the former strategy, we can illustrate the performance of SVMs with different kernel functions by processing the scene labeled as ROSIS subset #1. To do so, small training sets were randomly extracted from the training set given in Table 1, composed of 10, 20, 40, 60, 80 and 100 pixels by class, respectively. The SVMs were then trained with each of these training subsets and then evaluated with the entire test set. Each experiment was repeated five times, and the mean accuracy values were reported. Three kernels were used: Gaussian RBF (k_{gauss}), polynomial (k_{poly}) and SAM-based (k_{SAM}). For the training process, kernel parameters (γ, θ, P and C) were adjusted to maximize the estimated overall accuracy, which was computed using a fivefold *cross validation*. Table 2 summarizes the results obtained using the three kernels. These values were extracted from the *confusion matrix* (Foody, 2002). The average accuracies obtained after averaging the classification scores obtained for all considered classes are also reported.

From Table 2, it can be seen that SVMs generalize very well: with only 10 training pixels per class, more than 90% accuracy is reached by all kernels. 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 that the classification accuracy is correlated with the training set size. But the difference in terms of accuracy is fairly low: for instance, with the k_{gauss} kernel, the overall accuracy obtained with only 10 training pixels per class is only 2, 7% lower than the overall accuracy obtained with the complete training set. On the other hand, the use of the k_{SAM} kernel gives slightly degraded classification results for the overall and average accuracies and the kappa coefficient. Finally, the k_{poly} kernel seems to need more training samples than the two other kernels to perform appropriately. Summarizing, experimental results on Table 2 reveal that the k_{gauss} and the k_{poly} kernels seem to perform almost equally, with a slight advantage for the k_{gauss}. The reason why the k_{SAM} kernel performs worse than the others is because it does not use the energy of each pixel-spectrum (the norm). The differences between classes are in the shape of the pixel-spectrum (can be seen as the angle) and the energy of the pixel-spectrum (the norm). Using the k_{poly} or the k_{gauss} kernels, both types of information are used, thus leading to better results.

3.2. Exploitation of labeled and unlabeled samples for semi-supervised learning

In this subsection, we revise a TSVM method recently presented in the literature (Bruzzone et al., 2006, 2007; Chi & Bruzzone, 2007), designed to address the problem of hyperspectral image classification. TSVMs are based on specific iterative algorithms, which gradually search the optimal discriminant hyperplane in the feature space with a transductive process that incorporates unlabeled samples in the training phase. In the semi-supervised framework, two datasets are defined: a labeled training data set $\{\mathbf{x}_l,y_l\}_{l=1}^{n}$ and an unlabeled dataset $\{\mathbf{x}_u^*,y_u^*\}_{u=1}^{m}$, where $\mathbf{x}_l,\mathbf{x}_u^*\in\mathbb{R}^N$ and $y_l,y_u^*\in\{-1,+1\}$. Like in the inductive case, let us define a nonlinear mapping $\Phi(\cdot)$, usually to a higher (possibly infinite) dimensional (Hilbert) space, $\Phi:\mathbb{R}^N\rightarrow\mathcal{H}$. In the TSVM algorithm, we solve the following problem, in which both labeled $\mathbf{x}_l$ and unlabeled $\mathbf{x}_u^*$ samples are taken into account:

Table 1
Number of training and test sites used in subset #1 of the ROSIS urban data.

Class		Samples	
No	Name	Train	Test
1	Water	745	6527
2	Trees	785	6508
3	Meadow	797	2900
4	Brick	485	2140
5	Soil	820	6549
6	Asphalt	816	7555
7	Bitumen	808	6479
8	Tile	223	3122
9	Shadow	195	2165
Total		5536	103,504

Table 2
Overall, average classification accuracies (in percentage) and kappa coefficient obtained after applying Gaussian, polynomial and SAM kernels to subset #1 of the ROSIS urban data.

Training set size		10	20	40	60	80	100	All
k_{gauss}	Overall accuracy	93.85	94.51	94.51	94.71	95.36	95.29	96.45
	Average accuracy	88.76	91.00	92.66	92.04	93.24	93.39	95.08
	Kappa	0.90	0.91	0.92	0.91	0.92	0.92	0.94
k_{poly}	Overall accuracy	92.34	92.77	94.20	94.07	94.29	94.81	96.03
	Average accuracy	87.87	88.91	91.74	92.41	92.31	93.35	94.91
	Kappa	0.87	0.88	0.90	0.90	0.90	0.91	0.93
k_{SAM}	Overall accuracy	93.32	93.87	93.79	94.23	94.40	94.54	95.56
	Average accuracy	86.36	88.64	91.26	91.67	91.89	92.61	94.26
	Kappa	0.89	0.90	0.90	0.90	0.90	0.91	0.93

$$\min_{\mathbf{w},\xi_i,\xi_l^*,b}\left\{\frac{1}{2}\parallel\mathbf{w}\parallel^2+C\sum_{l=1}^{n}\xi_l+C^*\sum_{u=1}^{d}\xi_u^*\right\},\qquad(2)$$

where $\mathbf{w}$ and b define a linear classifier in the feature space, and d denotes the number of selected unlabeled samples in the transductive process ($d\leq m$). Note that, with this notation, if $d=m$ all the unlabeled samples are used for transductive learning, like in (Chen et al., 2003). This formulation leads to the following decision function, implemented for any test vector $\mathbf{x}$:

$$f(\mathbf{x})=\operatorname{sgn}\left(\sum_{l=1}^{n}y_l\alpha_l k(\mathbf{x}_l,\mathbf{x})+\sum_{u=1}^{d}y_u^*\alpha_u^*k\left(\mathbf{x}_u^*,\mathbf{x}\right)+b\right),\qquad(3)$$

where b can be easily computed from the α_l and α_u^* that are neither 0 nor C, as commonly done in the standard SVM algorithm (Scholkopf & Smola, 2002). At this level, several interesting issues must be noted:

Selection of transductive samples. From both the upper (positive) and the lower (negative) side of the margin, $P\geq 1$ transductive samples closest to the margin bounds are assigned to the label "+1" and "−1", respectively. If the number of unlabeled samples in one side of the margin is lower than P, the labeling is done anyway. A dynamical adjustment is necessary for taking into account that the position of the hyperplane changes at each iteration.
Regularization parameter of transductive samples. Typically, it is expected that the cost for the errors occurred on the transductive samples should be lower with respect to that occurred on the original training samples. Therefore, we assign to C^* a moderate constant value with respect to C in order to keep the risk of misclassification under control.
Convergence criterion. In practice, it is assumed that the convergence is reached if both the number of mislabeled samples at the previous iteration $t-1$ and the number of remaining unlabeled transductive samples in the margins at the current iteration are lower than or equal to $a\times m$, where a is fixed *a priori* and tunes the sensitivity of the learning process. A reasonable empirical choice is $a=0.02$.
Multiclass problems. The transductive inference forces the employment of a One-Against-All (OAA) strategy, when multiclass problems are addressed. At each iteration all the unlabeled samples have to be labeled. Thus, it is not possible, for example, to take into account the patterns supposed to belong to two different classes, without labeling all the others.

In order to illustrate the performance of the proposed TSVM approach, we used a subset of the AVIRIS Indian Pines image. From the

Table 3
Number of training and test samples used in the AVIRIS Indian Pines subset scene.

Class		Samples	
No	Name	Train	Test
1	Corn-no till	742	692
2	Corn-min till	442	392
3	Grass/Pasture	260	237
4	Grass/Trees	389	358
5	Hay-windrowed	236	253
6	Soybean-no till	487	481
7	Soybean-min till	1245	1223
8	Soybean-clean till	305	309
9	Woods	651	643
Total		4757	4588

16 different land-cover classes available in the original ground-truth, 7 were discarded since an insufficient number of training samples were available and thus, this fact would dismiss the planned experimental analysis. The remaining 9 classes were used to generate a set of 4757 training samples (for the learning phase) and 4588 test samples (for validating their performance). See Table 3 for details. We adopted a Gaussian RBF kernel because it involves less numerical difficulties than sigmoid, polynomial and linear kernels, and only one parameter (i.e., the Gaussian width) has to be tuned.[1] A *one versus the rest* architecture made up of nine different binary classifiers was adopted both for SVMs and TSVMs. With the above settings, we observed that the classification accuracies decreased significantly when few training samples were taken into account (i.e., 5%–25% of the original training set), which confirms the need for improving the performances when limited *a priori* information is available. We carried out experiments with the transductive approach in this particular framework. It should be underlined that this problem is particularly complex and ill-posed because the number of training patterns is only slightly higher than (or even comparable to) the size of the feature space.

Specifically, we carried out experiments taking into account three different sets containing 5%, 10%, and 25% of the original training samples, respectively. As representative classification metrics, we used the overall accuracy and the kappa coefficient (Foody, 2002). The accuracies obtained by SVMs proved to be particularly stable with respect to the random selection of labeled patterns, given a fixed percentage of samples to be chosen. Therefore, to test the proposed technique, we decided to use always the same subsets of labeled patterns. Specifically, we carried out experiments for different values of the C^* parameter, varying (at the same time) the number of P patterns labeled from both sides of the margin.

The performances provided by the TSVM framework, both in terms of overall accuracy and kappa coefficient, improved the ones yielded by the SVMs when C^* had low values (i.e., in the range of between 0.05–0.5): this is a further proof that the investigated problem was very complex and it was necessary to assign a low cost to errors occurred on the transductive samples. Moreover, in most of the nine considered binary sub-problems, a high number of unlabeled patterns fell into the margin bound. For this reason, we obtained the best results when only a small portion of the transductive patterns at each iteration was labeled. Table 4 reports the best results obtained. As one can see, we obtained improvements around 0.04 for the kappa coefficient and 3%–4% for the overall accuracy. These results confirm that semi-supervised TSVMs seem a promising approach to deal with hyperspectral data in challenging land-cover classification scenarios.

It is worth noting that we expect a further increase of the gap between the accuracy provided by TSVMs and that exhibited by standard SVMs when images associated to larger areas are considered, where the non-stationarity of spectral signatures of classes makes the proposed approach intrinsically more effective. For further details on this last issue and on TSVMs (or semisupervised SVMs) in hyperspectral images we refer the reader to (Bruzzone et al., 2006, 2007; Chi & Bruzzone, 2007).

3.3. Integration of contextual/textural information in kernel methods

In order to incorporate the spatial context into kernel-based classifiers, a pixel entity $\mathbf{x}_i$ is redefined simultaneously both in the spectral domain using its spectral content, $\boldsymbol{x}_i^{\omega} \in \mathbb{R}^{N_\omega}$, and in the spatial domain by applying some feature extraction to its surrounding area, $\boldsymbol{x}_i^{s} \in \mathbb{R}^{N_s}$, which yields N_s spatial (contextual) features, e.g., the mean or standard deviation *per* spectral band. These separated entities lead to two different kernel matrices, which can be easily computed using any suitable kernel function that fulfills Mercer's conditions. At this point, one can sum spectral and textural dedicated kernel matrices (k_ω and k_s, respectively), and introduce the cross-information between textural and spectral features ($k_{\omega s}$ and $k_{s\omega}$) in the formulation. This simple methodology yields a full family of new composite methods for hyperspectral data classification, including a stacked features approach, a direct summation kernel, a weighted summation kernel, a cross-information kernel, and kernels for improved versatility (all described in (Camps-Valls et al., 2006)).

To illustrate the advantages of integrating contextual information in the SVM framework through the incorporation of composite kernels, we use the AVIRIS Indian Pines data set. Here, we used the polynomial kernel with $d = \{1,\ldots 10\}$ for modeling the spectral features according to previous results in the literature (Gualtieri et al., 1999; Camps-Valls & Bruzzone, 2005; Camps-Valls et al., 2007), and the Gaussian RBF kernel (with $\sigma = \{10^{-1},\ldots,10^{3}\}$) for modeling the spatial features, according to the locality assumption in the spatial domain. For the "stacked" ($k_{\{s,\omega\}}$) and cross-information ($k_{s\omega}$, $k_{\omega s}$) approaches, we used the polynomial kernel. The penalization factor in the SVM was tuned in the range $C = \{10^{-1},\ldots,10^{7}\}$. A *one-against-one* multi-classification scheme was adopted in all cases.

The most simple but powerful spatial features $\mathbf{x}_i^s$ that can be extracted from a given region are based on moment criteria. Here, we take into account the first two moments to build the spatial kernels. Two situations were considered: (i) using the mean of the neighborhood pixels in a *window* $(dim(\mathbf{x}_i^s) = 200)$ *per* spectral channel or (ii) using the mean and standard deviation of the neighborhood pixels in a *window* per spectral channel ($dim(\mathbf{x}_i^s) = 400$). Inclusion of higher order moments or cumulants did not improve the results in our case study. The *window* size was varied between 3 × 3 and 9 × 9 pixels in the training set. Table 5 shows the validation results (averaged over 10 random realizations) from six kernel classifiers: spectral (k_ω), contextual (k_s), the stacked approach ($k_{\{s,\omega\}}$), and the three presented composite kernels on the AVIRIS Indian Pines scene. In addition, two standard methods are included for baseline comparison: bLOOC + DAFE + ECHO, which use contextual and spectral information to classify homogeneous objects, and the Euclidean classifier (Tadjudin & Landgrebe, 1998), which only uses the spectral information. All models are compared numerically (using the overall accuracy) and statistically, using the kappa and Wilcoxon rank sum tests (Foody, 2002).

Table 4
Overall accuracy (in percentage) and kappa coefficient obtained by (inductive) SVMs and the proposed (transductive) TSVMs, applied to the subset AVIRIS Indian Pines scene using three different subsets, containing 5%, 10%, and 25% of the original training set ($C^* = 0.1$, $P = 0.01\ m$).

Percentage of training samples	Number of samples	Overall accuracy		Kappa	
		SVMs	Proposed TSVMs	SVMs	Proposed TSVMs
5%	237	73.41	76.20	0.68	0.71
10%	475	76.46	80.21	0.73	0.77
25%	1189	82.17	84.83	0.79	0.82

[1] The sequential minimal optimization (SMO) algorithm (Platt, 1999) was used in the learning phase of both the standard SVM and the proposed TSVM (making proper modifications), thus exploiting the improvements proposed in (Keerthi et al.1999).

Table 5
Overall accuracy and kappa coefficient obtained for the whole AVIRIS Indian Pines scene using different classifiers.

	Overall accuracy	Kappa
Spectral classifiers[†]		
Euclidean (Tadjudin and Landgrebe, 1998)	48.23	–
bLOOC+DAFE+ECHO (Tadjudin and Landgrebe, 1998)	82.91	–
k_ω (Gualtieri and Cromp, 1998)	87.30	–
k_ω (developed in this paper)	88.55	0.87
Spatial/spectral classifiers		
Mean		
Spatial	84.55	0.82
Stacked	94.21	0.93
Summation	92.61	0.91
Weighted	**95.97**	**0.94**
Cross-terms	94.80	0.94
Summation + Stacked	95.20	0.94
Cross-terms + Stacked	95.10	0.94
Mean and standard deviation[‡]		
Spatial	88.00	0.86
Stacked	94.21	0.93
Summation	95.45	0.95
Weighted	**96.53**	**0.96**
Summation + stacked	96.20	0.95

The best scores for each class are highlighted in bold typeface. The overall accuracies that are statistically different from the best model (at 95% confidence level, as tested through paired Wilcoxon rank sum test) are underlined.

[†] Differences between the obtained accuracies reported in (Gualtieri et al.,1999) and those presented here could be due to the random sample selection, however they are not statistically significant. [‡] Note that by using mean and standard deviation features, $N_\omega \neq N_s$ and thus no cross kernels ($k_{s\omega}$ or $k_{\omega s}$) can be constructed.

Several conclusions can be obtained from Table 5. First, all kernel-based methods produce better (and more statistically significant) classification results than those reported by previous methods, such as simple Euclidean and LOOC-based method, as previously illustrated in (Gualtieri & Cromp, 1998). It is also worth noting that the contextual kernel classifier k_s alone produces good results, mainly due to the presence of large homogeneous classes. Note that the extracted textural features $\mathbf{x}_i^s$ contain spectral information to some extent as we computed them *per* spectral channel, thus they can be regarded as contextual or local spectral features. However, the accuracy is inferior to the best spectral kernel classifiers, i.e., both k_ω implemented here and in (Gualtieri et al., 1999), which demonstrates the relevance of the spectral information for hyperspectral image classification. It is also worth mentioning that all composite kernel classifiers improved the results obtained by the usual spectral kernel, which confirms the validity of the presented framework. However, the improved versatility kernels (summation or cross-terms in combination with stacked) do not improve results in our experiments, which suggest that a simpler model data specification is enough for this particular problem.

4. Integration of spatial and spectral information

This section describes further developments in the area of spatial/spectral data processing. Although in the previous section we already provided an example of this kind of integrated algorithm with the composite kernel-based approach, in this section we do not restrict our argument to data classification (addressed by the introduction of new methods based on extended morphological profiles and Markov random fields), but we also introduce a new morphological method for spectral unmixing. The section ends with an overview of the concept of hierarchical segmentation, and a discussion on its utilization for joint spatial/spectral processing of hyperspectral images.

4.1. Mathematical morphology-based classification of hyperspectral images

To analyze the structures of an image, (Benediktsson et al., 2003) have constructed the *morphological profile* (MP), stemming from the *granulometry* principle (Serra, 1982; Soille, 2003). The MP is composed of the *opening profile* (OP), which consists of an ensemble of opening by reconstruction of increasing size, and of the *closing profile* (CP), which is made with the dual operation (Soille, 2003). Opening and closing by reconstruction are connected operators that satisfy the following assertion: if the structure of the image cannot contain the structuring element, then it is totally removed, otherwise it is totally preserved. For a given structuring element, geodesic opening and closing allows one to know the size or shape of the objects present in the image: those which are deleted are smaller than the structuring element, while those which are preserved are bigger. To determine the shape or size of all elements present in an image, it is necessary to use a range of different structuring element sizes. This assumption leads to the MP:

$$MP(x,y) = \{CP_k(x,y), \ldots, f(x,y), \ldots, OP_k(x,y)\} \quad (4)$$

Spatial information (size, orientation and local contrast) are included in the MP. However, the above formulation refers to a single-band image and, therefore, the spectral information is not considered. A simple approach to deal with this problem is to extract several images that contain parts of the spectral information, and then build the MP on each of the individual images. This approach is called *extended morphological profile* (EMP).

In order to develop an extended morphological approach, characteristic images need to be first extracted from the hyperspectral data. It was suggested in (Benediktsson et al., 2005) that the first principal components (PCs) of the hyperspectral data could be used for this purpose. In this work, we propose to use the PCs that contain more than a certain amount of cumulative variance, and then build the MP on each of the individual PCs. The resulting EMP can be seen as a single stacked vector to be used as a collection of features for classification. Following the previous notation used in (5), the EMP at the pixel with spatial location (x,y) can be simply represented by:

$$MP_{\text{ext}}(x,y) = \left\{MP_{\text{PC}_1}(x,y), \ldots, MP_{\text{PC}_k}(x,y)\right\} \quad (5)$$

It should be noted that the EMP above provides an intuitive idea of both the spectral characterization of the pixel and the spatial distribution of its neighboring objects in the scene. As a result, the EMP can be used as a feature vector for subsequent classification based on a spatial/spectral criterion.

In order to illustrate the performance of this technique, we use subset #2 of the ROSIS urban data. The considered training and test samples are given in Table 6. Here, PCA was applied on the full spectrum, and the first three principal components were selected corresponding to the three largest Eigenvalues and 99% of the cumulative variance. Morphological profiles were then constructed for each component, based on 10 openings/closings by reconstruction, so each morphological profile was made of 11 bands. The structuring element of the morphological filter was a disk with a step size increment of 1 pixel. The resulting extended morphological profile was

Table 6
Information classes and training/test samples for subset #2 of the ROSIS data.

Class		Samples	
No	Name	Training	Test
1	Asphalt	548	6304
2	Meadow	540	18146
3	Gravel	392	1815
4	Tree	524	2912
5	Metal sheet	265	1113
6	Bare soil	532	4572
7	Bitumen	375	981
8	Brick	514	3364
9	Shadow	231	795
Total		3921	40002

Table 7
Overall, average classification accuracies (in percentage) and kappa coefficient after applying an SVM classifier to subset #2 of the ROSIS urban data using the original spectral information and extended morphological profiles as inputs to the classifier.

	Original spectral information	Extended morphological profile
Overall accuracy	80.99	85.22
Average accuracy	88.28	90.76
Kappa	76.16	80.86
Asphalt	83.71	95.36
Meadow	70.25	80.33
Gravel	70.32	87.61
Tree	97.81	98.37
Metal sheet	99.41	99.48
Bare soil	92.25	63.72
Bitumen	81.58	98.87
Brick	92.59	95.41
Shadow	96.62	97.68

obtained by combining the three standard morphological profiles, and contained 63 features. It should be noted that the classifiers adopted in this experiment to test the accuracy of morphological feature extraction were SVMs with a Gaussian kernel. The parameters (C,γ) were tuned using a five-fold cross validation. The entire training set was used to train the classifier, and testing results are shown in Table 7. From Table 7, it can be seen that the best overall accuracy was obtained when the extended morphological profile was used, resulting in an overall accuracy of 85.22%. On the other hand, the best kappa and average accuracies were also achieved with the extended profile. For the class-specific accuracies, the morphological approach gave higher accuracies for most classes than the classification based on the use of the spectrum. However, for the *bare soil* class, morphological processing gave the lowest accuracies. One possible explanation is that the spectral richness of the hyperspectral data cannot be fully characterized by using three principal components only for this particular class. Although the incorporation of additional components could be an alternative, we feel that the problem is more related with the nature of the PCA transform itself, which may not be the most appropriate tool for feature extraction from remotely sensed data sets. In this regard, our current experimentation is oriented towards the use of more suitable feature reduction techniques; for instance, random forests have shown remarkable preliminary results (Joelsson et al., 2006). In addition, we are also using data fusion techniques to optimize the exploitation of different classifiers and address situations like the one observed for the bare soil class, for which spectral classification is the best approach while the morphological method does better on the rest of the classes. Details can be found in (Fauvel et al., 2006a) with specific application to SVM reported in (Fauvel et al., 2006b). In spite of the issues above issue, it is clear from Table 7 that most of the individual classification results obtained using extended morphological profiles are significantly better than those obtained with the original spectral information, which allows us to conclude that morphological processing provides a good, simplified approach to perform joint spatial/spectral classification of urban hyperspectral data.

4.2. Spatial/spectral classification using Markov random fields

Another approach to characterize pixel entities using the spatial and the spectral information is the Markov random field (MRF) technique. To reduce spectral complexity prior to data modeling, discriminant analysis feature extraction (DAFE) (Landgrebe, 2003) is first applied. Then, spatial characterization is performed by modeling the spatial neighborhood of a pixel as a spatially distributed random process and attempts a regularization via the minimization of an energy functional. More precisely, let us denote by $\mathbf{g}(x,y) = \{f_1(x,y), f_2(x,y), \ldots, f_m(x,y)\}$ the pixel vector at spatial location (x,y) of the DAFE-reduced version $\mathbf{g}$ of an input hyperspectral image $\boldsymbol{f}$, where m$\leq n$ is the number of extracted features. Let us also assume that k land-cover classes $\boldsymbol{C}=\{C_1, C_2, \ldots C_k\}$ are known in advance, with the corresponding prior probabilities denoted by $P(C_1)$,$P(C_2)$,…,P(C_k). If we denote by $P(\mathbf{g}|\boldsymbol{C})$ the conditional probability density of $\mathbf{g}$ given the set $\boldsymbol{C}$, and $P(\boldsymbol{C}|\mathbf{g})$ denotes the posterior probability, then the proposed MRF classifier aims at assigning each pixel to the class which maximizes the posterior probability. This is achieved by minimizing (for each pixel vector) the following cost-function:

$$U(\mathbf{g}(x,y), C(x,y)) = \alpha U_{\text{spectral}}(\mathbf{g}(x,y), C(x,y)) + U_{\text{spatial}}(\mathbf{g}(x,y), C(x,y)), \tag{6}$$

where the term U_{spatial} is given by:

$$U_{\text{spatial}}(\mathbf{g}(x,y), C(x,y)) = \sum_{(i,j)\in G(x,y)} \beta I(C(x,y), C(i,j)) \tag{7}$$

and $G(x,y)$ denotes the local spatial neighborhood for the pixel at spatial location (x,y). It should be noted that $I(C(x,y), C(i,j)) = -1$ if $C(x,y) = C(i,j)$ and $I(C(x,y), C(i,j)) = 0$ if $C(x,y) \neq C(i,j)$. Similarly, the term U_{spectral} in (7) is given by:

$$U_{\text{spectr}}(\mathbf{g}(x,y), C(x,y)) = \frac{m}{2} ln|2\pi\Sigma_k| + \frac{1}{2}(\mathbf{g}(x,y) - \mu_k)^T \Sigma_k^{-1}(\mathbf{g}(x,y) - \mu_k), \tag{8}$$

where Σ_k and μ_k are, respectively, the class-conditional covariance matrix and mean vector for class k, and m is the number of spectral bands in the DAFE-reduced image. In this work, the novelty introduced in the standard methodology above is the use a neuro-fuzzy classifier to perform classification in the spectral domain and compute a first approximation of the posterior probabilities. The output to this step is then fed to the MRF spatial analysis stage, performed using a maximum likelihood (ML) probabilistic re-classification. In this framework, the function to be minimized is computed by integrating the pattern recognition capability of a neuro-fuzzy classifier (with superior performance on single-pixel classification (Gamba & Trianni, 2005) and the spatial/spectral nature of the probabilistic ML-based MRF framework.

The above methodology is illustrated by using subset #3 of the ROSIS urban data. For sake of a proper comparison, results by the proposed method (referred to as DAFE/MRF hereinafter) will be compared with those by obtained by a neuro-fuzzy classifier reported (among others) in (Landgrebe, 2003) for the same area, which is based on a spectral classification performed by a fuzzy ARTMAP approach (Baraldi et al., 2001), followed by a fixed-scale spatial reclassification.

Table 8 shows that the two considered approaches result in similar classification accuracies, mainly because the spatial analysis stage tends to reassign only border pixels to different classes, while pure pixels inside homogeneous regions such as roofs, parking lots or roads are essentially the same. However, the original geometrical characterization of the elements in urban areas is a very challenging problem

Table 8
Individual and overall classification accuracy (in percentage) achieved by a combined DAFE/MRF approach and by a neuro-fuzzy method for subset #3 of the ROSIS urban data.

	DAFE/MRF	Neuro-fuzzy
Overall accuracy	97.27	97.29
Water	99.04	99.71
Trees	91.27	93.19
Grass	97.09	94.80
Parking lot	76.83	71.99
Bare soil	93.33	93.36
Asphalt	99.65	81.87
Bitumen	88.45	96.42
Tiles	98.33	99.98
Shadow	99.86	99.93

(even for a “manual” classification approach) as indicated by Fig. 1. In this case, it is very difficult to define object borders in inner city areas, because of the limited ground spatial resolution of the sensor. At the borders, the spatial resolution of the sensor is more important than its ability to discriminate among land-cover spectra. As a result, the two considered algorithms perform differently in border areas. We experimentally tested that the DAFE/MRF achieves better geometrical characterization of buildings and roads, while the neuro-fuzzy procedure tends to perform better in homogeneous areas (thus lacking precision at the border areas). On the other hand, the DAFE/MRF cannot reduce spurious region classifications as effectively. This is probably due to the fact that the DAFE/MRF technique was applied using a fixed 3×3 pixel window. We expect that an adaptive procedure based on using wider windows in homogeneous areas and smaller windows at the borders between regions may result in more accurate classification results.

4.3. Spatial/spectral endmember extraction

While our focus in previous subsections has been on classification approaches, spectral signatures in hyperspectral data are invariably a mixture of the signatures of the various materials found within the spatial extent of the ground instantaneous field view. Due to the high spectral dimensionality of the data, the number of spectral bands usually exceeds the number of spectral mixture components, and the unmixing problem is cast in terms of an over-determined system of equations in which, given a correct set of pure spectral signatures called “endmembers” (Adams et al., 1986). Since each observed spectral signal is the result of an actual mixing process, the driving abundances must obey two rather common-sense constraints. First, all abundances must be non-negative. Second, the sum of abundances for a given pixel must be unity (Chang, 2003). However, it is the derivation and validation of the correct suite of endmembers that has remained a challenging and elusive goal for the past 20 years. Several approaches have been developed over the last 30 years for solving the abundance estimation problem (Smith et al., 1985). Automatic techniques for endmember extraction include the pixel purity index (PPI) algorithm (Boardman, 1993), the N-FINDR software (Winter, 1999), or the iterative error analysis (IEA) algorithm (Neville et al., 1999). Although these methods have shown considerable promise, they are exclusively based on the spectral information of the data. In this work, we develop a new spatial/spectral endmember extraction approach which makes use of mathematical morphology concepts, introduced in previous sections, but with the goal of using the full (multi-band) spectral information to account for subtle spectral differences in the endmember searching process. To do so, we first define a cumulative distance between a particular pixel vector $\boldsymbol{f}(x,y)$, i.e., an N-dimensional vector at discrete spatial coordinates (x,y), and all the pixel vectors in the spatial neighborhood given by B (B-neighborhood) as follows:

$$D_B(\boldsymbol{f}(x,y)) = \sum_{(i,j)\in Z^2(B)} \mathrm{Dist}(\boldsymbol{f}(x,y),\boldsymbol{f}(i,j)) \tag{9}$$

where (i,j) are the spatial coordinates in the B-neighborhood discrete domain, represented by $Z^2(B)$, and *Dist* is a pointwise distance measure between two N-dimensional vectors. In this work, we use the spectral angle mapper (SAM) as the baseline distance metric for extending morphological operations. With the above definitions in mind, a description of the proposed endmember extraction algorithm is given below. The algorithm, based on the previously developed automated morphological endmember extraction (AMEE) algorithm (Plaza et al., 2002), is called I-AMEE to account for the new iterative nature of the proposed algorithm. The inputs to I-AMEE are the full hyperspectral data cube $\boldsymbol{f}$, a structuring element B (used to define the spatial context around each image pixel), a maximum number of algorithm iterations I_{max}, and a number of endmembers to be extracted, p. The output is an endmember set, $\{\mathbf{e}_i\}_{i=1}^{q}$, with $q \leq p$. The algorithm consists of the following steps:

(1) Set $i=1$ and initialize a morphological eccentricity index MEI $(x,y)=0$ for each pixel $\boldsymbol{f}(x,y)$.

(2) Move B through all the pixels of the input data, defining a local spatial search area around each pixel $\boldsymbol{f}(x,y)$, and calculate the maximum and minimum pixel vectors at each B-neighborhood using extended morphological erosion and dilation, respectively defined as follows:

$$(\boldsymbol{f}\ominus B)(x,y) = \mathrm{argmin}_{(i,j)\in Z^2(B)}\{D_B[\boldsymbol{f}(x+i,y+j)]\} \tag{10}$$

$$(\boldsymbol{f}\ominus B)(x,y) = \mathrm{argmax}_{(i,j)\in Z^2(B)}\{D_B[\boldsymbol{f}(x+i,y+j)]\} \tag{11}$$

(3) Update the MEI at each spatial location (x,y) using:

$$MEI(x,y) = \mathrm{Dist}[(\boldsymbol{f}\ominus B)(x,y),(\boldsymbol{f}\oplus B)(x,y)] \tag{12}$$

(4) Set $i=i+1$. If $i=I_{max}$, then go to step (5). Otherwise, replace the original image with its dilation using B using $\boldsymbol{f}=\boldsymbol{f}\oplus B$. 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).

(5) 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 SAM for all pixel vector pairs.

As shown by the algorithm description above, the I-AMEE is based on the selection of a set of “local” endmembers at each spatial neighborhood defined by the morphological structuring element. These endmembers are then used to define a MEI score which reflects the degree of spectral purity of signatures at local spatial neighborhoods defined around each image pixel. The pixels with maximum MEI scores are then used to obtain the global endmembers by avoiding endmember repetitions. Therefore, our proposed spatial/spectral endmember extraction method follows a local-to-global approach in the search of image endmembers.

To illustrate the performance of the I-AMEE method above, we have conducted an experimental assessment of endmember extraction algorithms using the well-known AVIRIS Cuprite data set. Four algorithms: PPI, N-FINDR, IEA and I-AMEE were considered for comparison purposes. 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. The parameter values selected are in agreement with those used before in previous studies (Plaza et al., 2004).

Table 9 tabulates the SAM scores obtained after comparing some selected USGS library spectra with the corresponding endmembers

Table 9
SAM spectral similarity scores among selected USGS mineral spectra and the endmembers produced by different algorithms.

	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
Pyrophilite	0.094	0.090	0.112	0.090	0.084	0.080

The numbers of iterations executed by the I-AMEE algorithm are shown in the parentheses.

extracted by the four considered algorithms. The smaller the SAM values across the ten minerals considered, the better the results. As shown in the table the SAM 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.

4.4. Hierarchical segmentation of hyperspectral images

This subsection outlines a segmentation algorithm that makes use of the concept of segmentation hierarchy, defined as a set of several segmentations of the same image at different levels of detail in which the segmentations at coarser levels can be produced from simple merges of regions at finer levels (Beaulieu &Goldberg, 1989).

The algorithm is based on Hierarchical step-wise optimization (HSWO), a form of region growing segmentation which can be summarized in three steps:

(1) Initialize the segmentation by assigning each image pixel a region label. If a pre-segmentation is provided, label each image pixel according to the pre-segmentation. Otherwise, label each image pixel as a separate region.
(2) Calculate the dissimilarity criterion value between all pairs of spatially adjacent regions, find the pair of spatially adjacent regions with the smallest dissimilarity criterion value, and merge that pair of regions.
(3) Stop if no more merges are required. Otherwise, return to step (2).

The hierarchical segmentation (HSEG) algorithm (Tilton, 1998) is an augmentation of HSWO, which differs from the latter in one major aspect. As opposed to HSWO, the HSEG algorithm allows for the merging of non-adjacent regions (controlled by the S_{wght} input parameter).For $S_{wght}=0.0$, HSEG is essentially the same as HSWO where only spatially adjacent are allowed to merge, while for $S_{wght}=1.0$, spatially adjacent and non-adjacent merges are given equal weight. Finally, values of S_{wght} between 0.0 and 1.0 favors spatially adjacent merges by a factor $1.0/S_{wght}$.

The current version of HSEG provides a selection of dissimilarity functions, including functions based on vector norms, mean-squared error, entropy, spectral information divergence (SID), spectral angle mapper (SAM), and normalized vector distance (NVD) citeptilton2006. Results utilizing a band summed mean squared error (BSMSE) dissimilarity function are reported later in this paper.

In order to overcome the heavy computational demands introduced by the standard HSEG algorithm, a recursive approximation (called RHSEG) has been developed (Tilton, 2005). This approach can be summarized by the following steps:

(1) Given an input hyperspectral image $\boldsymbol{f}$ with N spatial dimensions, specify the number of levels of recursion (L_r) required and pad the input image, if necessary, so that each spatial dimension of the data set can be evenly divided by $2^{(L_r-1)}$. (A good value for L_r results in an image section at recursive level L_r consisting of about 1000 to 4000 pixels.) Set $L=1$.
(2) Call rhseg(L,$\boldsymbol{f}$).
(3) Execute the HSEG algorithm on the image $\boldsymbol{f}$ using as a pre-segmentation the segmentation output by the call to rhseg() in step (2).

where rhseg(*level*,$\boldsymbol{f}$) is as follows:

(2.1) If $L=L_r$, go to step (2.3). Otherwise, divide the image data into a set of 2^N equal spatial-domain partitions and call rhseg() for each partition with $L=L+1$.
(2.2) After all 2^N calls to rhseg() from step (2.1) complete processing, reassemble the image segmentation results.
(2.3) If $L<L_r$, initialize the segmentation with the reassembled segmentation results from step (2.2). Otherwise, initialize the segmentation with 1 pixel per region. Then execute the HSEG algorithm on the image $\boldsymbol{f}$ so that the algorithm is terminated when the number of regions reaches a preset value N_{min}.
(2.4) If $L=L_r$, exit. Otherwise, perform a split and remerge process designed to blend the results together from the processing windows to avoid processing window artifacts (Tilton, 2005, 2006), and then exit.

To illustrate the use of RHSEG for hierarchical segmentation of hyperspectral data, we present a processing example based on subset #1 of the ROSIS urban data. Specifically, Table 10 reports processing results of subset #1 in which the region means were initially modeled as the means of the pixels labeled as a particular ground cover class in the ground-truth data. The BSMSE dissimilarity criterion is used here since it produced the best results for this data setin tests (not reported here). Table 10 shows that the best results were produce with $S_{wght}=0.1$, where spatial information is given 10 times priority over spectral information. In a separate experiment, we set $S_{wght}=0.1$ along with the BSMSE dissimilarity criterion and processed the data with RHSEG with no *a priori* information, i.e., RHSEG was initialized with each pixel as a separate region. In doing this evaluation, the coarsest hierarchical segmentation level that separated all (or most) ground cover classes was selected, and the region segments were assigned to a ground cover class if a plurality of the pixels in the region were covered by the ground truth for that ground cover class. The total number of regions obtained in this test was 47, with 33 of them covering ground-truth (pixels with no ground-truth designation were ignored). The overall accuracy was very high (96.5%), which is comparable to the case in which a priori knowledge was used to initialize the hierarchical segmentation process.

5. Parallel implementations

5.1. Parallel implementation of standard SVM classifiers

As stated in Section 3, a standard approach in kernel methods is to decompose the multiple class problem into multiple two-class problems. 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 $1,2,\ldots,i-1,i+1,\ldots S$ (Hsu & Lin, 2002).

A simple approach, which in hindsight is naive, would be to run in groups of K pairs in lockstep, distributed across the 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, our parallel algorithm uses a master–slave approach (i.e., one processor is used as the master and the rest are used as workers). The master 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 K-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.

Table 11 reports parallel performance results of the proposed parallel SVM approach on the Medusa cluster at NASA's Goddard

Table 10
Overall classification accuracy (percentage) obtained by RHSEG on subset #1 of the ROSIS urban data using BSMSE dissimilarity function and different values of the S_{wght} parameter.

$S_{wght}=1.0$	$S_{wght}=0.5$	$S_{wght}=0.1$
90.5 (9 regions)	96.5 (14 regions)	97.7 (18 regions)

Region means were initialized by the ground-truth data.

Table 11
Processing time in seconds on the Medusa cluster for training (finding the support vectors), loading the support vectors for prediction, prediction (applying the support vectors from every pair to hyperspectral data to get a pair prediction vote), classification (totaling the votes), total time, and speedup for various numbers of processors using the whole AVIRIS Indian Pines scene.

Number of CPUs	2	4	6	8	10	12	14	16	18	20	30	40	50
Time for training	104	57	50	35	35	34	34	35	35	35	43	53	56
Time load predict	20	21	25	15	15	16	16	17	19	20	29	37	45
Time predict	156	65	48	40	36	36	34	33	32	31	31	32	32
Time classify	1	1	<1	<1	1	<1	<1	<1	1	1	<1	1	1
Time total	281	144	123	90	87	86	84	85	87	87	103	123	134
Speedup	1.0	2.0	2.3	3.1	3.2	3.3	3.3	3.3	3.2	3.2	2.7	2.3	2.1

Table 12
Processing times (in seconds) for the parallel version of I-AMEE algorithm executed on Thunderhead using the AVIRIS Cuprite scene.

Number of CPUs	1	4	16	64	256
Time total	9465	4085	929	220	60
Speedup	1.0	2.3	10.1	43.1	157.7

Space Flight Center. Medusa is composed of 64 nodes of dual 1.2 GHz processors/node with 1 GB memory shared between the 2 processors, with a 2 GByte/s Myrinet for communication and peak performance of 370 GFlops. The operating system is Fedora Core. As shown by the table, the time for data movement between processors can be substantial. Specifically, the most significant times reported in the table correspond to the following operations:

(1) Moving all the training vectors from the master to every worker at the beginning of the training phase, which results in a constant time of around 19 s (included in the table "time for training" reported in the table).
(2) Moving all the support vectors and the hyperspectral cube from the master to each worker at the beginning of the prediction phase, which varies in time from 20–45 s (explicitly shown in the table).

As shown in Table 11, the speedup (performance gain with regards to using one processor) is not linear, but grows and then levels out at around 14 processors with a value of 3.3, and then declines for more than 16 processors. 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. The saturation at 3.3 is due to there being a wide distribution of processing times in the training phase which depends on the number of training vectors for each pair classification, and a wide distribution of processing times for the prediction phase depending on the number of support vectors.

5.2. Parallel implementation of morphological approaches

Morphological operations rely on a structuring element or 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 morphologically processed in parallel without the need for additional inter-processor communications (Plaza et al., 2006). Therefore, a PSSP may be seen as a chunk of data that can be processed independently, using smaller structuring elements. 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. In order to avoid inter-processor communication when the structuring element computation is split among processing nodes, a spatial-domain partitioning module has been implemented in the master so that the partitioning of the data into PSSPs makes use of a data-replication function, implemented to avoid accesses outside the local domain of each partition. 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 K spatial-domain partitions (with their scratch borders to avoid inter-processor communications), denoted by $\{\mathbf{PSSP}_i\}_{i=1}^{K}$, and distributes the partitions among the workers.
(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)–(5) of the sequential I-AMEE algorithm for its corresponding $\mathbf{PSSP}_i$, 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 SAM for all possible endmember pairs in parallel.

Table 12 shows the total time spent by our parallel algorithm using a case study where the algorithm performs $I_{max} = 5$ iterations, reported to produce the most accurate endmember extraction results in Table 9. Experiments were obtained on Thunderhead, a Beowulf cluster at NASA's Goddard Space Flight Center. The system is composed of 256 dual 2.4 GHz Intel Pentium 4 Xeon nodes, 256 GByte DDR memory (1.0 GByte of RAM available per CPU), 20 TByte disk space and a 2.2 GByte/s Myrinet fiber interconnection system. The peak performance is 2.5728 TFlops (see http://thunderhead.gsfc.nasa.gov for additional details), and the operating system used at the time of measurement was Linux Fedora Core. From the results addressed in Table 12, we can conclude that our parallel algorithm scaled very well, even for a large number of processors, resulting in processing times of about 1 min for the full AVIRIS Cuprite image when 256 processors were used.

5.3. Parallel implementation of hierarchical segmentation

A simple parallelization approach for the algorithm above would be to process each of the $2^{N(L_r-1)}$ spatial-domain partitions produced by RHSEG on a separate CPU. However, for larger images, the number of spatial-domain partitions can easily greatly exceed the number of available CPUs. In this case, the practical solution is to determine the number of recursive levels, $L_i \leq L_r$, that divide the data into a number of partitions less than (or equal to) the available number of CPUs (K) in the parallel system, so that $K \geq 2^{N(L_r-1)}$. Then, RHSEG can be run sequentially for the recursive levels above L_i. After the sequential recursive levels complete processing, and parallel processing is completed at recursive level L_i, one can leave the input data and pixel-based results data, such as the current region label map, at recursive level L_i. This pixel-based data can be retrieved or updated as needed from the parallel tasks running at recursive levels L_i by the parallel tasks running at recursive levels below that level. In the following, we report processing time performance (on the Thunderhead system) for the parallel version of the RHSEG segmentation algorithm. Table 13 compares the processing times achieved (for different numbers of processors) using $S_{wght} = 0.1$ and the BSMSE

Table 13
Processing times (in seconds) for the parallel version of RHSEG algorithm executed on Thunderhead using Subset #1 of the ROSIS urban data.

Number of CPUs	1	4	16	64	256
Time total	2061	568	155	48	25
Speedup	1.0	3.6	13.3	49.9	82.4

Table 14
Main characteristics of the processing techniques discussed in this work.

Processing technique	Algorithm nature	Spatial/spectral integration	Algorithm type	Data set used	Application area	Compared with
Standard SVM	Supervised	No	Classification	ROSIS subset#1	Urban classification	Different types of kernels
Transductive SVM	Semi-supervised	No	Classification	AVIRIS Indian Pines	Land-cover classification	Standard SVM
Contextual SVM	Supervised	Yes	Classification	AVIRIS Indian Pines	Land-cover classification	Standard classifiers
Morphological profiles	Supervised	Yes	Classification	ROSIS subset #2	Urban classification	Original spectral information
Markov random fields	Unsupervised	Yes	Classification	ROSIS subset #3	Urban classification	Neuro-fuzzy classifier
Multichannel morphology	Unsupervised	Yes	Endmember extraction	AVIRIS Cuprite	Mineral mapping	PPI, N-FINDR, IEA
Hierarchical segmentation	Unsupervised	Yes	Segmentation	ROSIS subset #1	Urban classification	Different baseline distances

dissimilarity criterion (which resulted in the best segmentation results for the considered subset #1 of the ROSIS data). As can be seen from Table 13, the proposed parallel implementation is particularly efficient for a reduced number of processors (16 and below), while the speedup figures flatten out a little for a high number of processors (see the case for 256 processors in the table). The high parallel efficiency of RHSEG measured for a small number of processors, as compared to the serial version, makes it clear that it is more efficient to swap the pixel-oriented intermediate results back and forth between parallel tasks than to swap this information in and out of disk files, as required by the serial version.

6. Summary of processing achievements and future potentials

This section summarizes the main processing achievements observed for the techniques described in this paper. We intend to provide a quick reference of the main characteristics of each discussed processing algorithm. For that purpose, Table 14 summarizes the main characteristics of each considered technique, including relevant aspects such as the nature of each algorithm, the techniques it was compared against, or the application domain in which was evaluated. Table 14 also addresses the hyperspectral data sets used for validation purposes in each case. It should be noted that ground-truth information for the considered hyperspectral scenes is available in different formats and, therefore, some of the processing techniques could only be evaluated with certain data sets, depending on the nature of ground-truth information available. In this regard, we carefully selected the hyperspectral data set which, in our opinion, better illustrated the performance of each processing technique. We believe that the compendium of data processing techniques and their detailed evaluation in the context of real applications presented in this paper may help image analysts and practitioners in this field in the task of selecting advanced data processing techniques and strategies for specific applications. On the other hand, Table 15 summarizes the main processing achievements observed from the experimental validation conducted for each method. The table also addresses the availability of efficient parallel implementations for some of the considered algorithms.

To conclude this section, we provide an outlook on the future potential of the methods discussed in this work. As demonstrated by our experimental results, hyperspectral data classification approaches are rapidly changing from *hard* classifiers to *soft* classifiers, such as different types of SVMs discussed in this paper. This is because these techniques seem 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 imaging spectroscopy), and with the limited availability of training samples in remote sensing applications. We anticipate that the full adaptation of soft classifiers to sub-pixel analysis (e.g., via multi-regression) may push the frontiers of imaging spectroscopy to new application domains.

Further approaches for 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 Markov random fields and morphological approaches, 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 be crucial to help increase algorithm efficiency and meet timeliness needs in many applications. As a result, the future potential of hyperspectral data processing methods such as those discussed in this work will also be largely defined by their suitability for being implemented in parallel. In this regard, we anticipate that joint spatial/spectral methods will be particularly appealing for efficient implementations due to the regularity of their computations, as demonstrated by the parallel versions developed in this work.

7. Conclusions

The introduction of the concept of imaging spectroscopy by A. F. H. Goetz and his colleagues established the foundations for the field known today as hyperspectral imaging, and has significantly influenced the evolution of remote sensing data processing techniques ever since. 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 image processing encompasses very different mathematical formalisms, some of them already exploited in other fields such as multispectral imaging.

Table 15
Summary of processing achievements by the techniques discussed in this work.

Processing technique	Main contribution with regards to other hyperspectral image processing techniques	Parallel version
Standard SVM	Reduced sensitivity to Hughes phenomenon	Yes
Transductive SVM	Better performance in the presence of limited training samples	No
Contextual SVM	Integration of spatial information in standard SVM	No
Morphological profiles	Improved classification by integration of spatial/spectral info	No
Markov random fields	Improved spatial characterization of spectral data	No
Multichannel morphology	Integration of spatial/spectral info in endmember extraction	Yes
Hierarchical segmentation	Improved segmentation by integration of spatial/spectral info	Yes

However, the special characteristics of hyperspectral images pose new processing problems, not to be found in other types of remotely sensed data:

(1) The high-dimensional nature of hyperspectral data introduces important limitations in supervised classifiers, such as the limited availability of training samples or the inherently complex structure of the data.
(2) 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 data processing.
(3) There is a need to develop cost-effective algorithm implementations, 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 image processing 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. The compendium of techniques presented in this work reflects the increasing sophistication of a field that is rapidly maturing at the intersection of many different disciplines.

Acknowledgements

The authors gratefully acknowledge Profs. Susan L. Ustin and Michael E. Schaepman for their kind invitation to present this work at the IGARSS special session on the *State of Science of Environmental Applications of Imaging Spectroscopy*, held in honor of Dr. Alexander F. H. Goetz, whose outstanding career and contributions to the field of spectroscopy have served as a guide and inspiration to the authors throughout their careers and during the development of this work. The authors also wish to acknowledge the teams supporting the NASA Goddard Space Flight Center's Thunderhead cluster and the University of Zurich's Matterhorn cluster whose support made our parallel implementations possible. This work has been supported by the European Community's Marie Curie Research Training Networks Programme under reference MRTN-CT-2006-035927, Hyperspectral Imaging Network (HYPER-I-NET). Funding from the Spanish Ministry of Science and Innovation (HYPERCOMP/EODIX project, reference AYA2008-05965-C04-02) is also gratefully acknowledged.

References

Adams, J. B., Smith, M. O., & Johnson, P. E. (1986). Spectral mixture modeling: A new analysis of rock and soil types at the Viking Lander 1 site. *Journal of Geophysical Research*, *91*, 8098−8112.

Baraldi, A., Binaghi, E., Blonda, P., Brivio, P. A., & Rampini, A. (2001). Comparison of the multilayer perceptron with neuro-fuzzy techniques in the estimation of cover class mixture in remotely sensed data. *IEEE Transactions on Geoscience and Remote Sensing*, *39*, 994−1005.

Beaulieu, J. M., & Goldberg, M. (1989). Hierarchy in picture segmentation: A stepwise optimal approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*, 150−163.

Benediktsson, J. A., Pesaresi, M., & Arnason, K. (2003). Classification and feature extraction for remote sensing images from urban areas based on morphological transformations. *IEEE Transactions on Geoscience and Remote Sensing*, *41*, 1940−1949.

Benediktsson, J. A., Palmason, J. A., & Sveinsson, J. R. (2005). Classification of hyperspectral data from urban areas based on extended morphological profiles. *IEEE Transactions on Geoscience and Remote Sensing*, *42*, 480−491.

Boardman, J. W. (1993). Automating spectral unmixing of AVIRIS data using convex geometry concepts, Summaries of Airborne Earth Science Workshop. *JPL Publication*, *93–26*, 11−14.

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifier. *Fifth ACM Annual Workshop on Computational Learning* (pp. 144−152).

Brazile, J., Schaepman, M. E., Schlapfer, D., Kaiser, J. W., Nieke, J., & Itten, K. I. (2003). Cluster versus grid for large-volume hyperspectral image preprocessing. *Proceedings of SPIE*, *5542*, 480−491.

Bruzzone, L., Chi, M., & Marconcini, M. (2006). A novel transductive SVM for the semisupervised classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *44*, 3363−3373.

Bruzzone, L., Chi, M., & Marconcini, M. (2007). Semisupervised support vector machines for classification of hyperspectral remote sensing images. In C. -I. Chang (Ed.), *Hyperspectral Data Exploitation: Theory and Applications* (pp. 275−311). : John Wiley and Sons, Inc. Chapter 11.

Chi, M., & Bruzzone, L. (2007). Semi-supervised classification of hyperspectral images by SVMs optimized in the primal. *IEEE Transactions on Geoscience and Remote Sensing*, *45*, 1870−1880.

Camps-Valls, G., Bandos, T., & Zhou, D. (2007). Semi-supervised graph-based hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, *45*, 3044−3054.

Camps-Valls, G., & Bruzzone, L. (2005). Kernel-based methods for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, *43*, 1351−1362.

Camps-Valls, G., Gomez-Chova, L., Muñoz-Mari, J., Vila-Frances, J., & Calpe-Maravilla, J. (2006). Composite kernels for hyperspectral image classification. *IEEE Geoscience and Remote Sensing Letters*, *3*, 93−97.

Chang, C. -I. (2003). *Hyperspectral imaging: techniques for spectral detection and classification.* New York: Kluwer.

Chanussot, J., Benediktsson, J. A., & Fauvel, M. (2006). Classification of remote sensing images from urban areas using a fuzzy probabilistic model. *IEEE Geoscience and Remote Sensing Letters*, *3*, 40−44.

Chen, Y., Wang, G., & Dong, S. (2003). Learning with progressive transductive support vector machines. *Pattern Recognition Letters*, *24*, 1845−1855.

Chellappa, R., & Jain, A. (1993). *Markov random fields: theory and applications.* New York: Academic.

Clark, R. N., Swayze, G. A., Gallagher, A., King, T. V., & Calvin, W. M. (1993). The U.S. Geological Survey digital spectral library, version 1: 0.2 to 3.0 mm. *U.S. Geological Survey, Open File Report 93–592* Available online: http://speclab.cr.usgs.gov

Cortes, C., & Vapkik, V. N. (1995). Support vector networks. *Machine Learning*, *20*, 1−25.

Fauvel, M., Chanussot, J., & Benediktsson, J. A. (2006). Decision fusion for the classification of urban remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, *44*, 2828−2838.

Fauvel, M., Chanussot, J., & Benediktsson, J. A. (2006). A combined support vector machines classification based on decision fusion. *Proc. IEEE Intl. Geoscience and Remote Sensing Symposium* (pp. 2494−2497).

Foody, G. M. (2002). Status of land cover classification accuracy assessment. *Remote Sensing of Environment*, *90*, 185−201.

Foody, G. M., & Arora, M. (1996). Incorporating mixed pixels in the training, allocation and testing stages of supervised classifications. *Pattern Recognition Letters*, *17*, 1389−1398.

Foody, G. M., & Mathur, A. (2004). Toward intelligent training of supervised image classifications: Directing training data acquisition for SVM classification. *Remote Sensing of Environment*, *93*, 107−117.

Gamba, P., Dell'Acqua, F., Ferrari, A., Palmason, J. A., Benediktsson, J. A., & Arnasson, J. (2004). Exploiting spectral and spatial information in hyperspectral urban data with high resolution. *IEEE Geoscience and Remote Sensing Letters*, *1*, 322−326.

Gamba, P., Trianni, G. (2005). A novel MRF model for multisource data fusion in urban areas, Proc. of URSI General Assembly, New Delhi (India), Oct. 2005, unformatted CD-ROM.

Goetz, A. F. H., Vane, G., Solomon, J. E., & Rock, B. N. (1985). Imaging spectrometry for Earth remote sensing. *Science*, *228*, 1147−1153.

Green, R. O., Eastwood, M. L., Sarture, C. M., Chrien, T. G., Aronsson, M., Chippendale, B. J., et al. (1998). Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS). *Remote Sensing of Environment*, *65*, 227−248.

Gualtieri, J. A., Chettri, S. R., Cromp, R. F., & Johnson, L. F. (1999). Support vector machines applied to AVIRIS data. *Summaries of the Airborne Earth Science Workshop*.

Gualtieri, J. A., & Cromp, R. F. (1998). Support vector machines for hyperspectral remote sensing classification. *Proceedings of SPIE*, *3584*, 221−232.

Hsu, C. W., & Lin, C. J. (2002). A comparison of methods for multiclass support vector machines. *IEEE Transactions on Neural Networks*, *13*, 415−425.

Hughes, G. F. (1968). On the mean accuracy of statistical pattern recognizers. *IEEE Transactions on Information Theory*, *14*, 55−63.

Jimenez, L. O., & Landgrebe, D. A. (1998). Supervised classification in high-dimensional space: Geometrical, statistical and asymptotical properties of multivariate data. *IEEE Transactions Systems, Man and Cybernetics*, *28*, 39−54.

Jimenez, L. O., Rivera-Medina, J. L., Rodriguez-Diaz, E., Arzuaga-Cruz, E., & Ramirez-Velez, M. (2005). Integration of spatial and spectral information by means of unsupervised extraction and classification for homogenous objects applied to multispectral and hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, *43*, 844−851.

Jia, X., Richards, J. A., & Ricken, D. E. (1999). *Remote sensing digital image analysis: an introduction.* Berlin: Springer-Verlag.

Joelsson, S. R., Sveinsson, J. R., & Benediktsson, J. A. (2006). Feature selection for morphological feature extraction using random forests. *Proc. 7th IEEE Nordic Signal Processing Symposium* (pp. 138−141).

Kasetkasem, T., Arora, M. K., & Varshney, P. K. (2005). Super-resolution land cover mapping using a Markov random field based approach.*Remote Sensing of Environment*, *96*, 302−310 2005.

Keerthi, S. S., Shevade, S. K., Bhattacharyya, C., & Murthy, K. R. K. (1999). *Improvements to Platt's SMO Algorithm for SVM classifier design.* Available online: http://guppy.mpe.nus.edu.sg/mpessk/smo_mod.ps.gz

Keshava, N., & Mustard, J. F. (2002). Spectral unmixing. *IEEE Signal Processing Magazine*, *19*, 44−57.

King, R. L. (2003). Putting information into the service of decision making: the role of remote sensing analysis. *IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data* (pp. 25−29).

Landgrebe, D. A. (2003). *Signal theory methods in multispectral remote sensing.* Hoboken: John Wiley and Sons.

Mercier, G., & Lennon, M. (2003). Support vector machines for hyperspectral image classification with spectral-based kernels. *IEEE Geoscience & Remote Sensing Symposium*, *1*, 288−290.

Muller, K. R., Mika, S., Ratsch, G., Tsuda, K., & Scholkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, *12*, 181−202.

Neville, R. A., Staenz, K., Szeredi, T., Lefebvre, J., & Hauff, P. (1999). Automatic endmember extraction from hyperspectral data for mineral exploration. *Canadian Symposium on Remote Sensing* (pp. 21−24).

Platt, J. (1999). Fast training of support vector machines using sequential minimal optimization. *Advances in kernel methods: support vector learning* Cambridge: MIT Press.

Plaza, A., Martinez, P., Perez, R., & Plaza, J. (2002). Spatial/spectral endmember extraction by multi-dimensional morphological operations. *IEEE Transactions on Geoscience and Remote Sensing*, *40*, 2025−2041.

Plaza, A., Martinez, P., Perez, R., & Plaza, J. (2004). A quantitative and comparative analysis of endmember extraction algorithms from hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, *42*, 650−663.

Plaza, A., Valencia, D., Plaza, J., & Martinez, P. (2006). Commodity cluster-based parallel processing of hyperspectral imagery. *Journal of Parallel and Distributed Computing*, *66*, 345−358.

Richards, J. A. (2005). Analysis of remotely sensed data: The formative decades and the future. *IEEE Transactions on Geoscience and Remote Sensing*, *43*, 422−432.

Scholkopf, B., & Smola, J. (2002). *Learning with kernels.* : MIT Press.

Serra, J. (1982). *Image analysis and mathematical morphology.* New York: Academic.

Smith, M. O., Johnson, P. E., & Adams, J. B. (1985). Quantitative determination of mineral types and abundances from reflectance spectra using principal components analysis. *Journal of Geophysical Research*, *80*, 797−804.

Smith, M. O., Ustin, S. L., Adams, J. B., & Gillespie, A. R. (1990). Vegetation in deserts: I. A regional measure of abundance from multispectral images. *Remote Sensing of Environment*, *31*, 1−26.

Smith, M. O., Ustin, S. L., Adams, J. B., & Gillespie, A. R. (1990). Vegetation in deserts: II. Environmental influences on regional abundance. *Remote Sensing of Environment*, *31*, 27−52.

Soille, P. (2003). *Morphological image analysis: principles and applications.* Germany: Springer-Verlag.

Tadjudin, S., & Landgrebe, D. (1998). *Classification of high dimensional data with limited training samples.* Available online: http://dynamo.ecn.purdue.edu/ landgreb/Saldju_TR.pdf

Tilton, J. C. (1998). Image segmentation by region growing and spectral clustering with a natural convergence criterion. *IEEE Geoscience & Remote Sensing Symposium*, *4*, 1766−1768.

Tilton, J. C. (2005). A split-remerge method for eliminating processing window artifacts in recursive hierarchical segmentation. *Disclosure of Invention and New Technology: NASA Case No. GSC 14,994-1, March 9, 2005*.

Tilton, J. C. (2006). *RHSEG and HSEGViewer user's Manual, Provided with the demonstration version of RHSEG.* available from: http://ipp.gsfc.nasa.gov/RHSEG (version 1.25 released Dec. 14, 2006).

Tilton, J. C., Lawrence, W. J., & Plaza, A. (2006). Utilizing hierarchical segmentation to generate water and snow masks to facilitate monitoring change with remotely sensed image data. *GIScience & Remote Sensing*, *43*, 39−66.

Varshney, P. K., & Arora, M. K. (Eds.). (2004). *Advanced Image Processing Techniques for Remotely Sensed Hyperspectral Data* : Springer Verlag.

Watanachaturaporn, P., Arora, M. K., & Varshney, P. K. (2006). Hyperspectral image classification using support vector machines: A comparison with decision tree and neural network classifiers. *American Society for Photogrammetry & Remote Sensing (ASPRS) 2005 Annual Conference, Reno, NV*.

Winter, M. E. (1999). N-FINDR: Algorithm for fast autonomous spectral endmember determination in hyperspectral data. *Proceedings of SPIE*, *3753*, 266−277.