Simultaneous Joint Sparsity Model for Target Detection in Hyperspectral Imagery

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

Abstract—This letter proposes a simultaneous joint sparsity model for target detection in hyperspectral imagery (HSI). The key innovative idea here is that hyperspectral pixels within a small neighborhood in the test image can be simultaneously represented by a linear combination of a few common training samples but weighted with a different set of coefficients for each pixel. The joint sparsity model automatically incorporates the interpixel correlation within the HSI by assuming that neighboring pixels usually consist of similar materials. The sparse representations of the neighboring pixels are obtained by simultaneously decomposing the pixels over a given dictionary consisting of training samples of both the target and background classes. The recovered sparse coefficient vectors are then directly used for determining the label of the test pixels. Simulation results show that the proposed algorithm outperforms the classical hyperspectral target detection algorithms, such as the popular spectral matched filters, matched subspace detectors, and adaptive subspace detectors, as well as binary classifiers such as support vector machines.

Index Terms—Hyperspectral imagery, joint sparsity model, simultaneous orthogonal matching pursuit, sparse representation, target detection.

I. INTRODUCTION

MOST natural signals are inherently sparse in a certain basis or with respect to a given dictionary. They can be approximately represented by a few coefficients carrying the most relevant information. Sparsity has played a very important role in many classical signal processing applications such as compression and denoising. The recent development in the sparse modeling of signals and images [1] has provided an extremely powerful tool for computer vision and pattern recognition [2].

In hyperspectral imagery (HSI), pixels are represented by vectors whose entries correspond to spectral bands, and images are represented by 3-D cubes. One of the most important applications of HSI is target detection, which can be viewed as a binary classification problem where pixels are labeled as target or background based on their spectral characteristics. Support vector machines (SVMs) have been a powerful tool to solve supervised classification problems and have shown good performances in hyperspectral classification [3], [4]. A number

of statistical hypothesis testing techniques have also been proposed for target detection in HSI [5]. Among these approaches, the spectral matched filter (SMF), matched subspace detectors (MSDs), and adaptive subspace detectors (ASDs) have been widely used to detect various targets of interests [6].

In this letter, we propose a new HSI target detection algorithm based on a joint sparsity model [7], [8] for pixels in a small neighborhood. It is observed that pixels belonging to the same class approximately lie in a low-dimensional subspace. Therefore, an unknown pixel lies in the union of the low-dimensional target and background subspaces and can be approximately represented by very few training samples from target or background subdictionaries. However, a direct application of this pixelwise sparse representation would ignore spatial information in the detection process. In this letter, we incorporate the spatial information from neighboring pixels by using the joint sparsity model where pixels in a small neighborhood are assumed to be simultaneously represented by a linear combination of a few common training samples, but for each pixel, these training samples are weighted with a different set of coefficients. In other words, we force the representations of neighboring pixels to have a common sparse support with respect to the training dictionary. The support is recovered by simultaneously decomposing the test pixels over the training dictionary. The recovery process implicitly involves a competition between the target and background subspaces, and the recovered sparse representation is naturally discriminative. The labels of the test samples are then directly determined by the property of the recovered sparse vectors.

This letter is structured as follows. The joint sparsity model and the proposed target detection algorithm are presented in Section II. The effectiveness of the proposed method is demonstrated by simulation results presented in Section III. Conclusions are drawn in Section IV.

II. SIMULTANEOUS JOINT SPARSITY MODEL FOR HSI TARGET DETECTION

In this section, we first briefly introduce the HSI target detection technique based on the sparse representation for a single pixel. Next, we show how to incorporate a joint sparsity constraint across the neighboring pixels of the HSI by adopting the simultaneous joint sparsity model [7], [8].

A. Pixelwise Sparsity-Based Target Detection

Let $x \in \mathbb{R}^B$ be a B-dimensional hyperspectral pixel observation whose entries correspond to the spectral bands. The spectrum of x is modeled to lie in the union of two low-dimensional subspaces: the background and target subspaces

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 the U.S. Army Research Laboratory, Adelphi, MD 20783 USA (e-mail: nnasraba@arl.army.mil).

spanned by background training samples $\{\boldsymbol{a}_i^b\}_{i=1,2,\ldots,N_b}$ and target training samples $\{\boldsymbol{a}_i^t\}_{i=1,2,\ldots,N_t}$, respectively. Therefore, $\boldsymbol{x}$ can be written as a sparse linear combination of all training pixels [9]

$$
\begin{aligned}
\boldsymbol{x} &= \left(\alpha_1^b \boldsymbol{a}_1^b + \cdots + \alpha_{N_b}^b \boldsymbol{a}_{N_b}^b\right) + \left(\alpha_1^t \boldsymbol{a}_1^t + \cdots + \alpha_{N_t}^t \boldsymbol{a}_{N_t}^t\right) \\
&= \underbrace{\begin{bmatrix} \boldsymbol{a}_1^b & \cdots & \boldsymbol{a}_{N_b}^b \end{bmatrix}}_{\boldsymbol{A}^b} \underbrace{\begin{bmatrix} \alpha_1^b \\ \vdots \\ \alpha_{N_b}^b \end{bmatrix}}_{\boldsymbol{\alpha}^b} + \underbrace{\begin{bmatrix} \boldsymbol{a}_1^t & \cdots & \boldsymbol{a}_{N_t}^t \end{bmatrix}}_{\boldsymbol{A}^t} \underbrace{\begin{bmatrix} \alpha_1^t \\ \vdots \\ \alpha_{N_t}^t \end{bmatrix}}_{\boldsymbol{\alpha}^t} \\
&= \boldsymbol{A}^b \boldsymbol{\alpha}^b + \boldsymbol{A}^t \boldsymbol{\alpha}^t = \underbrace{\begin{bmatrix} \boldsymbol{A}^b & \boldsymbol{A}^t \end{bmatrix}}_{\boldsymbol{A}} \underbrace{\begin{bmatrix} \boldsymbol{\alpha}^b \\ \boldsymbol{\alpha}^t \end{bmatrix}}_{\boldsymbol{\alpha}} = \boldsymbol{A}\boldsymbol{\alpha}. \quad (1)
\end{aligned}
$$

In the aforementioned equation, $\boldsymbol{A}^b$ and $\boldsymbol{A}^t$ are the background and target subdictionaries consisting of the N_b background and N_t target training samples (also called atoms), respectively, whereas $\boldsymbol{\alpha}^b$ and $\boldsymbol{\alpha}^t$ are vectors whose entries correspond to the weighting of the atoms in $\boldsymbol{A}^b$ and $\boldsymbol{A}^t$, respectively. The matrix $\boldsymbol{A} \in \mathbb{R}^{B \times N}$ with $N = N_b + N_t$ is the training dictionary consisting of both the background and target training samples, and $\boldsymbol{\alpha} \in \mathbb{R}^N$ is a concatenation of the two vectors $\boldsymbol{\alpha}^b$ and $\boldsymbol{\alpha}^t$. In this sparsity model, $\boldsymbol{\alpha}$ turns out to be a sparse vector (i.e., a vector with only few nonzero entries). The number K of nonzero entries in $\boldsymbol{\alpha}$ is called the sparsity level of $\boldsymbol{\alpha}$, and the index set Λ_K on which the entries of $\boldsymbol{\alpha}$ are nonzero is called the support of $\boldsymbol{\alpha}$. The sparse representation $\boldsymbol{\alpha}$ is very discriminative and contains important information about the class of the test sample $\boldsymbol{x}$.

Given the training dictionary $\boldsymbol{A}$, the sparse representation $\boldsymbol{\alpha}$ satisfying $\boldsymbol{A}\boldsymbol{\alpha} = \boldsymbol{x}$ can be obtained by solving the following problem:

$$
\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_0 \quad \text{subject to} \quad \boldsymbol{A}\boldsymbol{\alpha} = \boldsymbol{x} \quad (2)
$$

where $\|\cdot\|_0$ denotes the ℓ_0-norm which is defined as the number of nonzero entries in the vector. The aforementioned problem is NP-hard. Fortunately, if the solution is sufficiently sparse, it can be relaxed to a linear programming problem by replacing the ℓ_0-norm by the ℓ_1-norm and solved by convex programming techniques [1]. The problem in (2) can also be solved by greedy pursuit algorithms such as the orthogonal matching pursuit (OMP) [10] which efficiently approximates the solution with computational complexity on the order of $\mathcal{O}(BNK)$ for K iterations. Due to the presence of approximation error in the empirical data, the equality constraint can be relaxed to an inequality one as follows:

$$
\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{\alpha}\|_0 \quad \text{subject to} \quad \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{x}\|_2 \leq \sigma \quad (3)
$$

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

$$
\hat{\boldsymbol{\alpha}} = \arg\min \|\boldsymbol{A}\boldsymbol{\alpha} - \boldsymbol{x}\|_2 \quad \text{subject to} \quad \|\boldsymbol{\alpha}\|_0 \leq K_0 \quad (4)
$$

where K_0 is a given upper bound on the sparsity level [11]. In fact, the greedy algorithm OMP solves (3) or (4) depending on the stopping criterion of the algorithm, which will be explained in more detail in the next section.

The decomposition of the test sample $\boldsymbol{x}$ over the entire training dictionary $\boldsymbol{A}$ for the few most representative atoms leads to a competition between the two subspaces, and therefore, the recovered sparse vector $\boldsymbol{\alpha}$ is itself discriminative. The class of $\boldsymbol{x}$ can be determined by comparing the residuals $r_b(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{A}^b \hat{\boldsymbol{\alpha}}^b\|_2$ and $r_t(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{A}^t \hat{\boldsymbol{\alpha}}^t\|_2$, where $\hat{\boldsymbol{\alpha}}^b$ and $\hat{\boldsymbol{\alpha}}^t$ represent the recovered sparse coefficients corresponding to the background and target subdictionaries, respectively. The output of the detector is calculated by

$$
D(\boldsymbol{x}) = r_b(\boldsymbol{x}) - r_t(\boldsymbol{x}) = \|\boldsymbol{x} - \boldsymbol{A}^b \hat{\boldsymbol{\alpha}}^b\|_2 - \|\boldsymbol{x} - \boldsymbol{A}^t \hat{\boldsymbol{\alpha}}^t\|_2. \quad (5)
$$

If $D(\boldsymbol{x}) > \delta$ with δ being a prescribed threshold, then $\boldsymbol{x}$ is determined as a target pixel; otherwise, $\boldsymbol{x}$ is labeled as a background pixel.

B. Joint Sparsity Model

In the previous section, detection is performed for each pixel in the test image independently regardless of the correlation between neighboring pixels. However, in a typical HSI, the neighboring pixels usually consist of similar materials, and thus, their spectral characteristics are highly correlated. To exploit this crucial interpixel correlation, we could explicitly include an additional smoothing term, such as the Laplacian [9] or total variation [12], in the optimization problem formulation (2). However, in this letter, we propose to exploit the interpixel correlation by adopting the joint sparsity model in which the underlying sparse representations of multiple neighboring pixels share a common sparsity pattern. Since neighboring HSI pixels consist of similar materials, they can be approximated by a sparse linear combination of a few common atoms in the training dictionary but weighted with a different set of coefficients. The problem of simultaneous sparse representation and approximation for multiple measurement vectors has previously been studied in the literature [7], [8], but it has never been applied in HSI for target detection and classification.

The joint sparsity model for pixels in a small neighborhood $\mathcal{N}_\varepsilon$ consisting of T pixels is formulated as follows. Let $\boldsymbol{A}$ be the $B \times N$ training dictionary and $\boldsymbol{X} = [\boldsymbol{x}_1 \ \boldsymbol{x}_2 \ \cdots \ \boldsymbol{x}_T]$ be a $B \times T$ data matrix whose columns $\{\boldsymbol{x}_t\}_{t=1,\ldots,T}$ are pixels in $\mathcal{N}_\varepsilon$. Since the pixels in $\mathcal{N}_\varepsilon$ could consist of similar materials, they should be represented by a linear combination of a common set of K training samples $\{\boldsymbol{a}_{\lambda_1}, \ldots, \boldsymbol{a}_{\lambda_K}\}$. Specifically, the sparse representation of $\boldsymbol{x}_t \in \mathcal{N}_\varepsilon$ is expressed as

$$
\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{\alpha}_t = \alpha_{t,\lambda_1} \boldsymbol{a}_{\lambda_1} + \alpha_{t,\lambda_2} \boldsymbol{a}_{\lambda_2} + \cdots + \alpha_{t,\lambda_K} \boldsymbol{a}_{\lambda_K}.
$$

The index set $\Lambda_K = \{\lambda_1, \lambda_2, \ldots, \lambda_K\}$ is the support of $\boldsymbol{\alpha}_t$, and each pixel $\boldsymbol{x}_t$ has a different set of coefficients $\{\alpha_{t,k}\}_{k \in \Lambda_K}$ associated with it. The data matrix $\boldsymbol{X}$ can then be represented as

$$
\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1 & \boldsymbol{x}_2 & \cdots & \boldsymbol{x}_T \end{bmatrix} = \boldsymbol{A} \underbrace{\begin{bmatrix} \boldsymbol{\alpha}_1 & \boldsymbol{\alpha}_2 & \cdots & \boldsymbol{\alpha}_T \end{bmatrix}}_{\boldsymbol{S}} = \boldsymbol{A}\boldsymbol{S} \quad (6)
$$

where each of the sparse vectors $\{\boldsymbol{\alpha}_t\}_{t=1,\ldots,T}$ has the same support Λ_K and $\boldsymbol{S}$ is a sparse matrix with only K nonzero rows. For convenience, we call the index set Λ_K of $\boldsymbol{\alpha}_t$ also as the support of $\boldsymbol{S}$.

Given the dictionary A, the matrix S is obtained by solving the following joint sparse recovery problem:

$$\begin{aligned} \text{minimize} \quad & \|S\|_{\text{row},0} \\ \text{subject to} \quad & AS = X \end{aligned} \quad (7)$$

where the notation $\|S\|_{\text{row},0}$ denotes the number of nonzero rows of S (also called the diversity of S in [8]). The solution to the aforementioned problem $\hat{S} = [\hat{\alpha}_1 \ \hat{\alpha}_2 \ \cdots \ \hat{\alpha}_T]$ is an $N \times T$ sparse matrix with only few nonzero rows. Similar to the sparse recovery problem in (2), the simultaneous sparse recovery problem in (7) is an NP-hard problem. It can be solved by greedy algorithms [7], [8] or relaxed to convex programming [1]. For empirical data, the problem in (7) can also be rewritten to account for the approximation errors [13], as was done in (3) and (4)

$$\begin{aligned} \text{minimize} \quad & \|S\|_{\text{row},0} \\ \text{subject to :} \quad & \|AS - X\|_F \le \sigma \end{aligned} \quad (8)$$

or

$$\begin{aligned} \text{minimize} \quad & \|AS - X\|_F \\ \text{subject to :} \quad & \|S\|_{\text{row},0} \le K_0. \end{aligned} \quad (9)$$

In this letter, the simultaneous sparse recovery problem is solved by a greedy algorithm, called the simultaneous orthogonal matching pursuit (SOMP), whose implementation details are presented in [7]. The SOMP recovers the common support set Γ_K in a sequential fashion (i.e., the atoms in the dictionary A are sequentially selected). At each iteration, the atom that simultaneously yields the best approximation to all of the residual vectors is selected. The SOMP algorithm terminates when the error residual $\|AS - X\|_F$ is sufficiently small or when the desired level of sparsity (controlled by the number of iterations) is achieved, which solves the problems in (8) or (9), respectively.

After the sparse matrix $\hat{S}$ is recovered, the labels of the test samples can be determined based on the characteristics of the sparse coefficients as is done in Section II-A. We calculate and compare the total error residuals between the original test samples and the approximations obtained from the background and target subdictionaries. The output of the proposed sparsity-based detector is computed as in (5) by the difference of the total residuals from all of the pixels in the neighborhood

$$D(x) = \|X - A^b \hat{S}^b\|_F - \|X - A^t \hat{S}^t\|_F \quad (10)$$

where $\hat{S}^b$ consists of the first N_b rows of the recovered matrix $\hat{S}$ corresponding to the background subdictionary A^b and $\hat{S}^t$ consists of the remaining N_t rows in $\hat{S}$ corresponding to the target subdictionary A^t. If the output is greater than a prescribed threshold, then the test sample is labeled as a target; otherwise, it is labeled as background.

An example of the proposed detection technique based on the joint sparsity model is illustrated in Fig. 1. The test pixel x_1 is taken from the background region in the upper right corner of the desert radiance II data collection (DR-II), shown in Fig. 3(a). More details about this image will be discussed in Section III. The dictionary A consists of $N_t = 18$ target training samples and $N_b = 216$ background training samples. Using
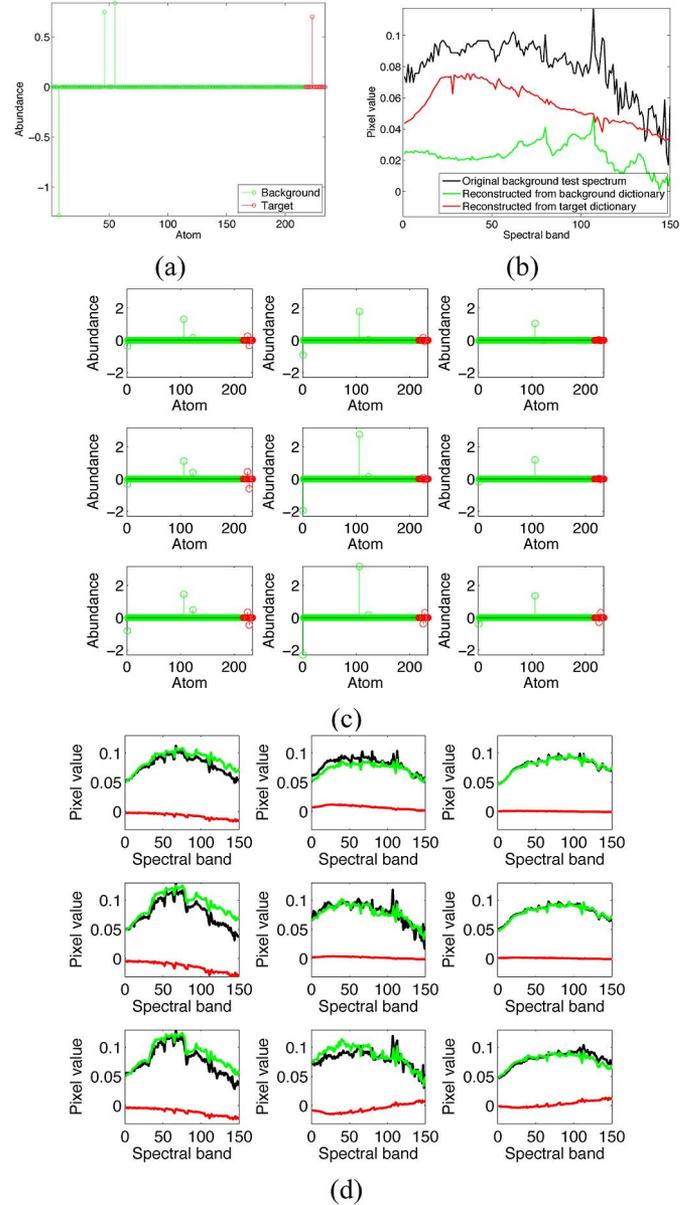

Fig. 1. Example comparing the pixelwise sparsity model and the joint sparsity model. (a) Solution to (4) for a background test pixel x_1 with sparsity $K = 5$. (b) Original test sample (black) with its reconstructions from the background subdictionary (green) and from the target subdictionary (red) by solving (4). (c) Solution to (9) for the nine pixels in a 3×3 window centered at the pixel x_1 with sparsity $K = 5$. (d) Nine original test pixels (black) with their reconstructions from the background subdictionary (green) and from the target subdictionary (red) by solving (9).

the sparsity model for a single pixel in (1) and the OMP algorithm, the sparse vector $\hat{\alpha}_1$ of this background involves both background and target training samples, as seen in Fig. 1(a). Neither of the approximations obtained from the background and target subdictionaries represents well the original test pixel [see Fig. 1(b)]. On the other hand, when incorporating the spatial correlation by applying the joint sparsity model in (6) on a 3×3 neighborhood centered at x_1, the test sample x_1 is well approximated by the background subdictionary, as seen in Fig. 1(d). Therefore, with the appropriate help from the joint sparsity model, the test pixel is correctly classified as background.
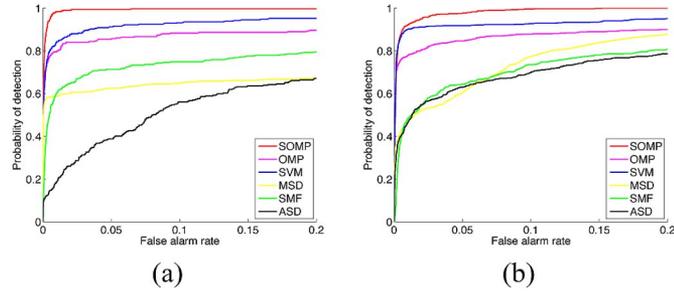
Fig. 2. ROC curves using various detection and classification algorithms for (a) DR-II and (b) FR-I using local background dictionary (dual-window approach). $N_t = 18$, and $N_b = 216$.

III. SIMULATION RESULTS AND ANALYSIS

The proposed target detection algorithm, as well as the classical techniques SMF, MSD, ASD, and SVM, is applied to several real HSI. The results are compared both visually and quantitatively by the receiver operating characteristic (ROC) curves. The ROC curve describes the probability of detection (PD) as a function of the probability of false alarms (PFA). To calculate the ROC curve, we pick thousands of thresholds between the minimum and maximum of the detector output. The class labels for all pixels in the test region are determined at each threshold. The PFA is calculated by the number of false alarms (background pixels determined as target) over the total number of pixels in the test region, and the PD is the ratio of the number of hits (target pixels determined as target) and the total number of true target pixels.

The two hyperspectral images used in the experiments, the DR-II and the forest radiance I data collection (FR-I), are from the hyperspectral digital imagery collection experiment (HYDICE) sensor [14]. The HYDICE sensor generates 210 bands across the whole spectral range from 0.4 to 2.5 μm which includes the visible and short-wave infrared bands. We use 150 of the 210 bands (23rd–101st, 109th–136th, and 152nd–194th) by removing the absorption and low-SNR bands. The DR-II image contains six military targets on the dirt road, while the FR-I image contains 14 targets along the tree line, as seen in Figs. 3(a) and 4(a), respectively. For these two HYDICE images, every pixel on the targets is considered a target pixel when computing the ROC curves. We use a small target subdictionary constructed by $N_t = 18$ pixels on the leftmost target in the scene. For the background subdictionary, we use an adaptive local background dictionary. Specifically, the background sub-dictionary $\boldsymbol{A}^b$ is generated locally for each test pixel using a dual window centered at the pixel of interest. The inner window should be larger than the size of the targets in the scene, and only pixels in the outer region will form the atoms in $\boldsymbol{A}^b$. In this way, the subspace spanned by the background subdictionary becomes adaptive to the local statistics. For the DR-II and the FR-I images, the outer and inner windows have sizes of 21×21 and 15×15, respectively, so there are $N_b = 216$ background training samples in $\boldsymbol{A}^b$.

The detection techniques using the single-pixel sparsity model (1) and the joint sparsity model (6) are applied on these two HYDICE images. The OMP algorithm [10] is used to solve the sparsity-constrained problem in (4). The SOMP
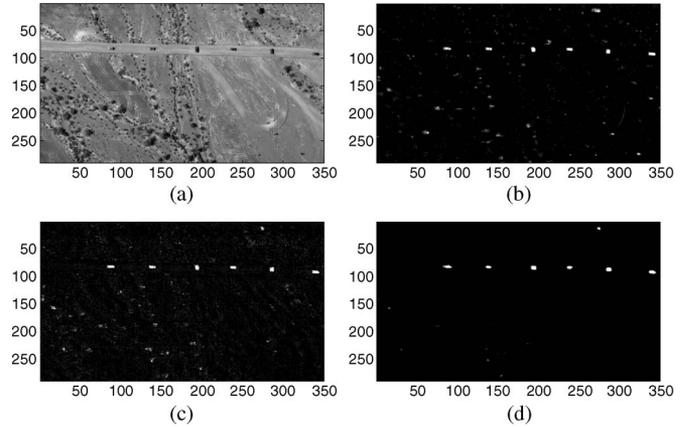

Fig. 3. DR-II. (a) Averaged image over 150 bands. Detection results for DR-II using (b) SVM with composite kernel, (c) OMP, and (d) SOMP.
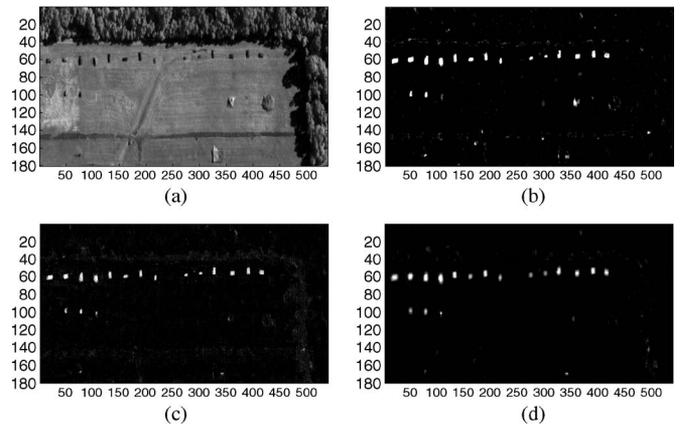

Fig. 4. FR-I. (a) Averaged image over 150 bands. Detection results for FR-I using (b) SVM with composite kernel, (c) OMP, and (d) SOMP.

algorithm is used to solve the simultaneous sparse recovery problem in (9). The ROC curves obtained using the OMP and the SOMP are shown in Fig. 2, with the sparsity level being set to $K_0 = 10$. For the SOMP, a 5×5 square neighborhood [$T = 25$ in the problem formulation (6)] is used. The detector output for these two sparsity-based algorithms is shown in Fig. 3(c) and (d) for the DR-II and in Fig. 4(c) and (d) for the FR-I. We see that the incorporation of the spatial correlation between neighboring pixels through the joint sparsity model yields better detection results than a direct application of the pixelwise sparsity model (1).

Under the same settings (i.e., same target and background training samples for all detectors), the classical statistical detectors SMF, MSD, and ASD, as well as the binary classifier SVM, are also applied to detect the targets of interests. The first three statistical detectors are pixelwise detectors, and their implementation details can be found in [6]. For the SVM, we use a composite kernel that combines the spectral and spatial information via a weighted summation, which has been shown to outperform the spectral-only SVM in HSI classification [4]. For each pixel, the spatial features (e.g., the mean and the standard deviation per spectral band) are explicitly extracted in the 5×5 neighborhood centered at that pixel. An SVM is then trained for each spectral–spatial pixel using atoms in $\boldsymbol{A}^b$ and $\boldsymbol{A}^t$ as belonging to two different classes
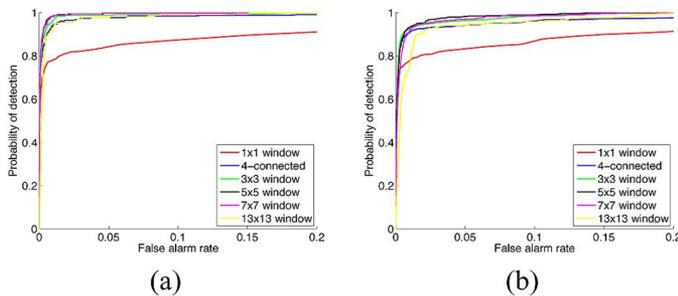
Fig. 5. Effects of the neighborhood size T on detection performance for (a) DR-II and (b) FR-I.
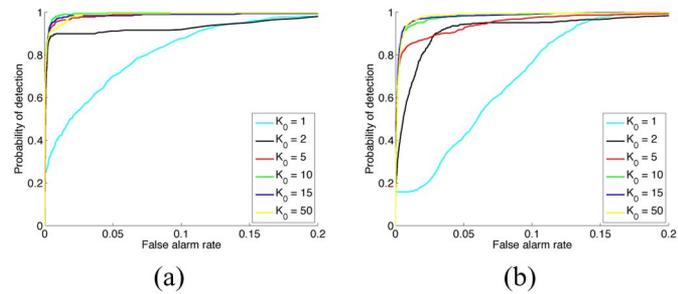

Fig. 6. Effects of the sparsity level K_0 on detection performance for (a) DR-II and (b) FR-I.

with radial basis function kernels for both the spectral and spatial features. The parameters associated with the SVM and composite kernels are defined and explained in detail in [3] and [4]. For comparison, the results obtained from the SVM with composite kernels are also shown in Figs. 3(b) and 4(b). The ROC curves for the two test HYDICE images using SMF, MSD, ASD, and SVM are shown in Fig. 2. Overall, from this figure, one can observe that, for both images, the detector based on the joint sparsity model yields the best performance. The simultaneous sparse recovery algorithms clearly outperform the classical target detection/classification algorithms.

Next, we illustrate the effect of the neighborhood size on the detection performance. In this experiment, the simultaneous sparse approximation problem in (9) is solved by the SOMP for different neighborhoods at a fixed sparsity level $K_0 = 10$. Specifically, we use a 1×1 neighborhood (equivalent to the pixelwise sparsity model), a four-connected neighborhood, and 3×3, 5×5, 7×7, and 13×13 window neighborhoods, corresponding to $T = 1, 5, 9, 25, 49$, and 169, respectively. The ROC curves for the various types of neighborhood are shown in Fig. 5. By incorporating the contextual interaction between neighboring pixels, the detector performance is significantly improved. As the neighborhood size increases, the performance tends to saturate.

Next, we demonstrate how the detection results are affected by the sparsity level of the representation. The sparsity level refers to the number of nonzero rows in S, which is also the number of the common atoms selected from the dictionary by the greedy algorithms to simultaneously approximate all of the neighboring pixels. The neighborhood size T is fixed to 25 (i.e., a 5×5 window is used) in this experiment. The ROC curves for both images using the SOMP with sparsity levels $K_0 = 1$, 2, 5, 10, 15, and 50 are shown in Fig. 6. For very small K_0, the sparsity-based technique reduces to simple template matching

and leads to underfitting. Generally, the detection performance improves as the sparsity level K_0 increases to a certain level. However, if K_0 is too large, the solution becomes dense and involves both the background and target atoms, and thus, its discriminative power degrades. In this letter, the sparsity level K_0 is chosen to be slightly smaller than the size of the target dictionary.

IV. CONCLUSION

In this letter, a joint sparsity model is proposed for target detection in HSI. The interpixel correlation in HSI is incorporated by employing the joint sparsity model where the pixels in a small neighborhood in the test image are represented by a linear combination of a few common training samples weighted with a different set of coefficients for each pixel. The sparse representations of the neighboring pixels are obtained from a simultaneous sparse recovery algorithm called the SOMP. The resulting sparse representations are then used directly for target detection. Simulation results show that the proposed algorithm outperforms the classical hyperspectral target detection algorithms, including SMF, MSD, ASD, and SVM.

REFERENCES

[1] 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, Feb. 2009.

[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] 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.

[4] 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.

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

[6] H. Kwon and N. M. Nasrabadi, "A comparative analysis of kernel subspace target detectors for hyperspectral imagery," *EURASIP J. Appl. Signal Process.*, vol. 2007, no. 1, p. 193, Jan. 2007.

[7] 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.

[8] 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.

[9] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Sparse representation for target detection in hyperspectral imagery," *IEEE J. Sel. Topics Signal Process.*, 2010, submitted for publication.

[10] J. A. 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.

[11] 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.

[12] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, no. 1–4, pp. 259–268, Nov. 1992.

[13] A. Rakotomamonjy, "Surveying and comparing simultaneous sparse approximation (or group-lasso) algorithms," HAL-00328185, 2010. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00328185/fr

[14] R. W. Basedow, D. C. Carmer, and M. E. Anderson, "HYDICE system: Implementation and performance," in *Proc. SPIE Conf. Algorithms Technol. Multispectral, Hyperspectral, Ultraspectral Imagery XV*, Jun. 1995, vol. 2480, pp. 258–267.