Clustering of Hyperspectral Images Based on Multiobjective Particle Swarm Optimization

Andrea Paoli, Farid Melgani, *Senior Member, IEEE*, and Edoardo Pasolli, *Student Member, IEEE*

***Abstract*—In this paper, we present a new methodology for clustering hyperspectral images. It aims at simultaneously solving the following three different issues: 1) estimation of the class statistical parameters; 2) detection of the best discriminative bands without requiring the *a priori* setting of their number by the user; and 3) estimation of the number of data classes characterizing the considered image. It is formulated within a multiobjective particle swarm optimization (MOPSO) framework and is guided by three different optimization criteria, which are the log-likelihood function, the Bhattacharyya statistical distance between classes, and the minimum description length (MDL). A detailed experimental analysis was conducted on both simulated and real hyperspectral images. In general, the obtained results show that interesting classification performances can be achieved by the proposed methodology despite its completely unsupervised nature.**

***Index Terms*—Feature selection, hyperspectral images, image clustering, k-means algorithm, multiobjective (MO) optimization, particle swarm optimization (PSO).**

I. Introduction

FROM a methodological viewpoint, a classification process consists of associating a pattern (sample) to a class label opportunely chosen from a predefined set of class labels. In the literature, two main approaches to the classification problem have been proposed: 1) the supervised approach and 2) the unsupervised approach. Supervised techniques require the availability of a training set for learning the classifier. Unsupervised methods, known also as clustering methods, perform classification just by exploiting information conveyed by the data, without requiring any training sample set. The supervised methods offer a higher classification accuracy compared to the unsupervised ones, but in some applications, it is necessary to resort to unsupervised techniques because training information is not available.

The last years have seen the publication of some works dealing with clustering problems in remote sensing applications. An early method is the one presented in [1], which exploits the notions of scale and cluster independence to classify multispectral and polarimetric synthetic aperture radar (SAR) images. Palubinskas *et al.* [2] introduced the concept of a global classification of remote sensing images in large archives, e.g., covering the whole globe. The classification is realized through a two-step procedure: 1) unsupervised clustering and 2) supervised hierarchical classification. Features, derived from different and noncommensurable models, are combined using an extended k-means clustering algorithm and supervised hierarchical Bayesian networks incorporating any available prior information. In [3], a fuzzy clustering method for multispectral images was presented. It groups data samples, even when the number of clusters is not known or when noise is present, by replacing the probabilistic constraint that memberships across clusters must sum to one with a composite constraint. In [4], the hybrid supervised–unsupervised approach to image classification was improved by introducing the concept of cluster-space classification for hyperspectral data. The cluster-space representation is used for associating spectral clusters with corresponding information classes automatically, thus overcoming the manual assignment of clusters and classes carried out in the hybrid approach. This method is further enhanced for efficient data transmission and classification in [5]. In [6], a method of hyperspectral band reduction based on rough sets and fuzzy C-means clustering was proposed. It consists of two steps. First, the fuzzy C-means clustering algorithm is used to classify the original bands into equivalent band groups. Then, data dimensionality is reduced by selecting only the band with the maximum grade of fuzzy membership from each of the groups. In [7], limitations of k-means algorithm implementation were discussed. In order to accelerate the k-means clustering, a hardware implementation was proposed. Lee and Crawford [8] presented a two-stage hierarchical clustering technique for classifying hyperspectral data. First, a "local" segmentator performs region-growing segmentation by merging spatially adjacent clusters. Then, a "global" segmentator clusters the segments resulting from the previous stage using an agglomerative hierarchical clustering scheme based on a context-free similarity measure. In [9], five clustering techniques were compared for classifying polarimetric SAR images. Two techniques are fuzzy clustering algorithms based on the standard l_1 and l_2 metrics. Two others combine a robust fuzzy C-means clustering technique with a distance measure based on the Wishart distribution. The fifth technique is an application of the expectation–maximization (EM) algorithm, assuming that data follow a Wishart distribution. Marcal and Castro [10] proposed an agglomerative hierarchical clustering method for multispectral images, which uses both spectral and spatial information for the aggregation decision. In [11], Markov random field (MRF) clustering, exploiting both spectral and spatial interpixel dependency information, for polarimetric SAR images was presented. Because of its strong sensitivity to initial conditions, an initialization scheme was suggested. It aims at deriving initial cluster parameters from a set of homogenous regions and

Manuscript received December 17, 2008; revised March 13, 2009. First published August 25, 2009; current version published November 25, 2009. This research was carried out within the framework of a project entitled "Development of Advanced Automatic Analysis Methodologies for Environmental, Industrial and Biomedical Monitoring," funded by the Italian Ministry of Education, University, and Research (MIUR).

The authors are with the Department of Information Engineering and Computer Science, University of Trento, 38100 Trento, Italy (e-mail: carbuner@yahoo.it; melgani@disi.unitn.it; pasolli@disi.unitn.it).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2009.2023666

at estimating the number of clusters with the pseudolikelihood information criterion. In [12], a two-step unsupervised artificial immune classifier for multispectral/hyperspectral images was proposed. In [13], unsupervised land cover classification is performed by clustering pixels in the spectral domain into several fuzzy partitions. A multiobjective (MO) optimization algorithm is utilized to tackle the fuzzy partitioning problem by means of a simultaneous optimization of different fuzzy cluster validity indexes. The resulting near-Pareto-optimal front contains a set of nondominated solutions, from which the user can pick up the most promising one according to the problem requirements. In [14], a weighted fuzzy C-means clustering algorithm was proposed to carry out the fuzzy or the hard classification of multispectral images. Xia *et al.* [15] presented a rapid clustering method for SAR images by embedding an MRF model in the clustering space and by using graph cuts to search for data clusters optimal in the sense of the maximum *a posteriori* (MAP) criterion. In [16], a multistage unsupervised classification technique for multispectral images was presented. It is composed of a context-sensitive initialization and an iterative procedure aiming at estimating the statistical parameters of classes to be used in a Bayesian decision rule. The initial steps exploit a graph cut segmentation algorithm followed by fuzzy C-means clustering, while the iterative procedure is based on the EM algorithm. In [17], unsupervised classification of hyperspectral images was performed by applying fuzzy C-means clustering as well as its extended version, i.e., Gustafson–Kessel clustering, which is based on an adaptive distance norm. An opportune phase-correlation-based similarity measure was used to improve the fuzzy clustering by taking spatial relations into account for pixels with similar spectral characteristics.

In this paper, we focus the attention on hyperspectral image clustering. Compared with conventional multispectral data, hyperspectral data are characterized by a higher spectral resolution, thus giving the opportunity to push further the information extraction capability. However, hyperspectral imagery involves a greater quantity of data to memorize and to process. Moreover, given a specific classification problem, hyperspectral data often exhibit redundant information, thus calling for opportune band (feature) selection algorithms. While feature selection has been widely studied in the supervised classification context, little has been done in the image clustering context due to the lack of training samples. Another intrinsic problem in image clustering in general and in hyperspectral image clustering in particular is how to set *a priori* the number of data classes because of the absence of prior information.

The objective of this paper is to propose a novel methodology for hyperspectral images capable of simultaneously solving the above problems, i.e., clustering, feature detection (i.e., selection of the features without requiring the desired number of most discriminative features *a priori* from the user), and class number estimation. Clustering and feature detection are dealt with within an MO optimization process based on particle swarm optimization (PSO) to estimate the cluster statistical parameters and to detect the most discriminative features. Class number estimation is performed using a strategy based on the minimum description length (MDL) criterion. To illustrate the performance of the proposed methodology, we conducted an experimental study based on simulated and real hyperspectral images. In particular, images acquired with the Airborne Visible InfraRed Imaging Spectrometer (AVIRIS) and the Reflective Optics System Imaging Spectrometer (ROSIS) sensors were considered. In general, the obtained experimental results show that interesting performances can be achieved though the processing context is completely unsupervised.

The remaining part of this paper is organized as follows. The problem formulation is described in Section II. The PSO fundamentals are recalled in Section III. The proposed clustering methodology is described in Section IV. The experimental results are reported in Sections V and VI. Finally, conclusions are drawn in Section VII.

II. Problem Formulation

A. Hypotheses

Let us consider a hyperspectral image composed of d bands and n pixels. Each pixel is represented by a vector $\mathbf{x_i} \in \Re^d = [x_{i,1}, x_{i,2}, \ldots, x_{i,d}],\ i = 1, 2, \ldots, n$. Let us assume that no prior knowledge is available for this image in terms of training samples. Moreover, let us suppose that the number C of data classes present in the image is not known. In optical imagery, the assumption that the distribution of images can be approximated as a mixture of normally distributed samples is generally well accepted [18], [19]. Accordingly, the probability distribution function (pdf) of the image can be written as

$$p(\mathbf{x}) = \sum_{j=1}^{C} P(\omega_j) \cdot p(\mathbf{x}|\omega_j) \tag{1}$$

where $P(\omega_j)$ and $p(\mathbf{x}|\omega_j) = N(\mu_j, \Sigma_j)$ are the prior probability and the conditional pdf associated with the jth data class (Gaussian mode) of the image, respectively. μ_j and Σ_j stand for the mean vector and the covariance matrix of the jth data class. Let us suppose that the features (bands) are independent and, hence, Σ_j is a diagonal matrix. We have

$$\Sigma_j = \begin{bmatrix} \sigma_{1j}^2 & 0 & \cdots & \cdots & 0 \\ 0 & \sigma_{2j}^2 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \sigma_{dj}^2 \end{bmatrix}. \tag{2}$$

We note that although the assumption of independence between adjacent bands is typically not satisfied, it is important to render our clustering problem computationally tractable.

B. Goal

The objective is to classify the image in an unsupervised way. Given its hyperspectral nature, it is preferable to perform beforehand a feature detection operation. Indeed, this last is useful to reduce data redundancy and to remove the bands that are characterized by a strong noise component. Note that by feature detection, we intend that the selection of the features be carried out without requiring the desired number of most discriminative features *a priori* from the user. A further goal is to automatically estimate the number of data classes characterizing the image.

We desire to meet all these requirements simultaneously, without any prior knowledge about the investigated study area.

C. Proposed Solution

The idea proposed in this work is to formulate this complex problem within a multiobjective particle swarm optimization (MOPSO) framework to simultaneously estimate the cluster statistical parameters, detect the most discriminative features, and estimate the class number. The MO approach to the problem is motivated by the different nature of the desired tasks. In particular, the first PSO fitness function will have the purpose of estimating the cluster parameters, while the second one will guide the detection of the best features and, thus, the removal of redundant and/or noisy bands. The class number estimation will be carried out by repeating the PSO process over a predefined range of values of class number for optimizing the MDL criterion.

III. PSO

A. Basic Principles of PSO

PSO is a stochastic optimization technique recently introduced by Kennedy and Eberhart, which is inspired by social behavior of bird flocking and fish schooling [20], [21]. Similar to other evolutionary computation algorithms such as genetic algorithms [22], [23], PSO is a population-based search method, which exploits the concept of social sharing of information. This means that each individual (called *particle*) of a given population (called *swarm*) can profit from the previous experiences of all other individuals from the same population. During the search process in the solution space, each particle (i.e., candidate solution) will adjust its flying velocity and position according to its own flying experience as well as the experiences of the other companion particles of the swarm. PSO has been shown to be promising for solving various engineering problems such as automatic control [24], antenna design [25], inverse problems [26], and ECG signal classification [27]. In the following, we will briefly describe the main concepts of the basic PSO algorithm.

Let us consider a swarm of size S. Each particle P_i, $i = 1, 2, \ldots, S$, from the swarm is characterized by the following: 1) its current position $\mathbf{p}_i(t) \in \Re^q$, which refers to a candidate solution of the optimization problem at iteration t; 2) its velocity $\mathbf{v}_i(t) \in \Re^q$; and 3) the best position $\mathbf{p}_{bi}(t) \in \Re^q$ identified during its past trajectory. Let $\mathbf{p}_g(t) \in \Re^q$ be the best global position found over all trajectories traveled by the particles of the swarm. The position optimality is measured by means of one or more fitness functions defined in relation to the considered optimization problem. During the search process, the particles move according to the following equations:

$$\begin{cases} \mathbf{v}_i(t+1) = w\mathbf{v}_i(t) + c_1 \cdot r_1(t)\left(\mathbf{p}_{bi}(t) - \mathbf{p}_i(t)\right) \\ \qquad\qquad + c_2 \cdot r_2(t)\left(\mathbf{p}_g(t) - \mathbf{p}_i(t)\right) & (3) \\ \mathbf{p}_i(t+1) = \mathbf{p}_i(t) + \mathbf{v}_i(t) & (4) \end{cases}$$

where $r_1(\cdot)$ and $r_2(\cdot)$ are random variables drawn from a uniform distribution in the range [0, 1] to provide stochastic weighting of the different components participating in the particle velocity definition. c_1 and c_2 are two acceleration constants that regulate the relative velocities with respect to the best global and local positions, respectively. The inertia weight w is used as a tradeoff between global and local exploration capabilities of the swarm. Large values of this parameter permit better global exploration, whereas small values lead to a fine search in the solution space. Equation (3) allows the computation of the velocity at iteration $t+1$ for each particle in the swarm by combining linearly its current velocity (at iteration t) and the distances that separate the current particle position from its best previous position and the best global position, respectively. The updating of the particle position is performed with (4). Both (3) and (4) are iterated until convergence of the search process is reached. Typical convergence criteria are based on the iterative behavior of the best value of the adopted fitness function(s) or/and simply on a user-defined maximum number of iterations.

B. MO Optimization

Usually, real-world applications require multiple measures of competing objectives to be simultaneously optimized. Optimizing multiple objectives involves finding a set of optimal solutions rather than a single one. The selection of a solution from this set is not trivial and is usually user-dependent. From a mathematical viewpoint, a general MO optimization problem can be formulated as follows.

Find the vector $\mathbf{p}^* = [p_1^*, p_2^*, \ldots, p_q^*]$ that minimizes the ensemble of M objective functions, i.e.,

$$\mathbf{f}(\mathbf{p}) = [f_i(\mathbf{p}), \qquad i = 1, \ldots, M] \tag{5}$$

subject to the J equality constraints, i.e.,

$$\mathbf{g}_j(\mathbf{p}) = 0, \qquad j = 1, 2, \ldots, J \tag{6}$$

and the K inequality constraints, i.e.,

$$\mathbf{h}_k(\mathbf{p}) \leq 0, \qquad k = 1, 2, \ldots, K \tag{7}$$

where $\mathbf{p}$ is a candidate solution to the considered optimization problem. In a PSO framework, it refers to the particle position in the solution space.

The solving of an MO optimization problem relies on an important concept, which is that of domination. A solution $\mathbf{p}_i = [p_{i1}, p_{i2}, \ldots, p_{iq}]$ is said to dominate another solution $\mathbf{p}_j = [p_{j1}, p_{j2}, \ldots, p_{jq}]$ if and only if $\mathbf{f}(\mathbf{p}_i)$ is partially less than $\mathbf{f}(\mathbf{p}_j)$, i.e.,

$$\begin{aligned} &\forall k \in \{1, 2, \ldots, M\},\ f_k(\mathbf{p}_i) \leq f_k(\mathbf{p}_j) \\ &\qquad \wedge\ \exists k \in \{1, 2, \ldots, M\} : f_k(\mathbf{p}_i) < f_k(\mathbf{p}_j). \end{aligned} \tag{8}$$

This concept leads to the definition of *Pareto optimality*: a solution $\mathbf{p_i^*} \in \Omega$ (Ω is the solution space) is said to be *Pareto optimal* if and only if there exists no other solution $\mathbf{p_j} \in \Omega$ that dominates $\mathbf{p_i^*}$. The latter is said to be *nondominated*, and the set of all nondominated solutions forms the so-called *Pareto front* of optimal solutions.

To find this front, several PSO-based approaches have been proposed in the literature [28], [29]. PSO are particularly suitable for dealing with MO optimization problems because of their ability to move uniformly toward the true Pareto optimal solutions.

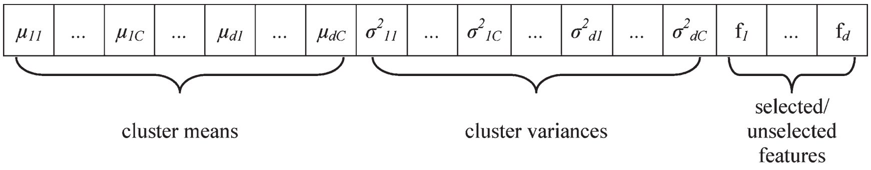


Fig. 1. PSO particle structure.

IV. Proposed Clustering Approach

A. PSO Setup

As mentioned previously, in an optimization problem that is formulated within a PSO framework, the solution space is explored by means of a swarm of particles whose positions point to candidate solutions. The first task to perform consists of defining the ingredients of the PSO algorithm, namely, the particle position $\mathbf{p}$ and the fitness functions $\mathbf{f}(\mathbf{p})$.

Since our clustering problem consists of finding the best estimate of the cluster statistical parameters and the best discriminative features, the position $\mathbf{p}$ of each particle will simply be a vector that encodes all these variables. A representation of the particle position is provided in Fig. 1. Given C data classes, the cluster parameters are defined by a number of real variables equal to $2 \times C \times d$ since for each class ω_j $(j = 1, \ldots, C)$, the mean vector $\boldsymbol{\mu}_j \in \Re^d = [\mu_{1j}, \mu_{2j}, \ldots, \mu_{dj}]$ and the variance vector $\boldsymbol{\sigma}_j^2 \in \Re^d = [\sigma_{1j}^2, \sigma_{2j}^2, \ldots, \sigma_{dj}^2]$ must be estimated. Moreover, the feature detection task requires the setting of d coordinates expressed in terms of Boolean values. The feature variable f_i, $i = 1, \ldots, d$, is equal to 1 if the correspondent feature is selected; otherwise, it is equal to 0.

Another important aspect in the setup of a PSO process is the choice of the fitness functions $\mathbf{f}(\mathbf{p})$, which will be used to evaluate the performance of each position $\mathbf{p}$. For this purpose, we propose to jointly optimize two different criteria to deal with both the class statistical parameter estimation and the feature detection issues.

The first fitness function is the log-likelihood function. It aims at determining the distribution parameter values that best approximate the data distribution. Supposing that samples are independent identically distributed, the likelihood function is given by

$$l(\mathbf{x}|\mathbf{p}, \mathbf{P}_c) = \prod_{i=1}^{n} p(\mathbf{x}_i|\mathbf{p}, \mathbf{P}_c) \tag{9}$$

where $\mathbf{P}_c = [P(\omega_1), P(\omega_2), \ldots, P(\omega_C)]$ is the prior probability vector.

Accordingly, the log-likelihood function becomes

$$L(\mathbf{x}|\mathbf{p}, \mathbf{P}_c) = \ln l(\mathbf{x}|\mathbf{p}, \mathbf{P}_c) = \sum_{i=1}^{n} \ln p(\mathbf{x}_i|\mathbf{p}, \mathbf{P}_c). \tag{10}$$

Since the samples come from a multivariate Gaussian distribution, the log-likelihood function can be rewritten as

$$L(\mathbf{x}|\mathbf{p}, \mathbf{P}_c) = \sum_{i=1}^{n} \ln \sum_{j=1}^{C} \frac{P(\omega_j)}{(2\pi)^{\tilde{d}(\mathbf{p})/2} \cdot \left|\tilde{\Sigma}_j(\mathbf{p})\right|^{1/2}} \times \exp\left\{-\frac{1}{2}\left(\tilde{\mathbf{x}}_i(\mathbf{p}) - \tilde{\boldsymbol{\mu}}_j(\mathbf{p})\right)^T \tilde{\Sigma}_j(\mathbf{p})^{-1}\left(\tilde{\mathbf{x}}_i(\mathbf{p}) - \tilde{\boldsymbol{\mu}}_j(\mathbf{p})\right)\right\} \tag{11}$$

where $\tilde{d}(\mathbf{p})$ is the number of features detected by $\mathbf{p}$, $\tilde{\mathbf{x}}_i(\mathbf{p})$ is the ith sample defined in the subspace formed by the detected features, and $\tilde{\boldsymbol{\mu}}_j(\mathbf{p})$ and $\tilde{\Sigma}_j(\mathbf{p})$ are the mean vector and the diagonal covariance matrix associated with the jth class in that subspace.

The second fitness function has the purpose of evaluating the statistical distance between classes in the subspace of detected features defined by $\mathbf{p}$. For such a purpose, in this paper, we will adopt the Bhattacharyya distance [30], which, for a couple of classes normally distributed (e.g., the ith and jth classes), is expressed as follows:

$$B_{i,j}(\mathbf{p}) = \frac{1}{8}\left(\tilde{\boldsymbol{\mu}}_i(\mathbf{p}) - \tilde{\boldsymbol{\mu}}_j(\mathbf{p})\right)^T \left\{\frac{\tilde{\Sigma}_i(\mathbf{p}) + \tilde{\Sigma}_j(\mathbf{p})}{2}\right\}^{-1} \times \left(\tilde{\boldsymbol{\mu}}_i(\mathbf{p}) - \tilde{\boldsymbol{\mu}}_j(\mathbf{p})\right) + \frac{1}{2}\ln\left\{\frac{\left|\frac{\tilde{\Sigma}_i(\mathbf{p}) + \tilde{\Sigma}_j(\mathbf{p})}{2}\right|}{\left|\tilde{\Sigma}_i(\mathbf{p})\right|^{\frac{1}{2}}\left|\tilde{\Sigma}_j(\mathbf{p})\right|^{\frac{1}{2}}}\right\}. \tag{12}$$

At this point, the multiclass distance can be determined using different strategies. In this paper, it will be calculated according to the following simple rule:

$$B(\mathbf{p}) = \min_{i=\{1,\ldots,C\}, j=\{1,\ldots,C\}, i \neq j} \left\{B_{i,j}(\mathbf{p})\right\}. \tag{13}$$

Since the particles of the swarm can define feature subspaces of different dimensionalities, in order to remove (reduce) the impact of the dimensionality on the fitness function values, the previously defined fitness functions are normalized with respect to the number of features. Therefore, the first fitness function becomes

$$L_{\text{nor}}(\mathbf{x}|\mathbf{p}, \mathbf{P}_c) = \frac{L(\mathbf{x}|\mathbf{p}, \mathbf{P}_c)}{d(\mathbf{p})}. \tag{14}$$

Similarly, the second fitness function takes the following form:

$$B_{\text{nor}}(\mathbf{p}) = \frac{B(\mathbf{p})}{d(\mathbf{p})}. \tag{15}$$

Moreover, because the two fitness functions above need to be maximized for best clustering performance, they will be rewritten in such a way that the maximization problem is converted into a minimization one, i.e.,

$$f_1(\mathbf{p}, \mathbf{P}_c) = \left|L_{\text{nor}}(\mathbf{x}|\mathbf{p}, \mathbf{P}_c)\right| \tag{16}$$

$$f_2(\mathbf{p}) = \frac{1}{B_{\text{nor}}(\mathbf{p})}. \tag{17}$$

The last issue to face is the estimation of the number $\hat{C}$ of data classes representing the observed data since it is not known *a priori*. We have to resort to a technique that deals with this important issue, which is typical of mixture modeling problems. Indeed, the selection of the number of components in a mixture raises a tricky tradeoff, since on the one hand, the higher the number of components is, the higher the risk of data overfitting becomes, while on the other, the smaller the number of components is, the lower the model flexibility will be. In the literature, the most popular methods for automatically estimating the number of data classes are based on approximate Bayesian criteria or on information theory concepts [31]. In this paper, we will use the MDL criterion, which takes origin from the information theory and is defined for a given number of classes, e.g., C classes, as [32]

$$\mathrm{MDL}(C) = -L_{\mathrm{nor}}(C) + \gamma \cdot K(C) \cdot \log(n) \tag{18}$$

where $L_{\mathrm{nor}}(C)$ represents the normalized log-likelihood function value found at the convergence of the PSO algorithm, $K(C)$ is the number of estimated statistical parameters, and γ is a constant. For the setting of γ, different values are proposed in the literature. According to [33], $\gamma = 5/2$ seems the most appropriate choice. The optimal number of data classes $\hat{C}$ is estimated by minimizing the MDL criterion, i.e.,

$$\hat{C} = \operatorname*{arg\,min}_{C=C_{\min},\ldots,C_{\max}} \{MDL(C)\} \tag{19}$$

where $C_{\min}$ and $C_{\max}$ are predefined minimal and maximal numbers of data classes.

B. Handling Priors

Another issue to solve is how to estimate the prior probability of each class since poor estimation of these probabilities can strongly affect the performance of the clustering process. A first strategy consists of including the probabilities as variables in the PSO particle as it is done for the mean and variance parameters. Since the sum of all prior probabilities must give one, a constrained PSO search implementation would be needed. As an alternative, we will adopt another simpler and faster strategy, which is based on the idea of optimizing by perturbation the priors outside but parallel with the PSO process. First, we start by clustering the n samples using the simple k-means algorithm [34] for getting an initial estimate of the priors. At each iteration of the PSO process, for each particle, a single prior probability $P(\omega_j)$ is selected at random. Then, its value is updated by adding a quantity Δ chosen randomly in the interval $[-P(\omega_j), 1 - P(\omega_j)]$. For the other classes, the prior probability values are updated by subtracting the amount $\Delta/(C - 1)$. This way, the constraint requirement is fulfilled, and all the particles remain in the feasible region of the optimization space.

C. Algorithm Description

In the following, we will describe the different phases on which the proposed MOPSO clustering method is based.

Phase 1—Parameter Setting:

Step 1.1) Choose the range of variation of the number of classes $[C_{\min}, C_{\max}]$.

Step 1.2) Set C to $C_{\min}$.

Phase 2—PSO Initialization:

Step 2.1) Initialize each particle position $\mathbf{p}_i$, $i = 1, 2, \ldots, S$, as follows:

a) Choose the number of detected features $\tilde{d}(\mathbf{p}_i)$ randomly in the interval [1, d].

b) Choose randomly $\tilde{d}(\mathbf{p}_i)$ feature detection coordinates; set them to 1, while fix all other feature detection coordinates to 0.

c) Run the k-means algorithm on the samples with a class number equal to C by considering only the selected features encoded by $\mathbf{p}_i$.

d) Initialize the mean value coordinates of the particle using the mean values given by the k-means algorithm. Coordinates associated with nonselected features are set to 0.

e) Initialize the variance value coordinates of the particle by computing for each class the variances on samples that are associated with that class by the k-means algorithm. Coordinates associated with nonselected features are set to 0.

f) Initialize the prior probabilities of each class by counting the sample number associated by the k-means algorithm with that class.

Step 2.2) Set the best position of each particle with its initial position, i.e.,

$$\mathbf{p}_{bi} = \mathbf{p}_i, \qquad i = 1, 2, \ldots, S. \tag{20}$$

Step 2.3) Set the velocity vectors $\mathbf{v}_i$, $i = 1, 2, \ldots, S$, that are associated with the S particles to 0.

Step 2.4) Compute for each candidate particle $\mathbf{p}_i$, $i = 1, 2, \ldots, S$, its fitness functions $f_1(\mathbf{p}_i, \mathbf{P}_c)$ and $f_2(\mathbf{p}_i)$.

Step 2.5) Identify the nondominated solutions by applying the nondominated sorting algorithm described in [35] and store them in a list *NL*. Assign a crowding distance to each particle of *NL*. This distance is computed basing on the two solutions of *NL* that surround the particle under consideration in the performance space (i.e., the space defined by the two fitness functions). The crowding distance plays a key role in an MO optimization process since it permits to force it in obtaining final solutions that are as spread as possible along the Pareto optimal front.

Phase 3—Search Process:

Step 3.1) Update the speed of each particle using (3). To perform the update, the best global position $\mathbf{p}_g$ is selected from the list *NL* according to a tournament-based selection [35]. This way, a nondominated solution with a higher crowding distance has a higher chance of being selected.

Step 3.2) Update the position of each particle by means of (4). If a particle goes beyond the predefined boundaries of the search space, truncate the updating by setting the position of the particle at the space boundary, and reverse its search direction (i.e., multiply its speed vector by -1). This will

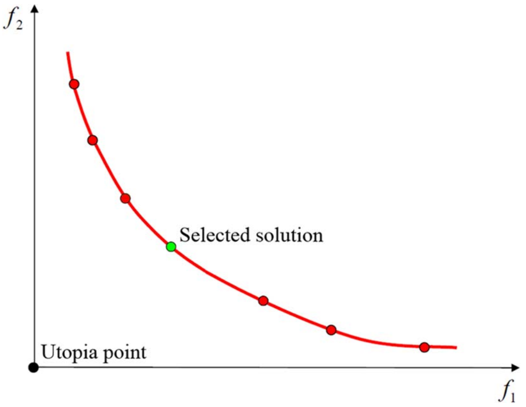


Fig. 2. Front of nondominated solutions.

stop the particles from further attempting to go outside the allowed search space.

Step 3.3) Update the prior probabilities of each class by means of the perturbation process described in Section IV-B.

Step 3.4) Compute the fitness functions $f_1(\mathbf{p}_i, \mathbf{P}_c)$ and $f_2(\mathbf{p}_i)$ for each candidate particle $\mathbf{p}_i$, $i = 1, 2, \ldots, S$.

Step 3.5) Update the content of *NL* by inserting the current nondominated solutions. Clean *NL* from previous nondominated solutions, which now become dominated.

Step 3.6) Update the best position $\mathbf{p}_{bi}$ of each particle if it is dominated by its current position $\mathbf{p}_i$, $i = 1, 2, \ldots, S$.

Phase 4—Convergence Check:

Step 4.1) Return to *Phase 3* if the convergence condition on the fitness functions or/and the maximal number of PSO iterations are not yet reached. Convergence conditions on the fitness functions are typically based on a check on the motion of the best global position in the current and the previous iterations.

Step 4.2) Increment the class number C by one unity, and return to *Phase 2* if C is less than or equal to the maximum class number $C_{\max}$.

Phase 5—Class Number Estimation:

Step 5.1) For each class number $C \in [C_{\min}, C_{\max}]$, select one of the nondominated solutions $\mathbf{p}_i^*$ from the list *NL*. As depicted in Fig. 2, in this paper, the closest solution to the origin of the performance space (utopia point) is selected. This solution will represent the best possible clustering model of the image when a data class number equal to C is assumed.

Step 5.2) Estimate the optimal number $\hat{C}$ of data classes by minimizing the MDL criterion.

Phase 6—Classification Map Generation: Generate a classification map by applying the MAP decision criterion for each image pixel. This criterion consists of assigning a generic sample $\mathbf{x}$ to the class ω_j if

$$\hat{P}(\omega_j|\mathbf{x}) > \hat{P}(\omega_i|\mathbf{x}), \qquad \forall i = 1, 2, \ldots, \hat{C}; \quad i \neq j. \tag{21}$$

Using Bayes' theorem, maximizing the posterior probability is equivalent to applying the following:

$$\hat{\omega} = \underset{i=1,\ldots,\hat{C}}{\arg\max} \left\{ \hat{P}(\omega_i)\hat{p}(\mathbf{x}|\omega_i) \right\} \tag{22}$$

where $\hat{P}(\omega_i)$ is the estimated prior, and

$$\hat{p}(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{\tilde{d}(\mathbf{p}^*)/2} \cdot \left|\tilde{\Sigma}_i(\mathbf{p}^*)\right|^{1/2}} \times \exp\left\{ -\frac{1}{2} \left(\tilde{\mathbf{x}}(\mathbf{p}^*) - \tilde{\boldsymbol{\mu}}_i(\mathbf{p}^*)\right)^T \tilde{\Sigma}_i(\mathbf{p}^*)^{-1} \left(\tilde{\mathbf{x}}(\mathbf{p}^*) - \tilde{\boldsymbol{\mu}}_i(\mathbf{p}^*)\right) \right\} \tag{23}$$

where $\mathbf{p}^*$ is the solution that minimizes the MDL criterion.

V. Experiments on Simulated Data

A. Experimental Design

The first part of the experiments refers to synthetic hyperspectral data. Three sets of experiments were carried out to assess the performance of the proposed MOPSO clustering methodology.

1) In the first set of experiments, we assessed its capability in terms of cluster parameter estimation. For this purpose, we generated 50 different hyperspectral images with a fixed spatial dimension of 100 × 100 pixels and with the following characteristics whose values were chosen randomly:
   a) number of features (bands) $d \in [1, 200]$;
   b) number of classes $C \in [2, 15]$;
   c) prior probability of each class $P(\omega_i) \in [0, 1]$, $i = 1, 2, \ldots, C$;
   d) mean value of each class and along each feature $\mu_{ij} \in [0, 100]$, with $i = 1, 2, \ldots, d$ and $j = 1, 2, \ldots, C$;
   e) variance of each class and along each feature $\sigma_{ij}^2 \in [0, 100]$, with $i = 1, 2, \ldots, d$ and $j = 1, 2, \ldots, C$.

   The hyperspectral value of each image sample was generated following a multivariate Gaussian distribution and according to the parameters previously defined. No noise was added to the data. The proposed methodology was run by minimizing only the first fitness function $f_1(\mathbf{p}, \mathbf{P}_c)$. This way, the feature detection process was inhibited. Moreover, we supposed that the class number C was known (and, thus, the MDL criterion was not used). The method performances were evaluated through the errors in terms of mean and variance estimation. The mean value error $Err(\mu_{ij})$, $i = 1, 2, \ldots, d$ and $j = 1, 2, \ldots, C$, is given by

$$Err(\mu_{ij}) = \frac{|\mu_{ij} - \mu_{r,ij}|}{\mu_{\max} - \mu_{\min}} \tag{24}$$

   where μ_{ij} is the estimated mean value of the jth class along the ith feature, $\mu_{r,ij}$ is the corresponding true mean value associated, $\mu_{\max}$ is the maximum true mean value over all features and all classes, and $\mu_{\min}$ is the minimum true mean value. Similarly, the variance

TABLE I
PERFORMANCES OF THE CLUSTER PARAMETER ESTIMATION PROCESS OBTAINED ON SIMULATED HYPERSPECTRAL IMAGES

	Minimum error [%]	Maximum error [%]	Average error [%]
Mean value	0.00	19.31	1.18
Variance	0.10	29.67	3.64

TABLE II
CONFUSION MATRIX YIELDED BY THE FEATURE DETECTION PROCESS ON SIMULATED HYPERSPECTRAL IMAGES

		Real	
		Noisy feature	Clean feature
Estimated	Noisy feature	95.0 %	1.3 %
	Clean feature	5.0 %	98.7 %

TABLE III
PERFORMANCES ACHIEVED ON SIMULATED HYPERSPECTRAL IMAGES BY THE ENTIRE MOPSO METHODOLOGY IN TERMS OF THE FOLLOWING: (a) CLUSTER PARAMETER ESTIMATION; (b) FEATURE DETECTION; AND (c) CLASS NUMBER ESTIMATION

(a)

	Minimum error [%]	Maximum error [%]	Average error [%]
Mean value	0.00	11.47	1.09
Variance	0.02	34.12	3.01

(b)

		Real	
		Noisy feature	Clean feature
Estimated	Noisy feature	94.5 %	1.7 %
	Clean feature	5.5 %	98.3 %

(c)

	Minimum absolute error	Maximum absolute error	Mean absolute error	Mean error
Number of data classes	0	6	2.60	-1.90

error $Err(\sigma_{ij}^2),\ i = 1, 2, \ldots, d$ and $j = 1, 2, \ldots, C$, is expressed as

$$Err\left(\sigma_{ij}^2\right) = \frac{\left|\sigma_{ij}^2 - \sigma_{r,ij}^2\right|}{\sigma_{\max}^2 - \sigma_{\min}^2}. \tag{25}$$

2) In the second set of experiments, we wanted to analyze the capability of the proposed methodology to detect noisy features. Accordingly, we generated 50 different hyperspectral images that have a number of features fixed to 200, while the other parameters (i.e., class number, prior probabilities, means, and variances) were generated randomly as in the previous experiments. Among the 200 features, 40 chosen at random were corrupted by an additive Gaussian noise to yield a signal-to-noise-ratio $(\mathrm{SNR}_{\mathrm{dB}})$ equal to 0 dB. After the generation of the images, the method was executed just by minimizing the fitness function $f_2(\mathbf{p})$. For each image, the number of noiseless features (160 features), the class statistical parameters, and the class number C were supposedly known.

3) Finally, the third experimental part was intended to evaluate the performances of the entire MOPSO clustering methodology. Again, we generated 50 different hyperspectral images by randomly setting the data characteristics (i.e., number of features, class number, prior probabilities, means, and variances) as in the first experiments. In addition, other parameters were generated randomly, which are listed as follows:
   a) number of noisy features $h \in [0,\ 0.8 \times d]$;
   b) signal-to-noise ratio of each noisy feature $\mathrm{SNR}_{\mathrm{dB},i} \in [-3, 10]$ dB, $i = 1, 2, \ldots, h$.

   From the d generated features, h features were chosen randomly. Then, each of them was corrupted by a noise component to produce a feature with a signal-to-noise ratio equal to $\mathrm{SNR}_{\mathrm{dB},i}, i = 1, 2, \ldots, h$. This time, the MOPSO method was run by using both fitness functions as well as the MDL criterion.

Concerning the PSO settings, in all experiments, we considered the following standard parameters: swarm size $S = 50$, inertia weight $w = 0.4$, acceleration constants c_1 and c_2 equal to unity, and maximum number of iterations fixed to 100.

B. Experimental Results

Experiment 1—Cluster Parameter Estimation: The achieved performances of the cluster parameter estimation process are summarized in Table I, in which we have reported the minimum, maximum, and average errors of the mean and variance estimates. The average errors suggest clearly that the proposed method exhibits a good estimation accuracy of the cluster statistical parameters.

Experiment 2—Feature Detection: The results of the feature detection process are shown in Table II, which represents a confusion matrix in terms of noisy/clean feature detection. Note that a noisy feature is correctly detected in 95.0% of the cases and the probability of correctly detecting clean features is 98.7%.

Experiment 3—Entire System Evaluation: When moving from a partial run (as in Experiments 1 and 2) to a complete run of the proposed methodology to perform simultaneously cluster parameter estimation, feature detection, and class number estimation, the results do not change significantly, as can be seen in Table III(a) and (b). The accuracy in terms of class number estimation is reported in Table III(c). A mean error equal to -1.90 was obtained, which means that on the average, the methodology underestimates the total number of classes with a difference of about two classes.

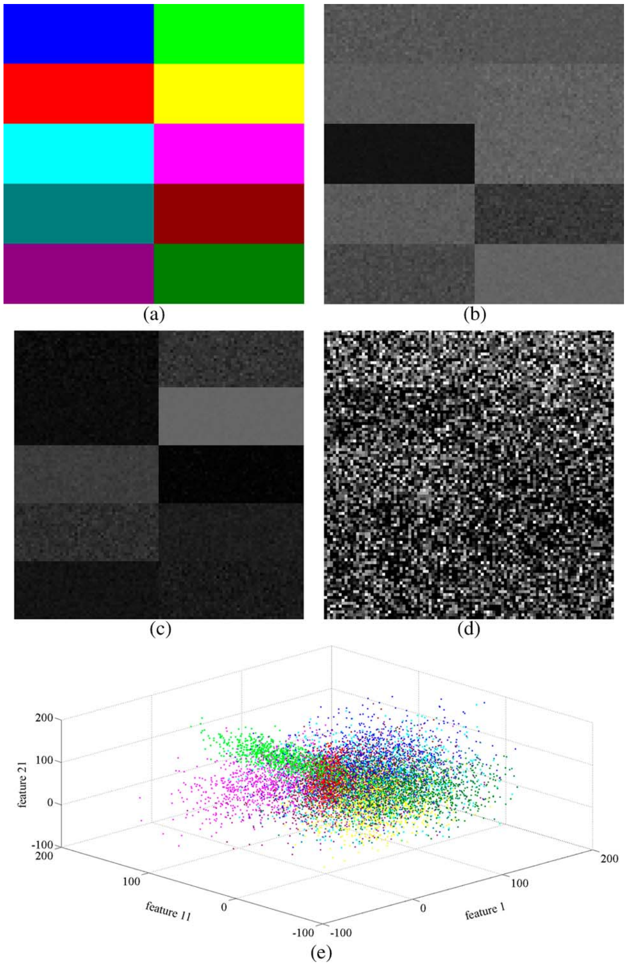


Fig. 3. Example of a simulated hyperspectral image on which the whole MOPSO methodology was tested. (a) Ground truth. (b) and (c) Examples of clean bands. (d) Example of a noisy band. (e) Class statistical distribution in a 3-D feature subspace.

In order to illustrate in detail the results achieved, two complete examples of simulated hyperspectral image clustering are reported. In the first case, shown in Fig. 3, data characteristics were set as follows:

1) number of features $d = 100$;
2) number of classes $C = 10$;
3) number of noisy features $h = 0.5 \times d = 50$;
4) signal-to-noise ratio of each noisy feature $\mathrm{SNR}_{\mathrm{dB},i} = 0\ \mathrm{dB},\ i = 1, 2, \ldots, 50$.

Fig. 3(a) represents the ground truth, in which each of the ten classes is represented with a different color. Some of the randomly generated image bands are illustrated in Fig. 3(b)–(d). The distribution of the ten classes along three clean features (features 1, 11, and 21) is given in Fig. 3(e).

The plot shown in Fig. 3(f) provides the values of the MDL criterion for a range of class numbers going from 3 to 13 classes. Note that the minimum MDL value is obtained for a class number of ten, which corresponds to the actual number of data classes in the image. The results in terms of feature detection are illustrated in Fig. 3(g). The 50 clean features of the hyperspectral image are represented in green (features $\{1 \div 25, 76 \div 100\}$), while the 50 noisy features are depicted in red (features $\{26 \div 75\}$). The MOPSO detected nine features,

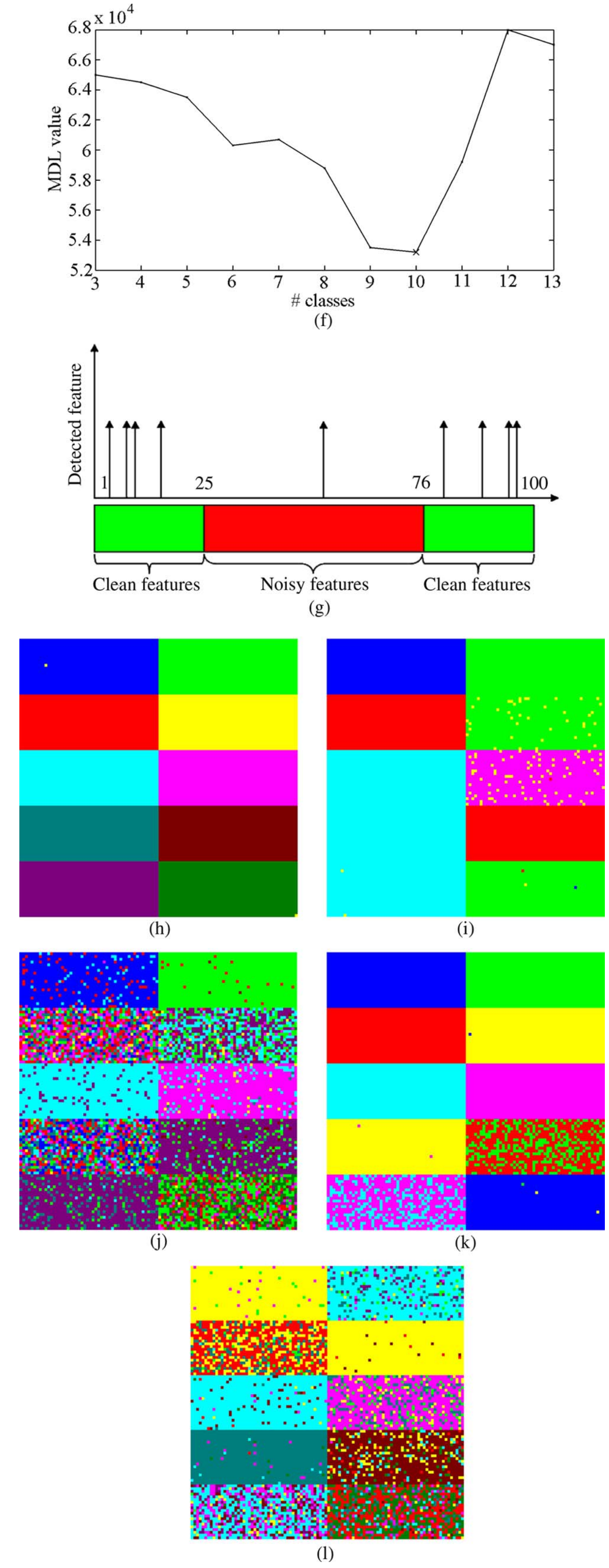


Fig. 3. *(Continued.)* Example of a simulated hyperspectral image on which the whole MOPSO methodology was tested. (f) Behavior of the MDL criterion. (g) Result of the feature detection process. Classification maps yielded by (h) the MOPSO, (i) the k-means, (j) the fuzzy C-means, (k) the PCA + k-means, and (l) the PCA + fuzzy C-means methods.

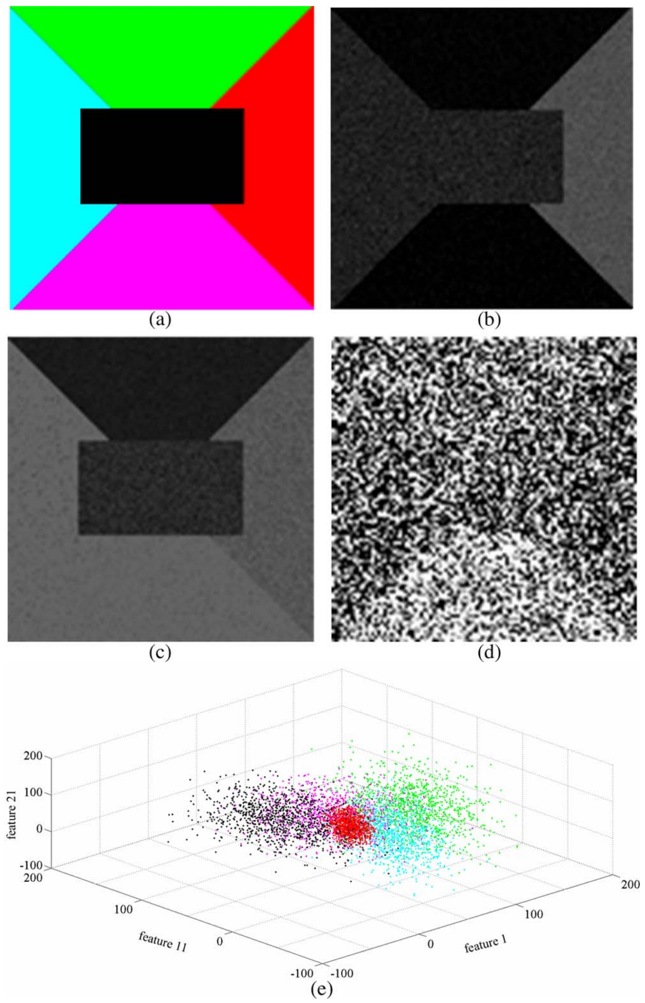


Fig. 4. Another example of a simulated hyperspectral image on which the whole MOPSO methodology was evaluated. Captions are similar to those of Fig. 3.

which are marked by arrows. Note that eight detected features were really clean features, while one feature was wrongly associated with a noisy band. The classification map generated using the MAP criterion is given in Fig. 3(h). The corresponding classification accuracy is equal to 99.98%. The same hyperspectral image was classified with the k-means and the fuzzy C-means algorithms. In these two cases, the classification accuracies are equal to 49.55% and 51.28%, respectively. The corresponding classification maps are shown in Fig. 3(i) and (j). We ran also the clustering with the k-means and the fuzzy C-means methods after applying a feature reduction step based on the first ten principal components produced by the well-known principal components analysis (PCA) technique [36]. The resulting classification maps are illustrated in Fig. 3(k) and (l). The classification accuracies are equal to 59.99% and 53.59%, respectively.

Another detailed example is given in Fig. 4, for which data characteristics were given as follows:

1) number of features $d = 100$;
2) number of classes $C = 5$;
3) number of noisy features $h = 0.4 \times d = 40$;
4) signal-to-noise ratio of each noisy feature $\mathrm{SNR}_{\mathrm{dB},i} = 0\ \mathrm{dB},\ i = 1, 2, \ldots, 40$.

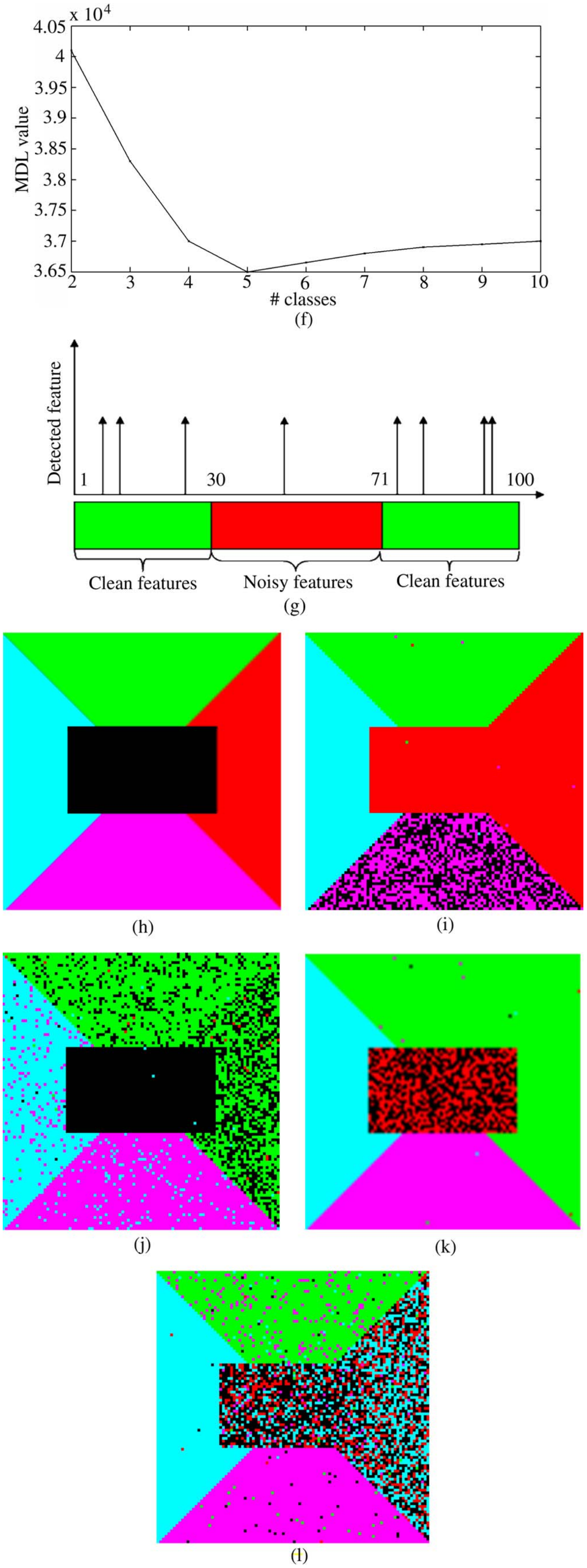
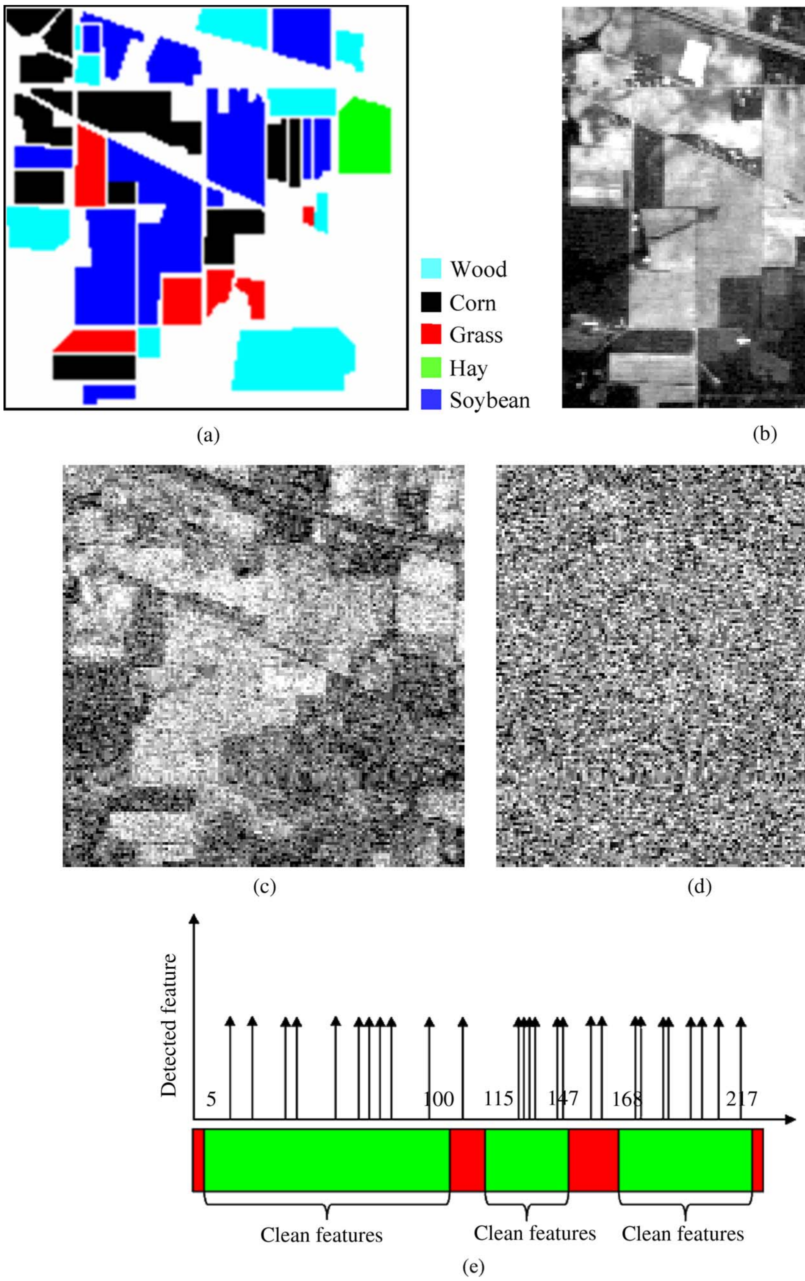


Fig. 4. *(Continued.)* Another example of a simulated hyperspectral image on which the whole MOPSO methodology was evaluated. Captions are similar to those of Fig. 3.



(a) (b) (c) (d) (e)

Fig. 5. Results obtained on a real hyperspectral image acquired by the AVIRIS sensor. (a) Ground truth. Examples of (b) clean, (c) partially noisy, and (d) noisy bands. (e) Result of the feature detection process.

The MDL criterion allowed to correctly find the class number [Fig. 4(f)]. The feature detection result is illustrated in Fig. 4(g). The classification accuracies yielded by the MOPSO, the k-means algorithm, the fuzzy C-means algorithm, the k-means, and the fuzzy C-means algorithms applied after PCA feature reductions (using the first ten components) are equal to 100%, 71.35%, 73.58%, 72.85%, 74.46%, respectively. The corresponding maps are shown in Fig. 4(h)–(l).

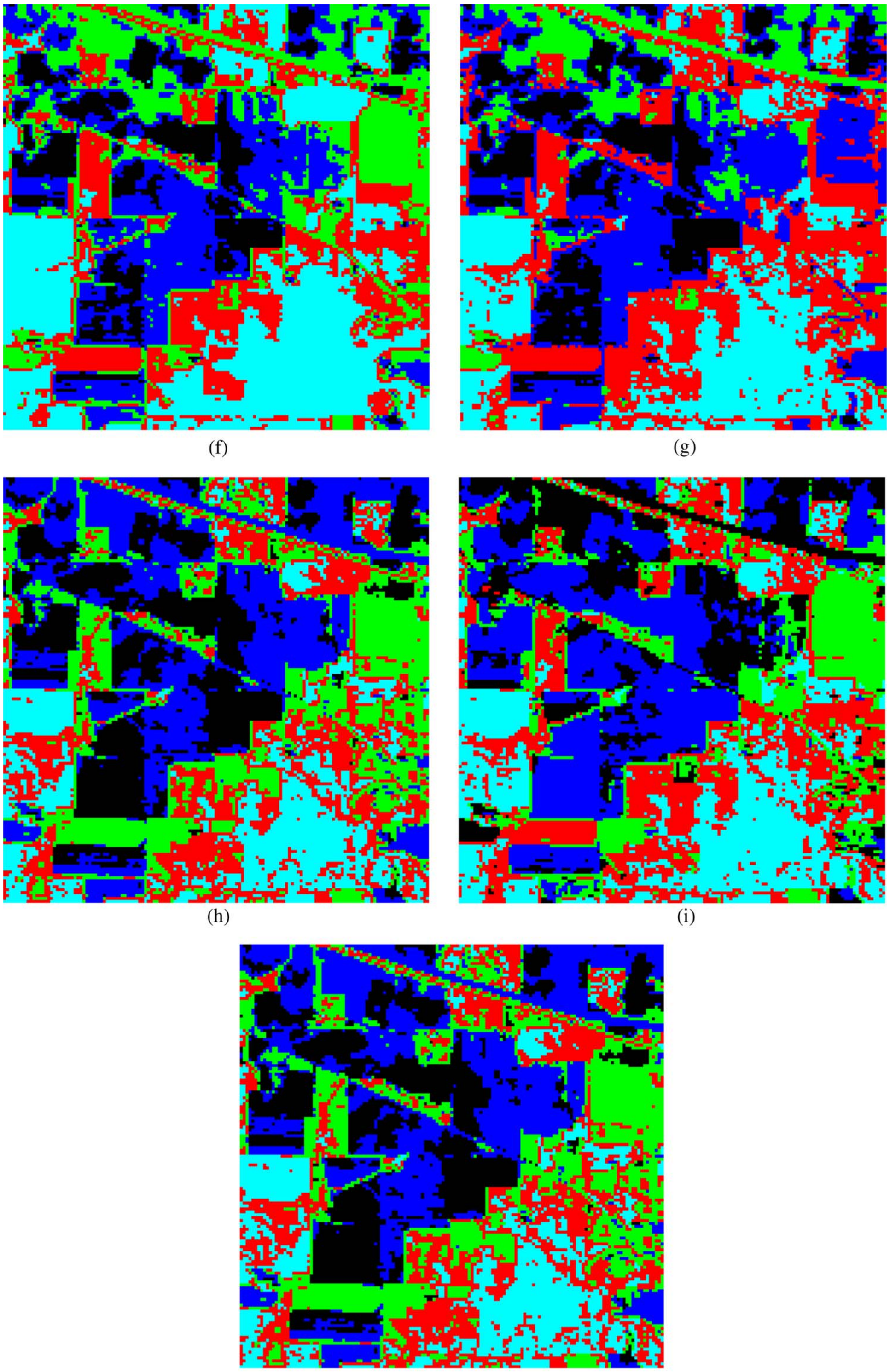
VI. Experiments on Real Data

A. Data Set Description

The second part of the experiments refers to real hyperspectral images. In particular, two different data sets were considered.

The first data set represents a hyperspectral image composed of 220 spectral bands. The image was acquired in 1992 by the AVIRIS over northwest Indiana's Indian Pines. The image has a dimension equal to 145 × 145 pixels and is characterized by 16 land cover classes. In this paper, we exploited the available ground truth to get an indication of the accuracy of the MOPSO clustering, which is intrinsically unsupervised. For such a purpose, we merged the land cover classes to get a ground truth composed of five classes as shown in Fig. 5(a). Some of the 220 bands are completely noisy and some partially as shown in Fig. 5(b)–(d).

The second image is characterized by 102 bands and was acquired by the ROSIS over a part of the city of Pavia, northern Italy. A 400 × 400 pixel crop of the original image was used in our experiments [see Fig. 6(a) and (b)]. For this image, we



(f) (g) (h) (i) (j)

Fig. 5. *(Continued.)* Results obtained on a real hyperspectral image acquired by the AVIRIS sensor. Classification maps given by (f) the MOPSO, (g) the k-means, (h) the fuzzy C-means, (i) the PCA + k-means, and (j) the PCA + fuzzy C-means methods.

relied on the very high spatial resolution (1.2 m) of the image to assess qualitatively the classification results.

B. Experimental Results

For the first data set, the MOPSO was run by fixing the desired class number to five (therefore, the MDL criterion was not used) to yield a classification map that can be compared with the ground truth. Therefore, for this image, we only assessed the cluster parameter estimation and feature detection capability of the method. At convergence, 27 (among 220) features were picked up. From a visual inspection, 24 features are associated with clean bands, while three features were affected by the presence of noise. The detected bands are depicted in

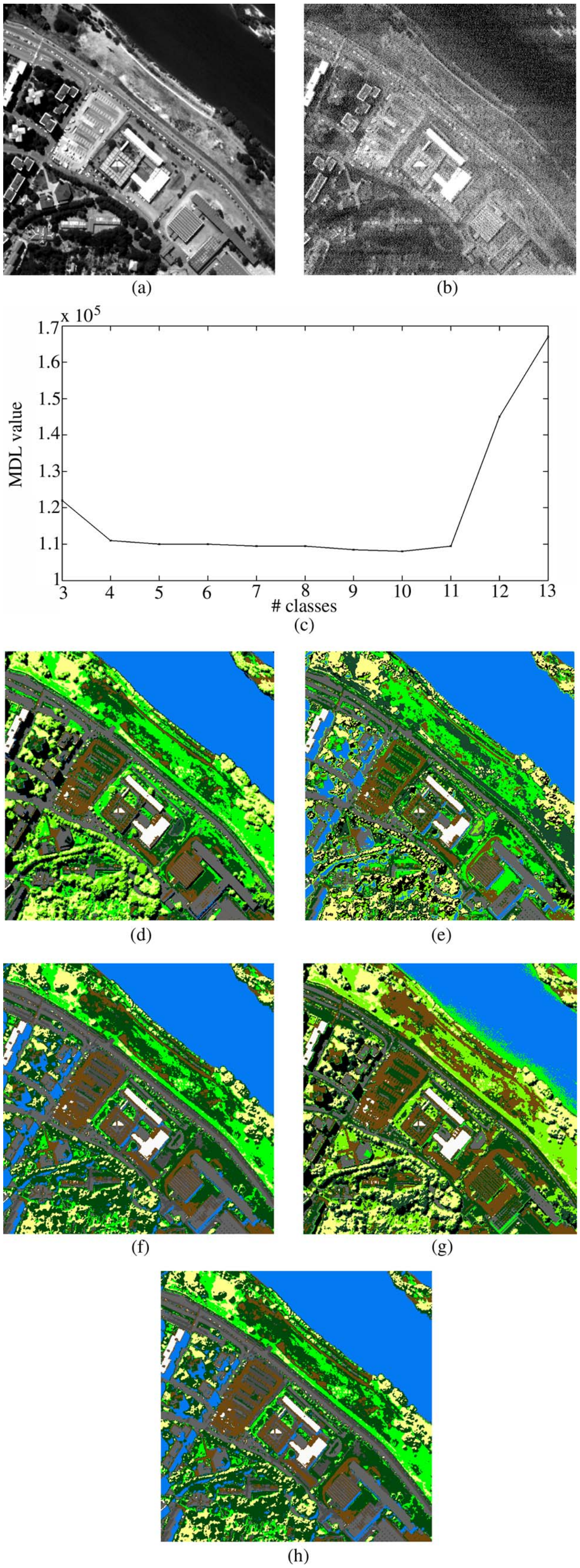


Fig. 6. Results obtained on a second real hyperspectral image acquired by the ROSIS sensor. Examples of (a) clean and (b) noisy bands. (c) Behavior of the MDL criterion. Classification maps given by (d) the MOPSO, (e) the k-means, (f) the fuzzy C-means, (g) the PCA + k-means, and (h) the PCA + fuzzy C-means methods.

Fig. 5(e). The classification maps given by the MOPSO, the k-means algorithm, the fuzzy C-means algorithm, the PCA + k-means algorithm, and the PCA + fuzzy C-means algorithm are illustrated in Fig. 5(f)–(j), respectively. The classification accuracies summarized in Table IV point out the superiority of our method, which yielded an overall accuracy (OA) of 67.22% against 53.41%, 53.98%, 57.30%, and 53.94% for the k-means, fuzzy C-means, PCA (first ten components) + k-means, and PCA (first ten components) + fuzzy C-means methods, respectively. The same holds in terms of average accuracy (AA). It is noteworthy that the accuracy values are not high since all the five methods are unsupervised.

For the second data set, the class number is not known. Therefore, the entire MOPSO methodology, including the MDL-based class number estimation procedure, was executed. The behavior of the MDL criterion is shown in Fig. 6(c), from which it comes out that the image contains ten data classes. The feature detection process picked up 16 features, which all correspond to clean bands. The classification maps given by the five investigated methods are provided in Fig. 6(d)–(h). In particular, Fig. 6(e) and (f) reveals that k-means and fuzzy C-means algorithms wrongly assign water areas outside the Ticino river. The PCA (first ten components) + k-means algorithm fixes this problem but wrongly assigns vegetation areas in parking areas and within the Ticino river.

VII. Conclusion

In this paper, a novel methodology for the unsupervised classification of hyperspectral images has been presented. It allows to solve simultaneously problems of clustering, feature detection, and class number estimation in a completely automatic and unsupervised way. To the best of our knowledge, in the literature, these multiple issues have not yet been addressed within one processing framework. The proposed MOPSO solution provides an effective answer to this complex challenge, as shown by the experimental results. Indeed, it provides a very satisfactory classification accuracy while drastically reducing the number of bands used for the classification task. It yields a good guess of the number of data classes characterizing the considered image. Such a guess refers to data classes and not to thematic classes, which do not necessarily match. It could be refined in a second step by the user if it is in possession of some prior knowledge about the scene. The main drawback of the proposed methodology is the computational time required by the optimization process, which may reach several hours, depending on the image size.

Acknowledgment

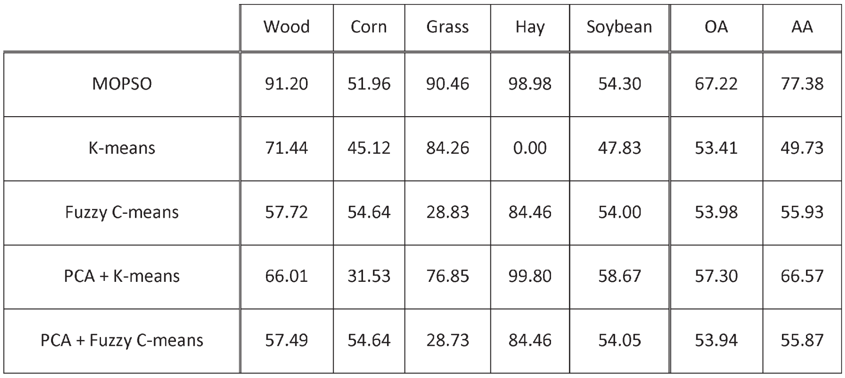
The authors would like to thank Prof. D. Landgrebe and Prof. P. Gamba (University of Pavia, Italy) for providing the hyperspectral images used in the experiments.

References

[1] Y. Wong and E. C. Posner, “A new clustering algorithm applicable to multispectral and polarimetric SAR images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 31, no. 3, pp. 634–644, May 1993.

[2] G. Palubinskas, M. Datcu, and R. Pac, “Clustering algorithms for large sets of heterogeneous remote sensing data,” in *Proc. IGARSS*, Hamburg, Germany, 1999, vol. 3, pp. 1591–1593.

TABLE IV
OA, AA, AND CLASS ACCURACIES IN PERCENT YIELDED BY THE FIVE INVESTIGATED METHODS ON THE AVIRIS IMAGE



	Wood	Corn	Grass	Hay	Soybean	OA	AA
MOPSO	91.20	51.96	90.46	98.98	54.30	67.22	77.38
K-means	71.44	45.12	84.26	0.00	47.83	53.41	49.73
Fuzzy C-means	57.72	54.64	28.83	84.46	54.00	53.98	55.93
PCA + K-means	66.01	31.53	76.85	99.80	58.67	57.30	66.57
PCA + Fuzzy C-means	57.49	54.64	28.73	84.46	54.05	53.94	55.87

[3] M. Barni, A. Garzelli, A. Mecocci, and L. Sabatini, "A robust fuzzy clustering algorithm for the classification of remote sensing images," in *Proc. IGARSS*, Honolulu, HI, 2000, vol. 5, pp. 2143–2145.

[4] X. Jia and J. A. Richards, "Cluster-space representation for hyperspectral data classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 3, pp. 593–598, Mar. 2002.

[5] X. Jia and J. A. Richards, "Efficient transmission and classification of hyperspectral image data," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 5, pp. 1129–1131, May 2003.

[6] H. Shi, Y. Shen, and Z. Liu, "Hyperspectral bands reduction based on rough sets and fuzzy C-means clustering," in *Proc. 20th IEEE IMTC*, 2003, vol. 2, pp. 1053–1056.

[7] A. Gda, S. Filho, A. C. Frery, C. C. de Araujo, H. Alice, J. Cerqueira, J. A. Loureiro, M. E. de Lima, M. G. S. Oliveira, and M. M. Horta, "Hyperspectral images clustering on reconfigurable hardware using the k-means algorithm," in *Proc. 16th Symp. Integr. Circuits Syst. Des. SBCCI*, 2003, pp. 99–104.

[8] S. Lee and M. M. Crawford, "Hierarchical clustering approach for unsupervised image classification of hyperspectral data," in *Proc. IGARSS*, Anchorage, AK, 2004, vol. 2, pp. 941–944.

[9] P. R. Kersten, J.-S. Lee, and T. L. Ainsworth, "Unsupervised classification of polarimetric synthetic aperture radar images using fuzzy clustering and EM clustering," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 3, pp. 519–527, Mar. 2005.

[10] A. R. S. Marcal and L. Castro, "Hierarchical clustering of multispectral images using combined spectral and spatial criteria," *IEEE Geosci. Remote Sens. Lett.*, vol. 2, no. 1, pp. 59–63, Jan. 2005.

[11] T. N. Tran, R. Wehrens, D. H. Hoekman, and L. M. C. Buydens, "Initialization of Markov random field clustering of large remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 43, no. 8, pp. 1912–1919, Aug. 2005.

[12] Y. Zhong, L. Zhang, B. Huang, and P. Li, "An unsupervised artificial immune classifier for multi/hyperspectral remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 2, pp. 420–431, Feb. 2006.

[13] S. Bandyopadhyay, U. Maulik, and A. Mukhopadhyay, "Multiobjective genetic clustering for pixel classification in remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 5, pp. 1506–1511, May 2007.

[14] X. Liu, X. Li, Y. Zhang, C. Yang, W. Xu, M. Li, and H. Luo, "Remote sensing image classification based on dot density function weighted FCM clustering algorithm," in *Proc. IGARSS*, Barcelona, Spain, 2007, vol. 2, pp. 2010–2013.

[15] G.-S. Xia, C. He, and H. Sun, "A rapid and automatic MRF-based clustering method for SAR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 4, pp. 596–600, Oct. 2007.

[16] M. Tyagi, F. Bovolo, A.K. Mehra, S. Chaudhuri, and L. Bruzzone, "A context-sensitive clustering technique based on graph-cut initialization and expectation–maximization algorithm," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 1, pp. 21–25, Jan. 2008.

[17] G. Bilgin, S. Ertürk, and T. Yildirim, "Unsupervised classification of hyperspectral-image data using fuzzy approaches that spatially exploit membership relations," *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 4, pp. 673–677, Oct. 2008.

[18] M. Dundar and D. Landgrebe, "A model-based mixture-supervised classification approach in hyperspectral data analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 12, pp. 2692–2699, Dec. 2002.

[19] A. Berge and A. H. S. Solberg, "Structured Gaussian components for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3386–3396, Nov. 2006.

[20] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Mateo, CA: Morgan Kaufmann, 2001.

[21] Y. Bazi and F. Melgani, "Semisupervised PSO-SVM regression for biophysical parameter estimation," *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 6, pp. 1887–1895, Jun. 2007.

[22] Y. Bazi and F. Melgani, "Toward an optimal SVM classification system for hyperspectral remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3374–3385, Nov. 2006.

[23] N. Ghoggali, F. Melgani, and Y. Bazi, "A multiobjective genetic SVM approach for classification problems with limited training samples," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 6, pp. 1707–1718, Jun. 2009.

[24] Z. L. Gaing, "A particle swarm optimization approach for optimum design of PID controller in AVR system," *IEEE Trans. Energy Convers.*, vol. 19, no. 2, pp. 384–391, Jun. 2004.

[25] M. Donelli, R. Azaro, F. G. B. De Natale, and A. Massa, "An innovative computational approach based on a particle swarm strategy for adaptive phased-arrays control," *IEEE Trans. Antennas Propag.*, vol. 54, no. 3, pp. 888–898, Mar. 2006.

[26] W. H. Slade, H. W. Resson, M. T. Musavi, and R. L. Miller, "Inversion of ocean color observations using particle swarm optimization," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 9, pp. 1915–1923, Sep. 2004.

[27] F. Melgani and Y. Bazi, "Classification of electrocardiogram signals with support vector machines and particle swarm optimization," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 5, pp. 667–677, Sep. 2008.

[28] S. L. Ho, S. Yang, G. Ni, E. W. C. Lo, and H. C. Wong, "A particle swarm optimization-based method for multiobjective design optimizations," *IEEE Trans. Magn.*, vol. 41, no. 5, pp. 1756–1759, May 2005.

[29] C. A. Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 256–279, Jun. 2004.

[30] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.

[31] G. McLachlan and D. Peel, *Finite Mixture Models*. New York: Wiley, 2000.

[32] J. Rissanen, *Stochastic Complexity in Statistical Inquiry*. Singapore: World Scientific, 1989.

[33] Z. Liang, R. J. Jaszczak, and R. E. Coleman, "Parameter estimation of finite mixtures using the EM algorithm and information criteria with application to medical image processing," *IEEE Trans. Nucl. Sci.*, vol. 39, no. 4, pp. 1126–1133, Aug. 1992.

[34] A. Webb, *Statistical Pattern Recognition*. Chichester, U.K.: Wiley, 2002.

[35] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001.

[36] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Phil. Mag.*, vol. 2, no. 6, pp. 559–572, 1901.

Andrea Paoli received the M.Sc. degree in telecommunications engineering from the University of Trento, Trento, Italy, in 2008.

He is currently a Project Engineer for a telecommunications company in Milan, Italy. His research interests include image processing, video surveillance, and networking.

Farid Melgani (M'04–SM'06) received the State Engineer degree in electronics from the University of Batna, Batna, Algeria, in 1994, the M.Sc. degree in electrical engineering from the University of Baghdad, Baghdad, Iraq, in 1999, and the Ph.D. degree in electronic and computer engineering from the University of Genoa, Genova, Italy, in 2003.

From 1999 to 2002, he cooperated with the Signal Processing and Telecommunications Group, Department of Biophysical and Electronic Engineering, University of Genoa. Since 2002, he has been an Assistant Professor of telecommunications with the University of Trento, Trento, Italy, where he has taught pattern recognition, machine learning, radar remote-sensing systems, and digital transmission. He is currently the Head of the Intelligent Information Processing (I2P) Laboratory, Department of Information Engineering and Computer Science, University of Trento. His research interests include processing, pattern recognition, and machine learning techniques applied to remote sensing and biomedical signals/images (classification, regression, multitemporal analysis, and data fusion). He is a coauthor of more than 80 scientific publications and is a referee for several international journals.

Dr. F. Melgani has served on the scientific committees of several international conferences and is an Associate Editor for the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS.

Edoardo Pasolli (S'08) received the M.Sc. degree in telecommunications engineering in 2008 from the University of Trento, Trento, Italy, where he is currently working toward the Ph.D. degree in information and communication technologies.

He is currently with the Intelligent Information Processing Laboratory, Department of Information Engineering and Computer Science, University of Trento. His research interests include processing and recognition techniques applied to remote sensing images and biomedical signals (classification, regression, and machine learning).