Hyperspectral Image Classification Using Dictionary-Based Sparse Representation

Yi Chen, Nasser M. Nasrabadi, *Fellow, IEEE*, and Trac D. Tran, *Senior Member, IEEE*

***Abstract*—A new sparsity-based algorithm for the classification of hyperspectral imagery is proposed in this paper. The proposed algorithm relies on the observation that a hyperspectral pixel can be sparsely represented by a linear combination of a few training samples from a structured dictionary. The sparse representation of an unknown pixel is expressed as a sparse vector whose nonzero entries correspond to the weights of the selected training samples. The sparse vector is recovered by solving a sparsity-constrained optimization problem, and it can directly determine the class label of the test sample. Two different approaches are proposed to incorporate the contextual information into the sparse recovery optimization problem in order to improve the classification performance. In the first approach, an explicit smoothing constraint is imposed on the problem formulation by forcing the *vector Laplacian* of the reconstructed image to become zero. In this approach, the reconstructed pixel of interest has similar spectral characteristics to its four nearest neighbors. The second approach is via a joint sparsity model where hyperspectral pixels in a small neighborhood around the test pixel are simultaneously represented by linear combinations of a few common training samples, which are weighted with a different set of coefficients for each pixel. The proposed sparsity-based algorithm is applied to several real hyperspectral images for classification. Experimental results show that our algorithm outperforms the classical supervised classifier support vector machines in most cases.**

***Index Terms*—Classification, hyperspectral imagery, joint sparsity model, simultaneous sparse recovery, sparse representation, spatial correlation.**

I. Introduction

SPARSITY OF signals has been an extremely powerful tool in many classical signal processing applications, such as compression and denoising [1], as most natural signals can be compactly represented by only a few coefficients that carry the most important information in a certain basis or dictionary. Recently, applications of sparse data representation have been extended to the area of computer vision and pattern recognition [2] with the development of the compressed sensing (CS) framework [3], [4] and sparse modeling of signals and images [5]. These applications are mainly based on the observation that despite the high dimensionality of natural signals, the signals in the same class usually lie in a low-dimensional subspace. Therefore, for every typical sample, there exists a sparse representation with respect to some proper basis which encodes the semantic information. The CS theories ensure that a sparse signal can be recovered from its incomplete but incoherent projections with a high probability. This enables the recovery of the sparse representation by decomposing the sample over a usually overcomplete dictionary generated by or learned from representative samples. Once the sparse representation vector is obtained, the semantic information can be directly extracted from the recovered vector. Applications of sparse representation in computer vision and pattern recognition can be found in various fields, including motion segmentation [6], [7], image super-resolution [8], image restoration [9], [10], and discriminative tasks including face recognition [11], iris recognition [12], tumor classification [13], and hyperspectral unmixing [14] and target detection [15]. In these applications, the usage of sparsity as a prior often leads to state-of-the-art performance.

In the hyperspectral case, the remote sensors capture digital images in hundreds of narrow spectral bands spanning the visible to infrared spectrum [16]. Pixels in hyperspectral imaging (HSI) are represented by vectors whose entries correspond to the spectral bands. Different materials usually reflect electromagnetic energy differently at specific wavelengths. This enables discrimination of materials based on the spectral characteristics. HSI has found many applications in various fields, such as military [17]–[19], agriculture [20], [21], and mineralogy [22]. A very important application of HSI is image classification where pixels are labeled to one of the classes. Various techniques have been developed for HSI classification. Among these previous approaches, support vector machines (SVMs) [23], [24] have been a powerful tool to solve supervised classification problems for high-dimensional data and have shown a good performance for hyperspectral classification [25], [26]. Variations of the SVM-based algorithms have also been proposed to improve the classification accuracy. These variations include semisupervised learning which exploits both labeled and unlabeled samples [27], postprocessing of the individually labeled samples based on certain decision rules [28], [29], and incorporating spatial information directly in the SVM kernels [30], [31]. More recent HSI classification techniques can be found in [32]–[39].

In this paper, we propose a classification algorithm for HSI that utilizes the sparsity of the input sample with respect to a given overcomplete training dictionary. The proposed algorithm is based on a sparsity model where a test spectral pixel is approximately represented by a few training samples (atoms) among the entire training dictionary. The sparse vector representing the atoms and their associated weights for the test

Manuscript received December 21, 2010; revised February 26, 2011; accepted March 3, 2011. This work is supported in part by ARO Grant 58110-MA-II and in part by NSF Grant CCF-0728893.

Y. Chen and T. D. Tran are with the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: ychen98@jhu.edu; trac@jhu.edu).

N. M. Nasrabadi is with US Army Research Laboratory, Adelphi, MD 20783 USA (e-mail: nnasraba@arl.army.mil).

Digital Object Identifier 10.1109/TGRS.2011.2129595

spectral pixel can be recovered by solving an optimization problem constrained by the sparsity level and reconstruction accuracy. The class of the test pixel can then be determined by the characteristics of the recovered sparse vector.

HSI usually have large homogeneous regions where the neighboring pixels within the regions consist of the same type of materials (same class) and have a similar spectral characteristics. Previous works have shown that it is important to take into account the contextual information in HSI classification [28]–[31], [40], [41]. This is usually done by either combining the spatial and spectral information in the classification stage or postprocessing the decisions obtained from individual pixels. In this paper, during the recovery of the sparse vector, in addition to the constraints on sparsity and reconstruction accuracy, we also exploit the spatial smoothness across neighboring HSI pixels. Two different approaches are proposed to incorporate the contextual information directly in the sparse recovery problem. In the first approach, a local smoothing constraint is imposed to the optimization problem by forcing the vector Laplacian at the reconstructed pixel to be zero. Thus, the reconstructed pixel of interest is forced to have spectral characteristics similar to its four nearest neighbors. The proposed reconstruction problem with the explicit smoothing constraint can be reformulated as a standard sparsity-constrained optimization problem and then solved efficiently by available optimization tools. In the second approach, we exploit the interpixel correlation between neighboring HSI pixels by adopting a joint sparsity model [42], [43], where pixels in a small neighborhood are assumed to be simultaneously represented by a few common training samples, but for each pixel, these selected training samples are weighted with a different set of coefficients. In this way, the sparse vector representations of neighboring pixels are forced to have a common support corresponding to the common atoms in the given training dictionary. The support is recovered by simultaneously decomposing the neighboring pixels over the given training dictionary. In both approaches, labels of the test samples are determined by the property of the recovered sparse vectors. Both proposed approaches enforce the smoothness constraint across neighboring pixels within the optimization process during the classification stage, rather than employing a postprocessing scheme to exploit the contextual information.

The remainder of this paper is structured as follows. The proposed sparsity-based classification algorithm is introduced in Section II. The details of the two approaches used to incorporate the spatial information in the proposed sparsity-based classification techniques are described in Section III. The effectiveness of the proposed method is demonstrated in Section IV by simulation results on several real hyperspectral images. Finally, Section V summarizes this paper and makes some closing remarks.

II. Classification of HSI via Sparse Representation

In this section, we first introduce a sparsity-based HSI classification algorithm by representing the test sample by a sparse linear combination of training samples from a dictionary. We then discuss the standard algorithms used to solve for the sparse representation, as well as the procedure used for determining the label of the test pixel.

A. Sparsity Model

In the proposed sparsity model, the spectral signatures of pixels belonging to the same class are assumed to approximately lie in a low-dimensional subspace. Suppose we have M distinct classes, and the mth class has N_m training samples $\{\boldsymbol{a}_j^m\}_{j=1,\ldots,N_m}$. Let $\boldsymbol{x}$ be a B-dimensional hyperspectral pixel observation. If $\boldsymbol{x}$ belongs to the mth class, then its spectrum approximately lies in a low-dimensional subspace spanned by the training samples $\{\boldsymbol{a}_j^m\}_{j=1,\ldots,N_m}$ in the mth class. The pixel $\boldsymbol{x}$ can be compactly represented by a linear combination of these training samples

$$\begin{aligned}\boldsymbol{x} &\approx \alpha_1^m \boldsymbol{a}_1^m + \alpha_2^m \boldsymbol{a}_2^m + \ldots + \alpha_{N_m}^m \boldsymbol{a}_{N_m}^m \\ &= \underbrace{\begin{bmatrix}\boldsymbol{a}_1^m & \boldsymbol{a}_2^m & \ldots & \boldsymbol{a}_{N_m}^m\end{bmatrix}}_{\boldsymbol{A}^m} \underbrace{\begin{bmatrix}\alpha_1^m & \alpha_2^m & \ldots & \alpha_{N_m}^m\end{bmatrix}^T}_{\boldsymbol{\alpha}^m} \\ &= \boldsymbol{A}^m \boldsymbol{\alpha}^m \end{aligned} \tag{1}$$

where $\boldsymbol{A}^m$ is a $B \times N_m$ class subdictionary whose columns are the training samples in the mth class; and $\boldsymbol{\alpha}^m$ is an unknown N_m-dimensional vector whose entries are the weights of the corresponding atoms in $\boldsymbol{A}^m$. In the sparsity model, $\boldsymbol{\alpha}^m$ is a sparse vector (i.e., $\boldsymbol{\alpha}^m$ has only a few nonzero entries).

Therefore, an unknown test sample can be modeled to lie in the union of the M subspaces associated with the M classes. By combining the class subdictionaries $\{\boldsymbol{A}^m\}_{m=1,\ldots,M}$, a test sample $\boldsymbol{x}$ can

$$\begin{aligned}\boldsymbol{x} &= \boldsymbol{A}^1\boldsymbol{\alpha}^1 + \boldsymbol{A}^2\boldsymbol{\alpha}^2 + \ldots + \boldsymbol{A}^M\boldsymbol{\alpha}^M \\ &= \underbrace{\begin{bmatrix}\boldsymbol{A}^1 & \ldots & \boldsymbol{A}^M\end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix}\boldsymbol{\alpha}^1 \\ \vdots \\ \boldsymbol{\alpha}^M\end{bmatrix}}_{\boldsymbol{\alpha}} = \boldsymbol{A}\boldsymbol{\alpha}\end{aligned} \tag{2}$$

where $\boldsymbol{A}$ is a $B \times N$ structured dictionary consisting of training samples from all classes with $N = \sum_{m=1}^{M} N_m$; and $\boldsymbol{\alpha}$ is an N-dimensional sparse vector formed by concatenating the sparse vectors $\{\boldsymbol{\alpha}^m\}_{m=1,\ldots,M}$. Note that ideally if $\boldsymbol{x}$ belongs to the mth class, then $\boldsymbol{\alpha}^j = \boldsymbol{0}, \forall j = 1,\ldots,M, j \neq m$. The sparse representation of $\boldsymbol{x}$ can also be written as a linear combination of only the K active dictionary atoms $\boldsymbol{a}_{\lambda_k}$ corresponding to the K nonzero entries $\alpha_{\lambda_k}, k = 1,\ldots,K$

$$\begin{aligned}\boldsymbol{x} &= \alpha_{\lambda_1}\boldsymbol{a}_{\lambda_1} + \alpha_{\lambda_2}\boldsymbol{a}_{\lambda_2} + \ldots + \alpha_{\lambda_M}\boldsymbol{a}_{\lambda_K} \\ &= \underbrace{\begin{bmatrix}\boldsymbol{a}_{\lambda_1} & \ldots & \boldsymbol{a}_{\lambda_K}\end{bmatrix}}_{\boldsymbol{A}_{\Lambda_K}} \underbrace{\begin{bmatrix}\alpha_{\lambda_1} \\ \vdots \\ \alpha_{\lambda_K}\end{bmatrix}}_{\boldsymbol{\alpha}_{\Lambda_K}} = \boldsymbol{A}_{\Lambda_K}\boldsymbol{\alpha}_{\Lambda_K}\end{aligned} \tag{3}$$

where $K = \|\boldsymbol{\alpha}\|_0$ denotes the ℓ_0-norm (or sparsity level) of $\boldsymbol{\alpha}$ which is defined as the number of nonzero entries in $\boldsymbol{\alpha}$; the index set $\Lambda_K = \{\lambda_1, \lambda_2, \ldots, \lambda_K\}$ is the support of $\boldsymbol{\alpha}$; $\boldsymbol{A}_{\Lambda_K}$ is a $B \times K$ matrix whose columns are the K atoms $\{\boldsymbol{a}_k\}_{k\in\Lambda_K}$;

and $\boldsymbol{\alpha}_{\Lambda_K}$ is a K-dimensional vector consisting of entries of $\boldsymbol{\alpha}$ indexed by Λ_K.

In this paper, the dictionary $\boldsymbol{A}$ consists of training samples collected directly from the image of interest. However, a more robust dictionary can be designed by dictionary-learning techniques [44], [45], or by creating dedicated subspaces for each class through principal component analysis [46], [47]. Next, we show how to obtain $\boldsymbol{\alpha}$ and how to classify a test sample from $\boldsymbol{\alpha}$.

B. Reconstruction and Classification

We first consider the reconstruction problem of finding the sparse vector $\boldsymbol{\alpha}$ for a test sample $\boldsymbol{x}$. Given the dictionary of training samples $\boldsymbol{A}$, the representation $\boldsymbol{\alpha}$ satisfying $\boldsymbol{A}\boldsymbol{\alpha} = \boldsymbol{x}$ is obtained by solving the following optimization problem:

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_0 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{\alpha} = \boldsymbol{x}. \tag{4}$$

To account for approximation errors in empirical data, the equality constraint in (4) can be relaxed to an inequality one

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_0 \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{x}\|_2 \leq \sigma \tag{5}$$

where σ is the error tolerance. The aforementioned problem can also be interpreted as minimizing the approximation error within a certain sparsity level

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{x}\|_2 \quad \text{subject to} \quad \|\boldsymbol{\alpha}\|_0 \leq K_0 \tag{6}$$

where K_0 is a given upper bound on the sparsity level [48]. The aforementioned problems are NP-hard, but they can be approximately solved by greedy pursuit algorithms, such as Orthogonal Matching Pursuit (OMP) [49] or Subspace Pursuit (SP) [50]. Both OMP and SP algorithms are used to locate the support of the sparse vector that approximately solves the problem in (6), but the atoms are selected from the dictionary in different ways. The OMP algorithm augments the support set by one index at each iteration until K_0 atoms are selected or the approximation error is within a preset threshold. The SP algorithm maintains a set of K_0 indices. At each iteration, the index set is refined by adding K_0 new candidates to the current list and then discarding K_0 insignificant ones from the list of $2K_0$ candidates. With the backtracking mechanism, SP is able to find the K_0 most significant atoms. The computational complexity is in the order of $\mathcal{O}(BNK_0)$ for OMP and is upper-bounded by $\mathcal{O}(BNK_0)$ for SP.

The NP-hard problem in (4) can also be relaxed to a linear programming problem, called basis pursuit (BP), by replacing $\|\cdot\|_0$ by $\|\cdot\|_1$ [51]

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_1 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{\alpha} = \boldsymbol{x}. \tag{7}$$

Similarly, the problems in (5) and (6) can also be relaxed to convex programming problems as

$$\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_1 \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{x}\|_2 \leq \sigma \tag{8}$$

$$\text{and} \quad \hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{x}\|_2 \quad \text{subject to} \quad \|\boldsymbol{\alpha}\|_1 \leq \tau \tag{9}$$

respectively. The aforementioned problems can be solved efficiently by an Interior Point method [52] or Gradient Projection method [53], [54] in polynomial time. In this paper, we have employed the OMP and SP greedy algorithms to solve the sparsity-constrained optimization problem in (6) and the SPG-L1 optimization toolbox [54] to solve the BP denoising problem in (8).

The class of $\boldsymbol{x}$ can be determined directly by the characteristics of the recovered sparse vector $\hat{\boldsymbol{\alpha}}$. Define the mth residual (i.e., error between the test sample and the reconstruction from training samples in the mth class) to be

$$r^m(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{A}^m\hat{\boldsymbol{\alpha}}^m\|_2, \quad m = 1, 2, \ldots, M \tag{10}$$

where $\hat{\boldsymbol{\alpha}}^m$ denotes the portion of the recovered sparse coefficients corresponding to the training samples in the mth class. The class of $\boldsymbol{x}$ is then determined as the one with the minimal residual

$$\text{Class}(\boldsymbol{x}) = \arg \min_{m=1,\ldots,M} r^m(\boldsymbol{x}). \tag{11}$$

III. Sparse Representation With Contextual Information

Neighboring hyperspectral pixels usually consist of similar materials; and thus, their spectral characteristics are highly correlated. Previous work has shown that by taking into account this interpixel correlation in HSI, the classification accuracy can be significantly improved [29], [31]. Therefore, it is necessary to incorporate the contextual information into the proposed sparsity model as well. In this section, we show two different approaches to achieve this within the sparsity-constrained problem formulation.

A. Laplacian Constraint

Let $\boldsymbol{I}$ represent the hyperspectral image. Let $\boldsymbol{x}_1$ be a pixel of interest in $\boldsymbol{I}$, $\boldsymbol{x}_i, i = 2, \ldots, 5$ be its four nearest neighbors in the spatial domain, as shown in Fig. 1, and $\boldsymbol{\alpha}_i$ be the sparse vector associated with $\boldsymbol{x}_i$ (i.e., $\boldsymbol{x}_i = \boldsymbol{A}\boldsymbol{\alpha}_i$). Define vector Laplacian at the reconstructed point $\hat{\boldsymbol{x}}_\mathbf{1}$ to be the B-dimensional vector

$$\begin{aligned}\nabla^2(\hat{\boldsymbol{x}}_1) &= 4\hat{\boldsymbol{x}}_1 - \hat{\boldsymbol{x}}_2 - \hat{\boldsymbol{x}}_3 - \hat{\boldsymbol{x}}_4 - \hat{\boldsymbol{x}}_5 \\ &= \boldsymbol{A}(4\hat{\boldsymbol{\alpha}}_1 - \hat{\boldsymbol{\alpha}}_2 - \hat{\boldsymbol{\alpha}}_3 - \hat{\boldsymbol{\alpha}}_4 - \hat{\boldsymbol{\alpha}}_5).\end{aligned} \tag{12}$$

In order to incorporate the smoothness across the neighboring spectral pixels, we propose to force the vector Laplacian at $\hat{\boldsymbol{x}}_\mathbf{1}$ to become zero in addition to the constraints on sparsity level and reconstruction accuracy in (4). In this way, the reconstructed test sample $\hat{\boldsymbol{x}}_1$ is forced to have a similar spectral characteristics to its four nearest neighbors. The new sparse recovery problem with the smoothing constraint is formulated as

$$\begin{aligned}\text{minimize} \quad & \sum_{i=1}^{5} \|\boldsymbol{\alpha}_i\|_0 \\ \text{subject to :} \quad & \boldsymbol{A}(4\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_3 - \boldsymbol{\alpha}_4 - \boldsymbol{\alpha}_5) = \mathbf{0} \\ & \boldsymbol{x}_i = \boldsymbol{A}\boldsymbol{\alpha}_i, \quad i = 1, \ldots, 5.\end{aligned} \tag{13}$$

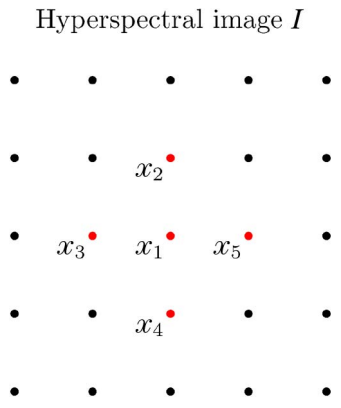

Fig. 1. Four nearest neighbors of a pixel $\boldsymbol{x}_1$.

In (13), the first set of linear constraints forces the reconstructed image vector Laplacian to become zero such that the reconstructed neighboring pixels have similar spectral characteristics, and the second set minimizes reconstruction errors. The optimization problem in (13) can be rewritten as

$$\begin{aligned} &\text{minimize} && \|\tilde{\boldsymbol{\alpha}}\|_0 \\ &\text{subject to :} && \tilde{\boldsymbol{A}}\tilde{\boldsymbol{\alpha}} = \tilde{\boldsymbol{x}} \end{aligned} \tag{14}$$

where

$$\tilde{\boldsymbol{A}} = \begin{bmatrix} 4\lambda\boldsymbol{A} & -\lambda\boldsymbol{A} & -\lambda\boldsymbol{A} & -\lambda\boldsymbol{A} & -\lambda\boldsymbol{A} \\ \boldsymbol{A} & & & & \\ & \boldsymbol{A} & & & \\ & & \boldsymbol{A} & & \\ & & & \boldsymbol{A} & \\ & & & & \boldsymbol{A} \end{bmatrix}$$

$$\tilde{\boldsymbol{\alpha}} = \begin{bmatrix} \boldsymbol{\alpha}_1 \\ \vdots \\ \boldsymbol{\alpha}_5 \end{bmatrix} \quad \tilde{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_5 \end{bmatrix}$$

and $\lambda > 0$ is a weighting factor that controls the relative importance between the smoothing constraint and the reconstruction accuracy.

In practice, the equality constraints in (13) (or equivalently in (14)) cannot be satisfied. Similar to the previous case in Section II-B, the problem in (14) can be relaxed to allow one for smoothing and approximation errors in the form of (5) or (6) as

$$\hat{\tilde{\boldsymbol{\alpha}}} = \arg\min \|\tilde{\boldsymbol{\alpha}}\|_0 \quad \text{subject to} \quad \|\tilde{\boldsymbol{A}}\tilde{\boldsymbol{\alpha}} - \tilde{\boldsymbol{x}}\|_2 \leq \sigma \tag{15}$$

$$\text{or} \quad \hat{\tilde{\boldsymbol{\alpha}}} = \arg\min \|\tilde{\boldsymbol{A}}\tilde{\boldsymbol{\alpha}} - \tilde{\boldsymbol{x}}\|_2 \quad \text{subject to} \quad \|\tilde{\boldsymbol{\alpha}}\|_0 \leq K_0 \tag{16}$$

respectively. The aforementioned problems are standard sparse recovery problems and can be solved using the aforementioned optimization techniques. In this paper, we implemented OMP and SP to solve the problem in (16).

Once the sparse recovery problem in (16) is obtained, the total error residuals between the original test samples and the approximations obtained from each of the M class subdictionaries are calculated as

$$r^m(\boldsymbol{x}) = \sqrt{\sum_{i=1}^{5} \|\boldsymbol{x}_i - \boldsymbol{A}^m \hat{\boldsymbol{\alpha}}_i^m\|_2^2}, \quad m = 1, 2, \ldots, M \tag{17}$$

where $\boldsymbol{x}$ represents a concatenation of the five pixels $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_5$ in the four-connected neighborhood centered at $\boldsymbol{x}_1$; and $\hat{\boldsymbol{\alpha}}_i^m$ denotes the portion of the recovered sparse vector for $\boldsymbol{x}_i$ associated with the mth-class subdictionary $\boldsymbol{A}^m$. The label of the center pixel $\boldsymbol{x}_1$ is then determined to be the class that yields the minimal total residuals

$$\text{Class}(\boldsymbol{x}_1) = \arg \min_{m=1,\ldots,M} r^m(\boldsymbol{x}). \tag{18}$$

B. Joint Sparsity Model

An alternative way to exploit the spatial correlation across neighboring pixels is through a joint sparsity model [42], [43]—assuming that the underlying sparse vectors associated with the neighboring pixels share a common sparsity pattern. That is, HSI pixels in a small spatial neighborhood are approximated by a sparse linear combination of a few common atoms from a given structured dictionary, but these atoms are weighted with a different set of coefficients for each pixel.

To illustrate the joint sparsity model, consider two neighboring hyperspectral pixels $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ consisting of similar materials. The sparse representation of $\boldsymbol{x}_i$ with respect to a given $B \times N$ structured dictionary $\boldsymbol{A}$ can be written as

$$\boldsymbol{x}_i = \boldsymbol{A}\boldsymbol{\alpha}_i = \alpha_{i,\lambda_1}\boldsymbol{a}_{\lambda_1} + \alpha_{i,\lambda_2}\boldsymbol{a}_{\lambda_2} + \ldots + \alpha_{i,\lambda_K}\boldsymbol{a}_{\lambda_K}$$

where the index set $\Lambda_K = \{\lambda_1, \lambda_2, \ldots, \lambda_K\}$ is the support of the sparse vector $\boldsymbol{\alpha}_i$. It is assumed that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ consist of similar materials. Therefore, $\boldsymbol{x}_j$ can also be approximated by the same set of training samples $\{\boldsymbol{a}_k\}_{k\in\Lambda_K}$, but with a different set of coefficients $\{\alpha_{j,k}\}_{k\in\Lambda_K}$

$$\boldsymbol{x}_j = \boldsymbol{A}\boldsymbol{\alpha}_j = \alpha_{j,\lambda_1}\boldsymbol{a}_{\lambda_1} + \alpha_{j,\lambda_2}\boldsymbol{a}_{\lambda_2} + \ldots + \alpha_{j,\lambda_K}\boldsymbol{a}_{\lambda_K}.$$

This can be extended to pixels in a small neighborhood $\mathcal{N}_\epsilon$ consisting of T pixels. Let $\boldsymbol{X} = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \ldots\ \boldsymbol{x}_T]$ be a $B \times T$ matrix, where the columns $\{\boldsymbol{x}_t\}_{t=1,\ldots,T} \in \mathcal{N}_\epsilon$ are pixels in a spatial neighborhood in the hyperspectral image. Now, using the joint sparsity model, $\boldsymbol{X}$ can be represented by

$$\begin{aligned} \boldsymbol{X} &= [\boldsymbol{x}_1 \quad \boldsymbol{x}_2 \quad \ldots \quad \boldsymbol{x}_T] = [\boldsymbol{A}\boldsymbol{\alpha}_1 \quad \boldsymbol{A}\boldsymbol{\alpha}_2 \quad \ldots \quad \boldsymbol{A}\boldsymbol{\alpha}_T] \\ &= \boldsymbol{A}\underbrace{[\boldsymbol{\alpha}_1 \quad \boldsymbol{\alpha}_2 \quad \ldots \quad \boldsymbol{\alpha}_T]}_{\boldsymbol{S}} = \boldsymbol{A}\boldsymbol{S}. \end{aligned} \tag{19}$$

The sparse vectors $\{\boldsymbol{\alpha}_t\}_{t=1,\ldots,T}$ share the same support Λ_K; and thus, $\boldsymbol{S} \in \mathbb{R}^{N\times T}$ is a sparse matrix with only K nonzero rows. For convenience, we call the support Λ_K of $\boldsymbol{\alpha}_t$ as also the support of the *row-sparse* matrix $\boldsymbol{S}$.

Given the training dictionary $\boldsymbol{A}$, the matrix $\boldsymbol{S}$ can be recovered by solving the following joint sparse recovery problem

$$\begin{aligned} &\text{minimize} && \|\boldsymbol{S}\|_{\text{row},0} \\ &\text{subject to :} && \boldsymbol{A}\boldsymbol{S} = \boldsymbol{X} \end{aligned} \tag{20}$$

where the notation $\|\boldsymbol{S}\|_{\text{row},0}$ denotes the number of nonzero rows of $\boldsymbol{S}$ (also called the diversity of $\boldsymbol{S}$ as in [43]). The solution to the earlier problem $\hat{\boldsymbol{S}} = [\hat{\boldsymbol{\alpha}}_1\ \hat{\boldsymbol{\alpha}}_2\ \ldots\ \hat{\boldsymbol{\alpha}}_T]$ is an $N \times T$ sparse matrix with only few nonzero rows. For empirical data,

the problem in (20) can also be rewritten to account for the approximation errors [55], as it was done in (5) and (6)

$$\hat{\boldsymbol{S}} = \arg\min \|\boldsymbol{S}\|_{\text{row},0} \quad \text{subject to} \quad \|\boldsymbol{AS} - \boldsymbol{X}\|_F \leq \sigma \quad (21)$$

or

$$\hat{\boldsymbol{S}} = \arg\min \|\boldsymbol{AS} - \boldsymbol{X}\|_F \quad \text{subject to} \quad \|\boldsymbol{S}\|_{\text{row},0} \leq K_0 \quad (22)$$

where $\|\cdot\|_F$ denotes the Frobenius norm. Similar to the pixel-wise sparse recovery problems, the simultaneous sparse recovery problems (20)–(22) are NP-hard problems, which can be approximately solved by greedy algorithms [42], [43], or relaxed to convex programming which can be solved in polynomial time [56], [57].

In this paper, two greedy pursuit algorithms are used to approximately solve the problem in (22). The first one is a generalized OMP algorithm, called Simultaneous OMP (SOMP) [42], which is summarized in Algorithm 1. In SOMP, the support of the solution is sequentially updated (i.e., the atoms in the dictionary $\boldsymbol{A}$ are sequentially selected). At each iteration, the atom that simultaneously yields the best approximation to all of the residual vectors is selected. Particularly, at the kth iteration, we calculate an $N \times T$ correlation matrix $\boldsymbol{C} = \boldsymbol{A}^T \boldsymbol{R}_{k-1}$, where $\boldsymbol{R}_{k-1}$ is the residual between the data matrix $\boldsymbol{X}$ and its approximation. The (i,t)th entry in $\boldsymbol{C}$ is the correlation between the ith dictionary atom $\boldsymbol{a}_i$ and the residual vector for $\boldsymbol{x}_t$ at the current iteration k. In SOMP, we need to compute the ℓ_p-norm for some $p \geq 1$ for each of the N rows of $\boldsymbol{C}$ [Step (1) in Algorithm 1]. The row index corresponding to the largest ℓ_p-norm is then selected to augment the support set. In the literature, $p = 1$ is used in [42], and $p = 2$ is used in [43]. In [58], $p = 2$ and $p = \infty$ are also proposed for weak matching pursuit. The SOMP algorithm usually terminates when the residual is sufficiently small or the desired level of sparsity (controlled by the number of iterations) is achieved, corresponding to the problem in (21) or (22), respectively. The implementation details of SOMP is summarized in Algorithm 1. It should be noted that the normalization of samples is not a requirement in SOMP and SSP. Normalization is implicitly implemented by these greedy algorithms since the atoms are selected from the training dictionary by maximal correlation, regardless of the magnitude of the atoms.

Algorithm 1: SOMP

Input: $B \times N$ dictionary $\boldsymbol{A} = [\boldsymbol{a}_1\ \boldsymbol{a}_2\ \ldots\ \boldsymbol{a}_N]$, $B \times T$ data matrix $\boldsymbol{X} = [\boldsymbol{x}_1\ \boldsymbol{x}_2\ \ldots\ \boldsymbol{x}_T]$, a stopping criterion {Make sure all columns in $\boldsymbol{A}$ and $\boldsymbol{X}$ have unit norm}

Initialization: residual $\boldsymbol{R}_0 = \boldsymbol{X}$, index set $\Lambda_0 = \emptyset$, iteration counter $k = 1$

while stopping criterion has not been met **do**

(1) Find the index of the atom that best approximates all residuals: $\lambda_k = \arg\max_{i=1,\ldots,N} \|\boldsymbol{R}_{k-1}^T \boldsymbol{a}_i\|_p$, $p \geq 1$

(2) Update the index set $\Lambda_k = \Lambda_{k-1} \bigcup \{\lambda_k\}$

(3) Compute $\boldsymbol{P}_k = (\boldsymbol{A}_{\Lambda_k}^T \boldsymbol{A}_{\Lambda_k})^{-1} \boldsymbol{A}_{\Lambda_k}^T \boldsymbol{X} \in \mathbb{R}^{k \times T}$, $\boldsymbol{A}_{\Lambda_k} \in \mathbb{R}^{B \times k}$ consists of the k atoms in $\boldsymbol{A}$ indexed in Λ_k

(4) Determine the residual $\boldsymbol{R}_k = \boldsymbol{X} - \boldsymbol{A}_{\Lambda_k} \boldsymbol{P}_k$

(5) $k \leftarrow k + 1$

end while

Output: Index set $\Lambda = \Lambda_{k-1}$, the sparse representation $\hat{\boldsymbol{S}}$ whose nonzero rows indexed by Λ are the K rows of the matrix $(\boldsymbol{A}_\Lambda^T \boldsymbol{A}_\Lambda)^{-1} \boldsymbol{A}_\Lambda^T \boldsymbol{X}$

TABLE I
16 GROUND-TRUTH CLASSES IN AVIRIS INDIAN PINES AND THE TRAINING AND TEST SETS FOR EACH CLASS

Class		Samples	
No	Name	Train	Test
1	Alfalfa	6	48
2	Corn-notill	144	1290
3	Corn-min	84	750
4	Corn	24	210
5	Grass/Pasture	50	447
6	Grass/Trees	75	672
7	Grass/Pasture-mowed	3	23
8	Hay-windrowed	49	440
9	Oats	2	18
10	Soybeans-notill	97	871
11	Soybeans-min	247	2221
12	Soybean-clean	62	552
13	Wheat	22	190
14	Woods	130	1164
15	Building-Grass-Trees-Drives	38	342
16	Stone-steel Towers	10	85
Total		1043	9323

The second simultaneous sparse recovery algorithm is our proposed simultaneous version of the SP algorithm (SSP), summarized in Algorithm 2. Similar to the pixel-wise SP algorithm [50], SSP also maintains a list of K_0 candidates. At each iteration, the K_0 atoms that yield the best simultaneous approximation to all of the T residual vectors are selected as the new candidates. Similar to SOMP, we also need to compute the ℓ_p-norm for some $p \geq 1$ for each of the N rows of the correlation matrix $\boldsymbol{C} = \boldsymbol{A}^T \boldsymbol{R}_{k-1}$ [Step (1) in Algorithm 2]. The row indices corresponding to the rows with the K_0 largest ℓ_p-norm are then selected as the new candidates to augment the support set. A backtracking step is then implemented, where the K_0 significant atoms are selected in a similar fashion from the $2K_0$-candidate list [Step (4) in Algorithm 2]. The SSP algorithm terminates when the residual begins to increase, or a maximum number of iterations is reached.

Algorithm 2: Simultaneous Subspace Pursuit (SSP)

Input: $B \times N$ dictionary $\boldsymbol{A} = [\boldsymbol{a}_1\ \boldsymbol{a}_2\ \ldots\ \boldsymbol{a}_N]$, $B \times T$ data matrix $\boldsymbol{X} = \boldsymbol{x}_1\ \boldsymbol{x}_2\ \ldots\ \boldsymbol{x}_T$, sparsity level K_0, a stopping criterion {Make sure all columns in $\boldsymbol{A}$ and $\boldsymbol{X}$ have unit norm}

Initialization: index set $\Lambda_0 = \{K_0$ indices corresponding to the K_0 largest numbers in $\|\boldsymbol{X}^T \boldsymbol{a}_i\|_p, p \geq 1, i = 1, \ldots, N\}$, residual $\boldsymbol{R}_0 = \boldsymbol{X} - \boldsymbol{A}_{\Lambda_0} (\boldsymbol{A}_{\Lambda_0}^T \boldsymbol{A}_{\Lambda_0})^{-1} \boldsymbol{A}_{\Lambda_0}^T \boldsymbol{X}$, iteration counter $k = 1$

while stopping criterion has not been met **do**

(1) Find the indices of the K_0 atoms that best approximate all residuals: $\mathcal{I} = \{K_0$ indices corresponding

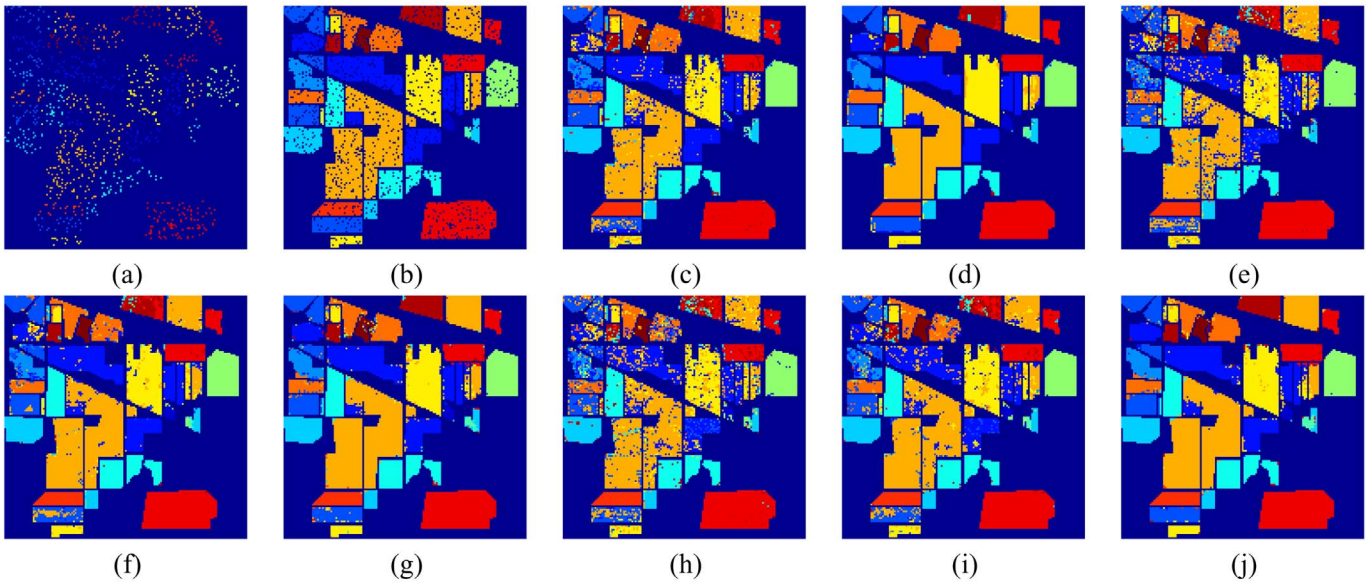

Fig. 2. For the Indian Pines image: (a) training set and (b) test set. Classification maps obtained by (c) SVM, (d) SVM-CK, (e) SP, (f) SP-S, (g) SSP, (h) OMP, (i) OMP-S, and (j) SOMP.

TABLE II
CLASSIFICATION ACCURACY (%) FOR THE INDIAN PINES IMAGE ON THE TEST SET

Class	SVM	SVM-CK	SP	SP-S	SSP	OMP	OMP-S	SOMP	ℓ_1
1	81.25	**95.83**	68.75	87.50	89.58	68.75	70.83	85.42	39.58
2	86.28	**96.67**	74.65	91.94	95.04	65.97	79.22	94.88	78.53
3	72.80	90.93	63.20	82.53	92.93	60.67	76.67	**94.93**	51.87
4	58.10	85.71	40.00	70.95	85.24	38.57	55.24	**91.43**	28.57
5	92.39	93.74	89.04	94.41	92.17	89.49	**95.30**	89.49	80.76
6	96.88	97.32	95.98	99.26	98.81	95.24	98.96	98.51	**99.40**
7	43.48	69.57	21.74	47.83	73.91	21.74	52.17	**91.30**	17.39
8	98.86	98.41	99.09	**99.77**	99.55	97.05	**99.77**	99.55	99.32
9	50.00	55.56	61.11	**94.44**	0	33.33	72.22	0	16.67
10	71.53	**93.80**	70.72	86.80	88.98	68.20	82.32	89.44	63.95
11	84.38	94.37	77.94	93.38	**97.34**	75.96	88.79	**97.34**	86.04
12	85.51	**93.66**	61.23	84.24	86.59	54.53	73.73	88.22	57.79
13	**100**	99.47	**100**	**100**	99.47	**100**	98.95	**100**	**100**
14	93.30	**99.14**	95.62	98.28	98.88	92.87	97.25	**99.14**	97.94
15	64.91	87.43	48.25	69.30	97.37	41.23	49.71	**99.12**	35.96
16	88.24	**100**	92.94	95.29	85.88	94.12	**100**	96.47	90.59
Overall	84.52	94.86	78.10	91.16	94.79	74.78	85.52	**95.28**	77.99
Average	79.24	**90.73**	72.52	87.25	86.36	68.61	80.70	88.45	65.27
κ	0.823	0.941	0.749	0.899	0.940	0.712	0.834	**0.946**	0.746

to the K_0 largest numbers in $\|\boldsymbol{R}_{k-1}^T \boldsymbol{a}_i\|_p, p \geq 1, i = 1, \ldots, N\}$

(2) Update the candidate index set $\tilde{\Lambda}_k = \Lambda_{k-1} \bigcup \mathcal{I}$

(3) Compute $\boldsymbol{P}_k = \left(\boldsymbol{A}_{\tilde{\Lambda}_k}^T \boldsymbol{A}_{\tilde{\Lambda}_k}\right)^{-1} \boldsymbol{A}_{\tilde{\Lambda}_k}^T \boldsymbol{X} \in \mathbb{R}^{2K_0 \times T}$

(4) Let $\boldsymbol{p}_k^i$ denote the ith row in $\boldsymbol{P}_k$. Update the index set $\Lambda_k = \{K_0$ indices corresponding to the K_0 largest numbers in $\|(\boldsymbol{p}_k^i)^T\|_p, p \geq 1, i = 1, \ldots, 2K_0\}$

(5) Determine the residual

$$\boldsymbol{R}_k = \boldsymbol{X} - \boldsymbol{A}_{\Lambda_k} \left(\boldsymbol{A}_{\Lambda_k}^T \boldsymbol{A}_{\Lambda_k}\right)^{-1} \boldsymbol{A}_{\Lambda_k}^T \boldsymbol{X}$$

(6) $k \leftarrow k + 1$

end while

Output: Index set $\Lambda = \Lambda_{k-1}$, the sparse representation $\hat{\boldsymbol{S}}$ whose K_0 nonzero rows indexed by Λ are the rows of the matrix $\left(\boldsymbol{A}_\Lambda^T \boldsymbol{A}_\Lambda\right)^{-1} \boldsymbol{A}_\Lambda^T \boldsymbol{X}$

Once the row-sparse matrix $\hat{\boldsymbol{S}}$ in (20) is obtained, we calculate the error residuals between the original test samples and the approximations obtained from each class subdictionaries as follows:

$$r^m(\boldsymbol{X}) = \|\boldsymbol{X} - \boldsymbol{A}^m \hat{\boldsymbol{S}}^m\|_F, \qquad m = 1, 2, \ldots, M \tag{23}$$

where $\hat{\boldsymbol{S}}^m$ consists of the N_m rows in $\hat{\boldsymbol{S}}$ that are associated with the mth-class subdictionary $\boldsymbol{A}^m$. The label of the center pixel $\boldsymbol{x}_1$ is then determined by the minimal total residual as it was done previously in Section II-B

$$\text{Class}(\boldsymbol{x}_1) = \arg \min_{m=1,\ldots,M} r^m(\boldsymbol{X}). \tag{24}$$

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In this section, we demonstrate the effectiveness of the proposed sparsity-based classification algorithms on three

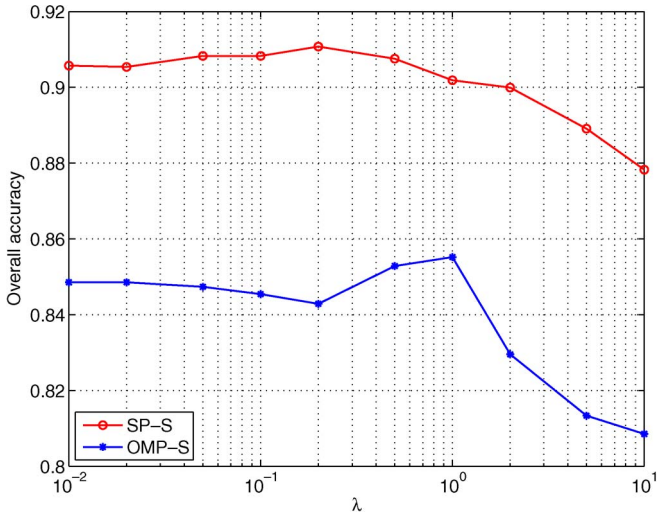

Fig. 3. Effect of the weighting factor λ of the Laplacian constraint for Indian Pines.

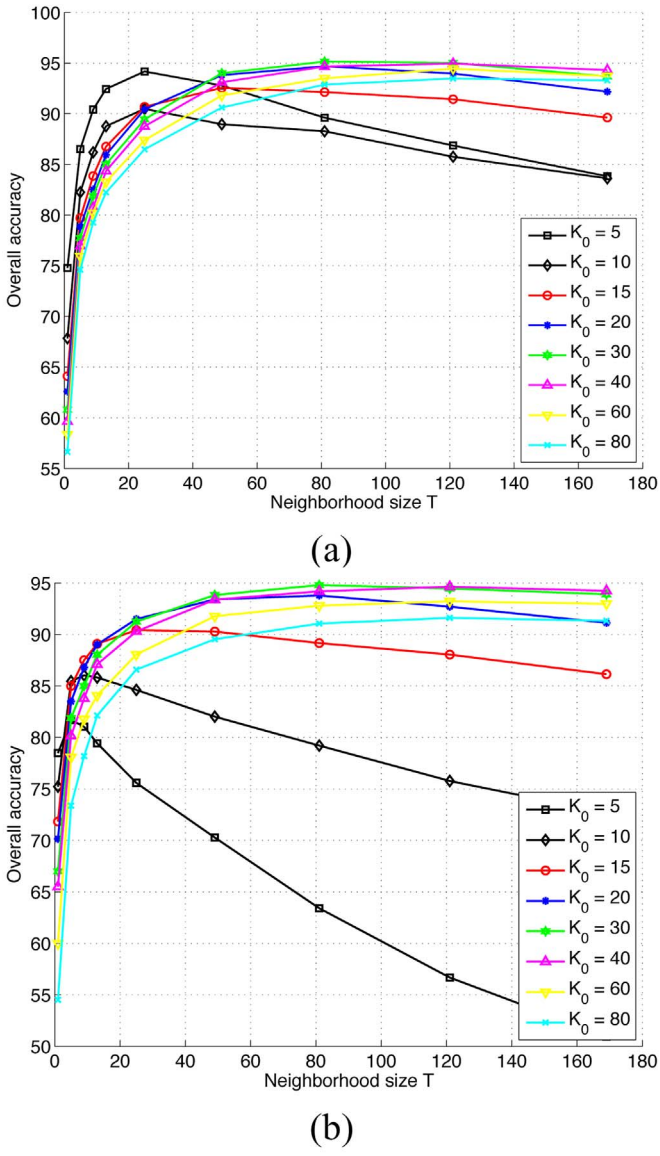

Fig. 4. Effect of the sparsity level K_0 and size of neighborhood T for Indian Pines. (a) SOMP and (b) SSP.

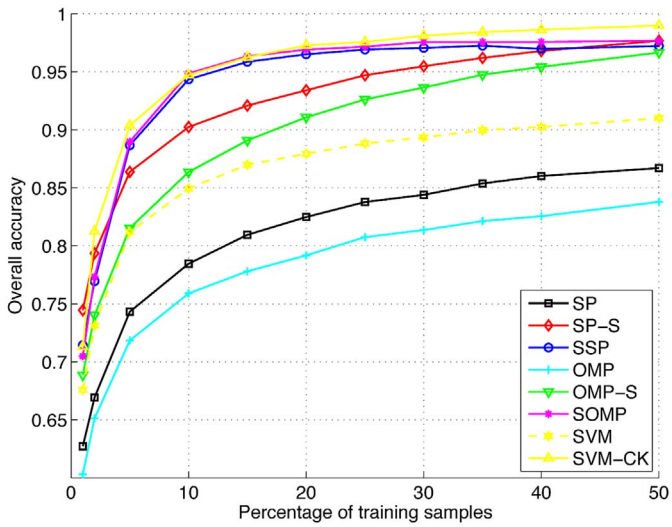

Fig. 5. Effect of the number of training samples for Indian Pines.

TABLE III
NINE CLASSES IN UNIVERSITY OF PAVIA AND THE TRAINING AND TEST SETS

Class		Samples	
No	Name	Train	Test
1	Asphalt	548	6304
2	Meadows	540	18146
3	Gravel	392	1815
4	Trees	524	2912
5	Metal sheets	265	1113
6	Bare soil	532	4572
7	Bitumen	375	981
8	Bricks	514	3364
9	Shadows	231	795
Total		3921	40002

hyperspectral images. For each image, we solve the sparse recovery problems in (4), (14), and (20) for each test sample and then determine the class by the minimal residual. The classification results are then compared visually and quantitatively to those obtained by the classical classifier SVMs, which has proven a very powerful tool to solve supervised classification problems and have shown good performances in hyperspectral classification [26], [30].

The SP [50] and OMP algorithms [49] are used to approximately solve the sparsity-constrained problems in (6) and (16), and the results are denoted as SP, OMP, SP-S (SP with smoothing), and OMP-S (OMP with smoothing). For comparison, the classification performance using the SPG-L1 package [54], which solves the linearized sparse recovery problem in (7), is also included. Algorithms 1 (SOMP) and 2 (SSP) are implemented to approximately solve the simultaneous sparse recovery problem in (22). For SVM, we use a composite kernel (denoted by SVM-CK) that combines the spectral and spatial information via a weighted kernel summation, which has been shown to outperform the spectral-only SVM in HSI classification [30], [31]. The SVM parameters (RBF-kernel parameter γ, regularization parameter C, and weight μ for composite kernels) are obtained by cross-validation. Details of these parameters are explained in [26], [30]. The one-against-one strategy is employed for M-class classification using SVM

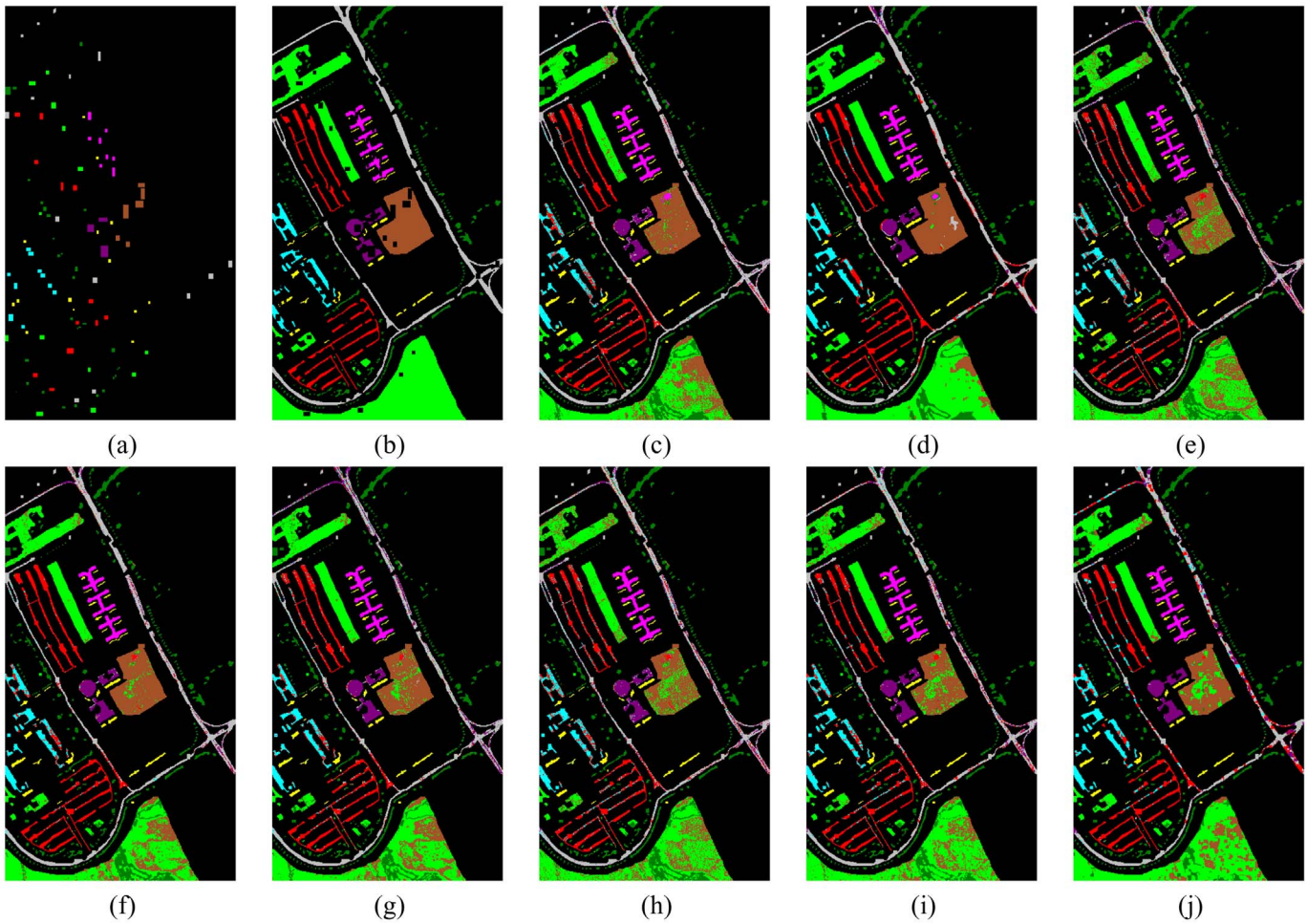

Fig. 6. For the University of Pavia image: (a) training set and (b) test set. Classification maps obtained by (c) SVM, (d) SVM-CK, (e) SP, (f) SP-S, (g) SSP, (h) OMP, (i) OMP-S, and (j) SOMP.

and SVM-CK. That is, a total of $M(M-1)/2$ binary classifiers are constructed for each pair of classes in the training stage. These classifiers are then applied to each test sample and vote for the winning class. The class label of the test sample is determined by majority voting. The sparsity-based algorithms, unlike SVM, do not involve a training stage. They search for dedicated atoms in the training dictionary for each test pixel (i.e., the support of the sparse vector is dynamic). Therefore, the sparsity-based algorithms are more computationally intensive than SVM.

A. AVIRIS Data Set: Indian Pines

The first hyperspectral image in our experiments is the commonly-used Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) Indian Pines image [59]. The AVIRIS sensor generates 220 bands across the spectral range from 0.2 to 2.4 μm. In the experiments, the number of bands is reduced to 200 by removing 20 water absorption bands [25]. This image has spatial resolution of 20 m per pixel and spatial dimension 145×145. It contains 16 ground-truth classes, most of which are different types of crops (e.g., corns, soybeans, and wheats), as seen in Table I. For each of the 16 class, we randomly choose around 10% of the labeled samples for training and use the rest 90% for testing. The number of training and test samples for each class is shown in Table I and the training, and test sets are visually shown in Fig. 2(a) and (b), respectively.

The classification accuracy for each class, the overall accuracy, average accuracy, and the κ coefficient measure [60] are shown in Table II using different classifiers on the test set. The overall accuracy is computed by the ratio between correctly classified test samples and the total number of test samples, and the average accuracy is the mean of the 16 class accuracies. The κ coefficient is computed by weighting the measured accuracies. It incorporates both of the diagonal and off-diagonal entries of the confusion matrix and is a robust measure of the degree of agreement.

The parameters for SVM and SVM-CK are obtained by fivefold cross-validation. For the sparsity-based algorithms, the sparsity level K_0 is chosen between $K_0 = 5$ to $K_0 = 30$. For SP-S and OMP-S, the weighting factor λ is fixed to 1. Since the Indian Pines image consists of large homogenous regions, a large window of size 9×9 $(T = 81)$ is used in SSP and SOMP. For pixelwise algorithms, the ℓ_1-relaxation technique (7) yields comparable results to SP. In most cases, the proposed

TABLE IV
CLASSIFICATION ACCURACY (%) FOR UNIVERSITY OF PAVIA ON THE TEST SET USING DIFFERENT CLASSIFIERS

Class	SVM	SVM-CK	SP	SP-S	SSP	OMP	OMP-S	SOMP	ℓ_1
1	**84.30**	79.85	69.70	83.79	69.59	68.07	72.53	59.33	80.65
2	67.01	**84.86**	67.69	72.35	72.31	67.07	73.12	78.15	64.74
3	68.43	81.87	67.11	71.85	74.10	65.45	75.87	**83.53**	73.22
4	97.80	96.36	97.46	**98.94**	95.33	97.32	98.18	96.91	98.35
5	99.37	99.37	99.82	**100**	99.73	99.73	**100**	99.46	99.91
6	92.45	**93.55**	75.79	92.63	86.72	73.29	77.69	77.41	92.54
7	89.91	90.21	88.79	91.44	90.32	87.26	94.90	**98.57**	86.95
8	92.42	92.81	84.96	**95.57**	90.46	81.90	87.60	89.09	81.54
9	97.23	95.35	93.58	98.24	90.94	94.72	97.48	91.95	**98.99**
Overall	79.15	**87.18**	74.45	82.09	78.39	73.27	78.48	79.00	76.87
Average	87.66	**90.47**	82.77	89.42	85.50	81.65	86.38	86.04	86.32
κ	0.737	**0.833**	0.675	0.772	0.724	0.661	0.724	0.728	0.709

sparsity-based algorithm with spatial information outperforms the classical SVM and overall the SOMP algorithm provides the best performance. However, both SSP and SOMP fail to identify any samples belonging to the ninth class consisting of oats. This is partly due to the lack of training samples (20 ground-truth samples in total, and only two are used for training). Moreover, Oat pixels cover a very narrow region of size 10×2 located in the midleft of the image. In SSP/SOMP, the 9×9 local window centered at each Oat pixel is dominated by the population of the pixels in the two adjacent classes, which are Class 3 of Corn-min (on the left) and Class 6 of Grass/Trees (on the right), where each of them occupies a large homogenous region. By forcing the joint sparsity within this window size, the Oat-covered region is completely smoothed and misclassified as Corn-min and Grass/Trees. More results for this image using SVM and other supervised classifiers can be found in [31].

The classification maps on labeled pixels obtained from the various techniques are shown in Fig. 2(c)–(j). Fig. 2(c) and (d) show the results of the one-against-one multiclass SVM technique using a spectral-only kernel and a composite kernel combining both spectral and spatial information through a weighted summation, respectively. Fig. 2(e)–(j) show the visual results obtained by solving the original ℓ_0-norm minimization problem in (6), the Laplacian-constrained sparse recovery problem in (16) and the simultaneous sparse recovery problem in (22) using the SP-based and OMP-based reconstruction algorithms SP/OMP, SP-S/OMP-S, and SSP/SOMP, respectively. One can see from Fig. 2 that by incorporating the spatial information, the sparsity-based algorithm leads to a much smoother classification map than the pixelwise algorithms.

For the sparsity-based algorithm with the Laplacian constraint, the scalar value λ in matrix $\tilde{\boldsymbol{A}}$ used in (16) controls the relative importance between smoothness and reconstruction accuracy, leading to different classification results. Now we demonstrate how λ affects the classification accuracy. As previously done, 10% of the data are chosen as training samples, and the remaining 90% are used for testing. The sparsity level K_0 is fixed to $K_0 = 10$ for SP-S and $K_0 = 5$ for OMP-S. The weighting factor λ varies from 10^{-2} to 10. The overall accuracy as a function of λ is shown in Fig. 3. For both SP-S and OMP-S, with $\lambda \leq 1$, there is only a slight difference in the classification accuracy. As λ increases (i.e., more weight on smoothness and less weight on approximation accuracy), the classification performance degrades quickly.

TABLE V
NINE GROUND-TRUTH CLASSES IN *Center of Pavia* AND THE TRAINING AND TEST SETS

Class		Samples	
No	Name	Train	Test
1	Water	745	64533
2	Trees	785	5722
3	Meadow	797	2094
4	Brick	485	1667
5	Soil	820	5729
6	Asphalt	678	6847
7	Bitumen	808	6479
8	Tile	223	2899
9	Shadow	195	1970
Total		5536	97940

Next, we demonstrate the effect of the sparsity level K_0 and the neighborhood size T on the performance of the simultaneous sparse approximation algorithms SOMP and SSP. In this experiment, we randomly choose 10% of the data in each class as training samples and use the remaining 90% as test samples. In each test, we apply the SOMP and SSP algorithms with different sparsity level K_0 and neighborhood size T to solve the problem in (22). The sparsity level K_0 ranges from $K_0 = 5$ to $K_0 = 80$, and the neighborhood ranges from the four-connected neighborhood ($T = 5$) to a 13×13 window ($T = 169$). The overall classification accuracy plots on the entire test set for Indian Pines are shown in Fig. 4(a) and (b) for SOMP and SSP, respectively. The horizontal axis indicates the size of the neighborhood T and the vertical axis is the overall accuracy (%). For small sparsity level K_0, if the neighborhood size T is too large, then the neighboring pixels cannot be faithfully approximated by few training samples and the classification accuracy is significantly reduced. On the other hand, as K_0 increases toward the size of the dictionary, the solution converges to the pseudoinverse solution, which is no longer sparse and may involve atoms in multiple subdictionaries, leading to a performance degradation. One can also see that for sufficiently large sparsity level K_0, in general the classification performance increases almost monotonically as T increases. However, large neighborhood may cause oversmoothing over neighboring classes, which would lead to a decrease in the overall classification accuracy.

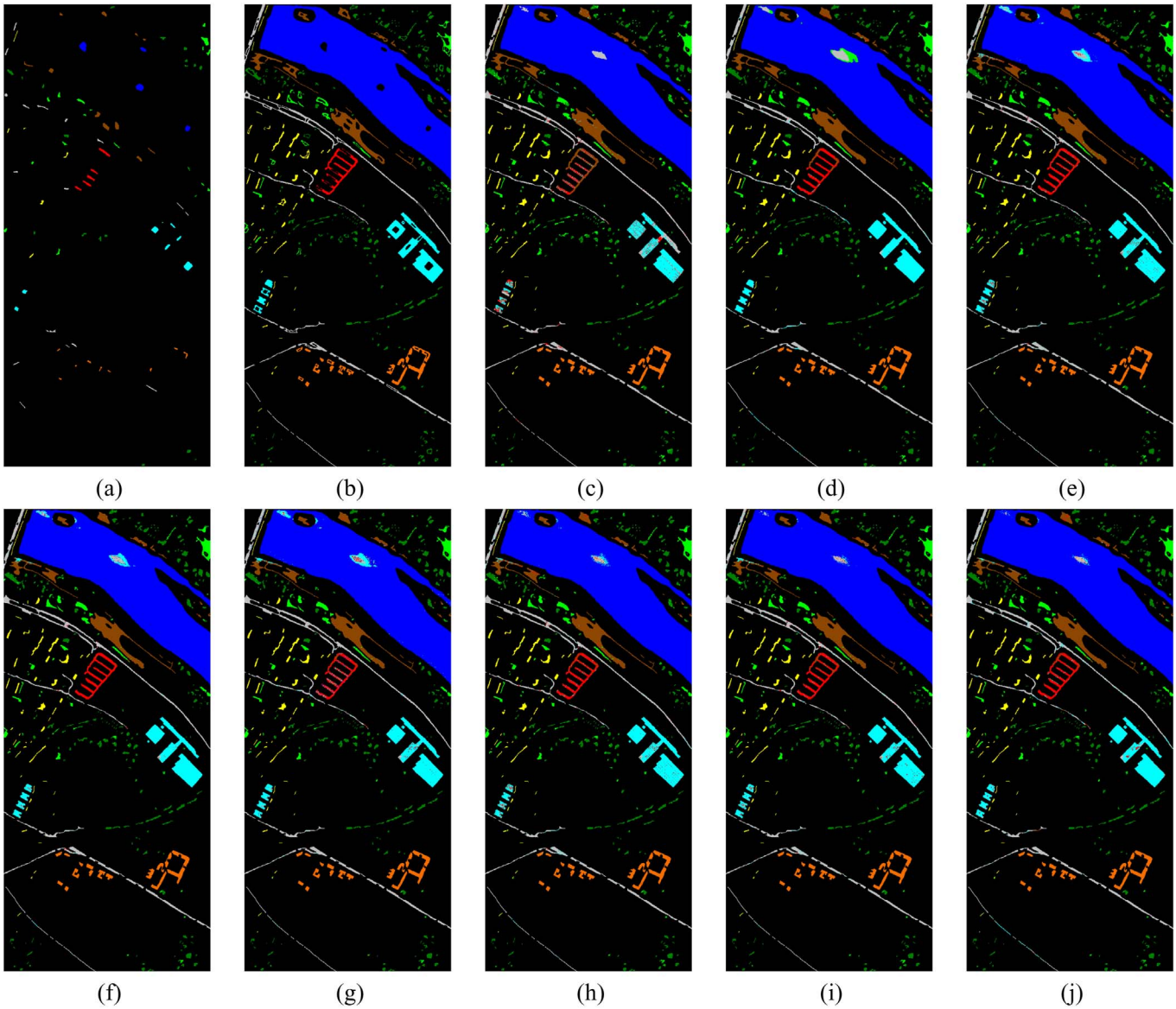

Fig. 7. For Center of Pavia: (a) training set and (b) test set. Classification maps obtained by (c) SVM, (d) SVM-CK, (e) SP, (f) SP-S, (g) SSP, (h) OMP, (i) OMP-S, and (j) SOMP.

Next, we show how the number of training samples affects the classification performance for various algorithms. The parameters are fixed to be the same as those used to generate the maps in Fig. 2. In each test, we randomly choose 1 to 50% of the labeled data in each class as the training samples and the remaining samples as the test ones. The classification accuracy plots under various conditions are shown in Fig. 5, where the x-axis denotes the percentage of training samples in all labeled samples, and the y-axis is the overall classification accuracy on the test set. The accuracy are averaged over ten runs at each percentage to avoid any bias induced by random sampling. It is obvious from Fig. 5 that in most cases the overall accuracy increases monotonically as the percentage of training samples increases. The pixelwise sparsity model in (2) for HSI does not lead to better performance than the standard kernel SVM. However, by incorporating the contextual information via either the explicit smoothing term or the joint sparsity model, the sparsity-based algorithms constantly outperform SVM. Note that in the training stage of SVM-CK, in order to extract the spatial features for each training sample, SVM-CK requires knowledge of the neighboring pixels which may not be available in the training set. Therefore, we could say that SVM-CK is using more training samples than the other methods, especially in our experiment setting where the training sets are randomly selected.

B. ROSIS Urban Data Over Pavia, Italy

The next two hyperspectral images used in our experiments, University of Pavia and Center of Pavia, are urban images acquired by the Reflective Optics System Imaging Spectrometer (ROSIS). The ROSIS sensor generates 115 spectral bands ranging from 0.43 to 0.86 μm and has a spatial resolution of 1.3 m per pixel [31].

TABLE VI
CLASSIFICATION ACCURACY (%) FOR *Center of Pavia* ON THE TEST SET USING DIFFERENT CLASSIFIERS

Class	SVM	SVM-CK	SP	SP-S	SSP	OMP	OMP-S	SOMP	ℓ_1
1	99.19	97.46	98.20	98.21	97.79	98.91	99.32	99.32	**99.35**
2	77.74	**93.08**	86.98	91.63	92.82	86.75	88.17	92.38	83.52
3	86.74	97.09	96.61	97.42	97.80	96.04	**97.99**	95.46	97.09
4	40.38	77.02	84.16	**94.24**	78.52	81.22	83.44	85.66	73.67
5	97.52	**98.39**	94.01	96.79	95.81	94.40	95.95	96.37	95.92
6	94.77	94.32	92.92	96.98	96.52	91.94	93.94	93.81	**97.01**
7	74.37	**97.50**	93.80	97.39	95.96	93.18	95.15	94.68	92.65
8	98.94	99.83	98.79	**99.86**	99.79	98.62	**99.86**	99.69	99.45
9	**100**	99.95	99.34	99.54	98.83	98.07	98.93	98.68	99.09
Overall	94.63	96.81	96.40	97.59	96.93	96.68	97.52	**97.66**	97.13
Average	85.52	94.96	93.87	**96.90**	94.87	93.24	94.74	95.01	93.08
κ	0.899	0.943	0.935	0.957	0.945	0.940	0.955	**0.958**	0.948

The University of Pavia image consists of 610 $\times$ 340 pixels, each having 103 bands with the 12 noisiest bands removed. There are nine ground-truth classes of interests, as shown in Table III. For this image, we follow the same experiment settings for the training and test sets as used in [29], in which about 9% of all labeled data are used as training and the rest are used for testing. Details about the training and test sets are shown in Table III and in Fig. 6(a) and (b), respectively.

The classification results using SVM, SVM-CK, SP, SP-S, SSP, OMP, OMP-S, SOMP, and SPG-L1 are summarized in Table IV. The classification maps on labeled pixels are shown in Fig. 6(c)–(j). For SVM and SVM-CK, we use a one-against-one strategy with RBF kernels, and the parameters are obtained by tenfold cross-validation. The sparsity level for the SP- and OMP-based algorithms is chosen between $K_0 = 5$ and $K_0 = 30$, and the weighting factor λ is fixed to 1 for SP-S and OMP-S. Since this image is obtained from an urban area with small buildings, it lacks the large spatial homogeneity that was present in the Indian Pines image. Thus, a smaller neighborhood (3 $\times$ 3 to 5 $\times$ 5 window) is used in the experiment for SOMP and SSP for joint sparse recovery. The proposed sparsity-based classification algorithms with spatial information achieve better or comparable performance in most cases than the SVM classifier. However, SVM-CK yields the best overall performance and a smoother visual effect.

The third image, Center of Pavia, is the other urban image collected by the ROSIS sensor over the center of Pavia City. This image consists of 1096 $\times$ 492 pixels, each having 102 spectral bands after 13 noisy bands are removed. The nine ground-truth classes and the number of training and test samples for each class are shown in Table V and in Fig. 7(a) and (b), respectively. For this image, about 5% of the labeled data are used as training samples.

The classification results are summarized in Table VI, and the classification maps are shown in Fig. 7(c)–(j). For SVM and SVM-CK, again, the one-against-one strategy is applied and the parameters are optimized by tenfold cross-validation. For all sparsity-based algorithms, the sparsity level is chosen between $K_0 = 5$ to $K_0 = 25$. For SP-S and OMP-S, the weighting factor λ is set to 1. For SSP and SOMP, similar to the case of University of Pavia, we use a small neighborhood of $T = 5$ for SSP and $T = 25$ for SOMP. In this case, all of the sparsity-based algorithms SP-S, SSP, OMP-S, and SOMP outperform SVM and SVM-CK, and SOMP yields the best overall performance.

V. CONCLUSION

In this paper, we have proposed a new algorithm for HSI classification based on sparse representation. In the proposed algorithm, an HSI pixel is assumed to be sparsely represented by a few atoms in a given training dictionary. The sparse representation of a test spectral sample is recovered by solving a sparsity-constrained optimization problem via greedy pursuit algorithms. To improve the classification performance, we propose two different ways to incorporate the contextual information of HSI. One approach is to explicitly include a smoothing term through the vector Laplacian at the reconstructed pixel of interest in the optimization formulation, and the other is through a joint-sparsity model for neighboring pixels centered at the pixel of interest. Experimental results on three hyperspectral images show that the proposed algorithm yields highly accurate classification results, especially for images with large homogeneous areas.

ACKNOWLEDGMENT

The authors would like to thank the University of Pavia and the HySenS project for kindly providing the ROSIS images of University of Pavia and Center of Pavia.

REFERENCES

[1] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*. New York: Academic, Dec. 2008.
[2] J. Wright, Y. Ma, J. Mairal, G. Sapiro, T. Huang, and S. Yan, "Sparse representation for computer vision and pattern recognition," *Proc. IEEE*, vol. 98, no. 6, pp. 1031–1044, Jun. 2010.
[3] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.
[4] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[5] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, 2009.
[6] S. Rao, R. Tron, R. Vidal, and Y. Ma, "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.
[7] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 2790–2797.
[8] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution as sparse representation of raw image patches," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2008, pp. 1–8.

[9] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[10] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[11] J. Wright, A. Y. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[12] J. K. Pillai, V. M. Patel, and R. Chellappa, "Sparsity inspired selection and recognition of iris images," in *Proc. IEEE 3rd Int. Conf. Biometrics: Theory, Appl. Syst.*, Sep. 2009, pp. 1–6.

[13] X. Hang and F.-X. Wu, "Sparse representation for classification of tumors using gene expression data," *J. Biomed. Biotechnol.*, vol. 2009, pp. 1–6, 2009, DOI: 10.1155/2009/403689.

[14] Z. Guo, T. Wittman, and S. Osher, "L1 unmixing and its application to hyperspectral image enhancement," in *Proc. SPIE—Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery XV*, Apr. 2009, vol. 7334, p. 733 41M.

[15] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Sparsity-based classification of hyperspectral imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2010, pp. 2796–2799.

[16] M. Borengasser, W. S. Hungate, and R. Watkins, *Hyperspectral Remote Sensing—Principles and Applications*. Boca Raton, FL: CRC Press, 2008.

[17] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 29–43, Jan. 2002.

[18] D. W. J. Stein, S. G. Beaven, L. E. Hoff, E. M. Winter, A. P. Schaum, and A. D. Stocker, "Anomaly detection from hyperspectral imagery," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 58–69, Jan. 2002.

[19] M. T. Eismann, A. D. Stocker, and N. M. Nasrabadi, "Automated hyperspectral cueing for civilian search and rescue," *Proc. IEEE*, vol. 97, no. 6, pp. 1031–1055, Jun. 2009.

[20] N. K. Patel, C. Patnaik, S. Dutta, A. M. Shekh, and A. J. Dave, "Study of crop growth parameters using airborne imaging spectrometer data," *Int. J. Remote Sens.*, vol. 22, no. 12, pp. 2401–2411, Aug. 2001.

[21] B. Datt, T. R. McVicar, T. G. Van Niel, D. L. B. Jupp, and J. S. Pearlman, "Preprocessing EO-1 hyperion hyperspectral data to support the application of agricultural indexes," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 6, pp. 1246–1259, Jun. 2003.

[22] B. Hörig, F. Kühn, F. Oschütz, and F. Lehmann, "HyMap hyperspectral remote sensing to detect hydrocarbons," *Int. J. Remote Sens.*, vol. 22, no. 8, pp. 1413–1422, 2001.

[23] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. 5th Annu. Workshop Comput. Learn. Theory*, 1992, pp. 144–152.

[24] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

[25] J. A. Gualtieri and R. F. Cromp, "Support vector machines for hyperspectral remote sensing classification," *Proc. SPIE*, vol. 3584, pp. 221–232, Jan. 1998.

[26] F. Melgani and L. Bruzzone, "Classification of hyperspectral remote sensing images with support vector machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 8, pp. 1778–1790, Aug. 2004.

[27] L. Bruzzone, M. Chi, and M. Marconcini, "A novel transductive SVM for the semisupervised classification of remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 44, no. 11, pp. 3363–3373, Nov. 2006.

[28] F. Bovolo, L. Bruzzone, and M. Marconcini, "A novel context-sensitive SVM for classification of remote sensing images," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, Jul. 2006, pp. 2498–2501.

[29] Y. Tarabalka, J. A. Benediktsson, and J. Chanussot, "Spectra-spatial classification of hyperspectral imagery based on partitional clustering techniques," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2973–2987, Aug. 2009.

[30] G. Camps-Valls, L. Gomez-Chova, J. Muñoz-Marí, J. Vila-Francés, and J. Calpe-Maravilla, "Composite kernels for hyperspectral image classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 1, pp. 93–97, Jan. 2006.

[31] A. Plaza, J. A. Benediktsson, J. W. Boardman, J. Brazile, L. Bruzzone, G. Camps-Valls, J. Chanussot, M. Fauvel, P. Gamba, A. Gualtieri, M. Marconcini, J. C. Tilton, and G. Trianni, "Recent advances in techniques for hyperspectral image processing," *Remote Sens. Environ.*, vol. 113, no. Supplement 1, pp. S110–S122, Sep. 2009.

[32] H.-Y. Huang and B.-C. Kuo, "Double nearest proportion feature extraction for hyperspectral-image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4034–4046, Nov. 2010.

[33] H. R. Kalluri, S. Prasad, and L. M. Bruce, "Decision-level fusion of spectral reflectance and derivative information for robust hyperspectral land cover classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4047–4058, Nov. 2010.

[34] B. D. Bue, E. Merényi, and B. Csathó, "Automated labeling of materials in hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4059–4070, Nov. 2010.

[35] B. Demir and S. Ertürk, "Empirical mode decomposition of hyperspectral images for support vector machine classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4071–4084, Nov. 2010.

[36] J. Li, J. M. Bioucas-Dias, and A. Plaza, "Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4085–4098, Nov. 2010.

[37] L. Ma, M. M. Crawford, and J. Tian, "Local manifold learning-based k-nearest-neighbor for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4099–4109, Nov. 2010.

[38] W. Kim and M. M. Crawford, "Adaptive classification for hyperspectral image data using manifold regularization kernel machines," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4110–4121, Nov. 2010.

[39] Y. Tarabalka, J. A. Benediktsson, J. Chanussot, and J. C. Tilton, "Multiple spectral–spatial classification approach for hyperspectral data," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 11, pp. 4122–4132, Nov. 2010.

[40] R. S. Rand and D. M. Keenan, "Spatially smooth partitioning of hyperspectral imagery using spectral/spatial measures of disparity," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 6, pp. 1479–1490, Jun. 2003.

[41] Y. Tarabalka, M. Fauvel, J. Chanussot, and J. A. Benediktsson, "SVM- and MRF-based method for accurate classification of hyperspectral images," *IEEE Geosci. Remote Sens. Lett.*, vol. 7, no. 4, pp. 736–740, Oct. 2010.

[42] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit," *Signal Process.—Special Issue on Sparse Approximations in Signal and Image Processing*, vol. 86, no. 3, pp. 572–588, Mar. 2006.

[43] S. F. Cotter, B. D. Rao, K. Engan, and K. Kreutz-Delgado, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Process.*, vol. 53, no. 7, pp. 2477–2488, Jul. 2005.

[44] M. Aharon, M. Elad, and A. M. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[45] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proc. IEEE*, vol. 98, no. 6, pp. 1045–1057, Jun. 2010.

[46] J. M. Bioucas-Dias and J. M. P. Nascimento, "Hyperspectral subspace identification," *IEEE Trans. Geosci. Remote Sens.*, vol. 46, no. 8, pp. 2435–2445, Aug. 2008.

[47] J.-M. Yang, B.-C. Kuo, P.-T. Yu, and C.-H. Chuang, "A dynamic subspace method for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 48, no. 7, pp. 2840–2853, Jul. 2010.

[48] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proc. IEEE*, vol. 98, no. 6, pp. 948–958, Jun. 2010.

[49] J. Tropp and A. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[50] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[51] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, no. 1, pp. 33–61, Aug. 1998.

[52] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale l1-regularized least squares," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 606–617, Dec. 2007.

[53] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, "Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 586–597, Dec. 2007.

[54] E. van den Berg and M. P. Friedlander, "Probing the pareto frontier for basis pursuit solutions," *SIAM J. Sci. Comput.*, vol. 31, no. 2, pp. 890–912, Nov. 2008.

[55] A. Rakotomamonjy, "Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms," *Signal Process.*, vol. 91, no. 7, pp. 1505–1526, Jul. 2011.

[56] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation. Part II: Convex relaxation," *Signal Process.—Special Issue on Sparse Approximations in Signal and Image Processing*, vol. 86, no. 3, pp. 589–602, Mar. 2006.

[57] E. van den Berg and M. P. Friedlander, "Theoretical and empirical results for recovery from multiple measurements," *IEEE Trans. Inf. Theory*, vol. 56, no. 5, pp. 2516–2527, May 2010.
[58] D. Leviatan and V. N. Temlyakov, "Simultaneous approximation by greedy algorithms," *Adv. Comput. Math.*, vol. 25, no. 1–3, pp. 73–90, Jul. 2006.
[59] AVIRIS NW Indiana's Indian Pines 1992 Data Set. [Online]. Available: http://cobweb.ecn.purdue.edu/biehl/MultiSpec/documentation.html
[60] J. A. Richards and X. Jia, *Remote Sensing Digital Image Analysis: An Introduction*, 4th ed. New York: Springer-Verlag, 2006.

Yi Chen received the B.Eng. degree in electronic engineering from Tsinghua University, Beijing, China, in 2002, the M.A.Sc. degree in electrical engineering from the University of Victoria, Victoria, BC, Canada, in 2006, and she is currently working toward the Ph.D. degree in the Department of Electrical and Computer Engineering at The Johns Hopkins University, Baltimore, MD.

Her research interests include the applications of compressed sensing and sparse representations, image processing, wavelets, multirate systems, and filter banks.

Nasser M. Nasrabadi (S'80–M'84–SM'92–F'01) received the B.Sc. (Eng.) and Ph.D. degrees in electrical engineering from the Imperial College of Science and Technology (University of London), London, England, in 1980 and 1984, respectively.

From October 1984 to December 1984, he worked with IBM (UK) as a Senior Programmer. During 1985 to 1986, he worked with Philips Research Laboratory in NY as a Member of the Technical Staff. From 1986 to 1991, he was an Assistant Professor with the Department of Electrical Engineering at Worcester Polytechnic Institute, Worcester, MA. From 1991 to 1996, he was an Associate Professor with the Department of Electrical and Computer Engineering at State University of New York at Buffalo, Buffalo. Since September 1996, he has been a Senior Research Scientist with the US Army Research Laboratory, Adelphi, MD, working on image processing and automatic target recognition. He has served as an Associate Editor for the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON CIRCUITS, SYSTEMS AND VIDEO TECHNOLOGY, and the IEEE TRANSACTIONS ON NEURAL NETWORKS. His current research interests are in hyperspectral imaging, automatic target recognition, statistical machine learning theory, robotics, and neural networks applications to image processing.

He is also a Fellow of ARL, SPIE, and IEEE.

Trac D. Tran (S'94–M'98–SM'08) received the B.S. and M.S. degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1993 and 1994, respectively, and the Ph.D. degree in electrical engineering from the University of Wisconsin, Madison, in 1998.

In July of 1998, he joined the Department of Electrical and Computer Engineering, The Johns Hopkins University, Baltimore, MD, where he currently holds the rank of Associate Professor. His research interests are in the field of digital signal processing, particularly in sparse representation, sparse recovery, sampling, multirate systems, filter banks, transforms, wavelets, and their applications in signal analysis, compression, processing, and communications. His pioneering research on integer-coefficient transforms and pre/postfiltering operators has been adopted as critical components of Microsoft Windows Media Video 9 and JPEG XR C the latest international still-image compression standard ISO/IEC 29199-2. Dr. Tran was the codirector (with Prof. J. L. Prince) of the 33rd Annual Conference on Information Sciences and Systems (CISS'99), Baltimore, in March 1999. In the summer of 2002, he was an ASEE/ONR Summer Faculty Research Fellow with the Naval Air Warfare Center C Weapons Division at China Lake, CA. He is currently a Regular Consultant for the U.S. Army Research Laboratory in Adelphi, MD. He has served as Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, as well as IEEE TRANSACTIONS ON IMAGE PROCESSING.

Dr. Tran was a former member of the IEEE Technical Committee on Signal Processing Theory and Methods and is a current member of the IEEE Image Video and Multidimensional Signal Processing Technical Committee.. He received the NSF CAREER award in 2001, the William H. Huggins Excellence in Teaching Award from The Johns Hopkins University in 2007, and the Capers and Marion McDonald Award for Excellence in Mentoring and Advising in 2009.