Denoising Hyperspectral Imagery and Recovering Junk Bands using Wavelets and Sparse Approximation

Adam C. Zelinski and Vivek K Goyal
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
{zelinski, vgoyal}@mit.edu

Abstract— **In this paper, we present two novel algorithms for denoising hyperspectral data. Each algorithm exploits correlation between bands by enforcing simultaneous sparsity on their wavelet representations. This is done in a non-linear manner using wavelet decompositions and sparse approximation techniques. The first algorithm denoises an entire cube of data. Our experiments show that it outperforms wavelet-based global soft thresholding techniques in both a mean-square error (MSE) and a qualitative visual sense. The second algorithm denoises a set of noisy, user designated bands ("junk bands") by exploiting correlated information from higher quality bands within the same cube. We prove the utility of our junk band denoising algorithm by denoising ten bands of actual AVIRIS data by a significant amount. Preprocessing data cubes with these algorithms is likely to increase the performance of classifiers that make use of hyperspectral data, especially if the denoised and/or recovered bands contain spectral features useful for discriminating between classes.**

I. INTRODUCTION

Many important applications, such as target detection, material identification and material mapping, require high signal-to-noise ratio (SNR) hyperspectral imagery to achieve good performance. Thus, algorithms that denoise hyperspectral data are worthwhile.

In modern hyperspectral imaging systems, such as AVIRIS [8], many of the spectral bands have high SNR, but a significant number of bands (up to 20 percent) are extremely noisy due to atmospheric effects. Many researchers in the classification community simply discard these *junk bands* when training and testing their hyperspectral data classification algorithms [7], [12]. An algorithm that denoises and recovers these junk bands would allow them to be used in classification systems. This will lead to higher classification performance, especially if the recovered bands contain spectral features useful for discriminating between classes.

Current denoising techniques. Several techniques exist to denoise hyperspectral data on a band-by-band basis. Since each band is a two-dimensional (2-D) image, one might consider using a Wiener filter, since this method provides the minimum mean-square error (MSE) estimate of an image from its noisy observation. However, this is infeasible in many cases because it requires second-order statistics that are typically unavailable.

A more feasible and useful denoising technique is *wavelet thresholding* or *shrinkage*, pioneered by Donoho and Johnstone [5], [6]. In this method, one performs a 2-D Discrete Wavelet Transform (DWT) on an image and then soft thresholds the coefficients in the detail subbands.

Atkinson et al. created a denoising technique that uses Discrete Fourier Transforms (DFTs), 2-D Discrete Wavelet Transforms (DWTs), and soft thresholding of wavelet coefficients to denoise hyperspectral imagery [1]. The DFTs exploit signal correlation across bands, but only to a limited extent, since they are signal independent. Furthermore, this algorithm uses complex wavelet transforms and

performs soft thresholding on *complex* coefficients, the latter of which is not explained in their paper.

A different approach. We propose two algorithms: the first denoises an entire data cube while the second recovers user designated junk bands. Both algorithms utilize a combination of wavelet and sparse approximation techniques, as discussed below.

The first algorithm denoises each band in a data cube. It exploits correlation across a set of band images, enforcing sparsity on the wavelet representations of the images in a simultaneous, non-linear manner. This sparsity-enforcing technique is inspired by recent work in the array processing community [10].

The second algorithm is for junk band recovery. To use this technique, the user designates a set of high SNR "good bands" and a set of noisy "junk bands." After first determining the sparsity profile of the good bands' wavelet coefficients, the algorithm forces each of the junk bands to have a similar sparsity profile, which denoises them simultaneously. To the best of our knowledge, our junk band recovery algorithm is a novel idea.

Organization of the paper. We start by introducing the basics of hyperspectral data, wavelet theory, and sparse approximation in Section II. We explain our proposed denoising algorithms in Section III and present experimental results in Section IV. Finally, we conclude our paper, providing ideas for future work.

II. BACKGROUND

Hyperspectral imagery. Hyperspectral sensors collect surface reflectance data at hundreds of closely spaced wavelengths. Thus, for each pixel in an imaged scene, we obtain a *spectrally overcomplete* look at a point on the surface. This oversampled representation is useful for classifying each pixel's contents [13].

Hyperspectral imagery is thus a *data cube*, having two spatial dimensions and a third spectral dimension. Fixing the wavelength band yields a 2-D image of the scene at a particular wavelength, so hyperspectral imagery may also be visualized as a stack of 2-D band images, each corresponding to a certain wavelength. Since the bands are so closely spaced in wavelength, *sets of band images are highly correlated*.

Wavelet theory. Wavelets are a popular multiresolution analysis technique in the image compression, denoising and remote sensing communities. Using wavelets to decompose a signal allows one to capture broad, general behavior as well as localized, high-frequency content [15]. A wavelet basis is formed by shifts and scales of a single *mother wavelet*; in the DWT, this wavelet is shifted and scaled by powers of two.

In this paper, we perform wavelet analysis on a band-by-band basis, so 2-D DWTs are used. They are implemented via Mallat's efficient filterbank structure [11]. Wavelet *scaling coefficients* capture the broad, smooth structure of each image, whereas *detail coefficients*

capture higher frequency, localized behavior (e.g., edges). For all experiments in this paper, we use the Daubechies order-2 wavelet ("db2") and a 4-level wavelet decomposition.

Sparsity of wavelet representations. Applying a 2-D DWT to an image corrupted by additive white Gaussian noise (AWGN) results in a *sparse representation*. That is, in terms of magnitude, only a few of the wavelet coefficients are large (those containing signal + noise), and the majority are small (noise only) [15]. Wavelet-based denoising algorithms exploit this compactness of the true signal in the wavelet domain by zeroing out the noise coefficients while retaining signal coefficients; the most prevalent denoising method is to soft threshold each coefficient [5], [6].

Let I be an N x M noisy image. Let $W_J(\cdot)$ and $W_J^{-1}(\cdot)$ denote the 2-D DWT and 2-D Inverse DWT (IDWT) operators, respectively. Then, in the sense of the above discussion, $W_J(I)$ is a *sparse* matrix. Lexicographically arranging $W_J(I)$ into a column vector results in a sparse vector (rearranging the elements of a matrix has no effect on their sparsity). Let us denote the lexicographic ordering of a matrix into a vector by vec$(\cdot)$.

Wavelet representations of hyperspectral imagery. As mentioned above, band images within a cube are highly correlated in high SNR situations. Let $\{I_i\}_{i=1}^T$ be a set of high SNR band images that form a data cube. Based on our discussion above, each $t_i = \text{vec}(W_J(I_i))$ is sparse, and the *sparsity profiles* of the $\{t_i\}$ are similar. By "sparsity profile," we mean the set of indices where large-magnitude coefficients are located.

If we are no longer in a high SNR situation, then the $\{I_i\}$ are noisy and the $\{t_i\}$ no longer necessarily have the same sparsity profiles. The wavelet representations of the *true* signals within each of the noisy I_i, however, *should* have the same sparsity pattern (identical to the pattern in the high SNR case).

Sparse approximation. The goal of sparse approximation is to find a vector of unknowns *with a small number of nonzero elements* such that a system of equations (approximately) holds.

Consider a linear system of equations where we have more unknowns than equations, e.g., $y = Ax$, where $y \in \Re^M$, $A \in \Re^{M \text{ x } N}$, $x \in \Re^N$, and $M < N$. This problem is ill-posed because there are infinitely many x that solve the system.

Consider, however, enforcing sparsity on x by requiring it to have few nonzero coefficients. This is the same as requiring that $||x||_0$ be small, where $|| \cdot ||_0$ is a pseudo-norm equal to the number of nonzero elements of the vector input argument. Introducing a sparsity constraint on x makes the problem more challenging. For example, the following optimization problem is NP-complete [4]:

$$\min_x \|x\|_0 \text{ s.t. } y = Ax.$$

Thus, finding the sparsest x when the system is large is computationally infeasible. Let us relax the problem and minimize the l_1 norm of x rather than its number of nonzero elements:

$$\min_x \|x\|_1 \text{ s.t. } y = Ax.$$

This problem is convex and quickly solved via a linear program; furthermore, various empirical studies have shown that minimizing the l_1 norm of x usually leads to a sparse solution (e.g., [3]).

Let us now perturb the system with a noise vector, $n \in \Re^N$, such that $y = Ax + n$. Due to the noise, it is unlikely that a sparse x exists such that Ax exactly equals y. A more practical solution is to find a sparse x that *approximately* solves the noisy system. The following optimization is useful to find such an x:

$$\min_x (1 - \lambda)\|y - Ax\|_2 + \lambda\|x\|_1. \tag{1}$$

The first term of (1) keeps the residual error down, whereas the second enforces sparsity on x. As the regularization parameter, λ, is increased from zero to one, the algorithm yields sparser and sparser solutions and the residual error increases. Thus, λ is a control parameter that trades off sparsity versus residual error.

Now suppose we have a set of noisy vectors, $y_i = Ax_i + n_i$, $i = 1, \ldots, T$. Letting $Y = [y_1, \ldots, y_T]$ and $X = [x_1, \ldots, x_T]$, we generalize (1) into a program that finds an X that approximately solves $Y = AX$, with the constraint that each column of X (i.e., each of the x_i) have a similar sparsity profile. We pose the problem as follows:

$$\min_X (1 - \lambda)\|Y - AX\|_{\mathbf{F}} + \lambda\|X\|_{\mathbf{S}}, \tag{2}$$

where $\|X\|_{\mathbf{F}}$ is the Frobenius norm of X and $\|X\|_{\mathbf{S}}$ is the l_1 norm of the l_2 norms of the rows of X. The latter operator is a *simultaneous sparsity norm*: it penalizes the program (i.e., produces large values) when the columns of X have dissimilar sparsity profiles and rewards the program (i.e., produces small values) when the columns of X have similar sparsity profiles. More details about this simultaneous sparsity norm and (2) may be found in [10].

Reformulating (2) into a Conic-Quadratic Program allows one to solve the problem with optimization software. For experiments in this paper, we use the SeDuMi toolbox [14]. Using SeDuMi, (2) is solvable in a reasonable amount of time on a modern personal computer when the number of variables (i.e., the number of elements of X) is under a thousand.

III. PROPOSED DENOISING ALGORITHMS

In this section, we summarize our denoising and junk band recovery algorithms. The first algorithm takes a noisy data cube as input and outputs a denoised cube. The second denoises junk bands by enforcing the sparsity profiles of known good bands on the junk bands.

Denoising algorithm. In Section II, we explained that band images in a high SNR data cube are correlated, and thus their wavelet coefficients have similar sparsity patterns. Let $\{I_i\}_{i=1}^T$ be a set of noisy band images. There is a true (non-noisy) image within each I_i, each of which has a similar sparse representation in the wavelet domain. Our algorithm exploits this fact, seeking to recover the clean images within each noisy I_i by imposing the same sparsity pattern on the wavelet coefficients of each I_i.

Let $\hat{I}_i$ denote a denoised version of I_i, i.e., an estimate of the true signal within the noisy I_i. If we could denoise the I_i perfectly, then each $\hat{I}_i$ would be sparse in the wavelet domain and each would have a similar sparsity pattern.

By using the sparse approximation framework described in Section II, we can formulate an optimization problem to enforce our sparsity assumption across the $\{I_i\}$. Our program will penalize the *lack of sparsity* of wavelet representations of the band images. We now outline and explain the algorithm:

1. Perform a J-level 2-D DWT on each I_i, setting aside the scaling coefficients and lexicographically arranging the N_D detail coefficients into a vector, $d_i \in \Re^{N_D}$.

2. Let $D = [d_1, \ldots, d_T]$.

3. For each i, we want to find a $\hat{d}_i$ that is close to d_i in an l_2 norm sense, while at the same time making sure that the $\hat{d}_i$ are simultaneously sparse. Letting $\hat{D} = [\hat{d}_1, \ldots, \hat{d}_T]$, we restate the problem: find a $\hat{D}$ that is close to D in the Frobenius-norm sense and whose columns have similar sparsity profiles. This is accomplished via:

$\sigma = 250$	DA	DU	AT	HS	SL	MM
MSE_1	**2.38**	3.86	3.63	4.05	4.05	3.76
MSE_2	**1.93**	3.61	3.29	3.77	3.76	3.12
MSE_3	**1.72**	3.66	3.37	4.02	4.01	3.23
MSE_4	**1.69**	3.82	3.21	3.91	3.99	3.21
MSE_5	**1.75**	4.75	3.93	5.29	5.03	3.84
MSE_6	**1.97**	5.64	4.73	6.55	6.53	4.62
MSE_7	**2.20**	6.60	5.19	7.67	7.64	5.43
MSE_8	**2.21**	6.89	5.52	8.05	7.83	5.65
MSE_9	**2.50**	7.76	5.90	8.73	8.84	6.10
MSE_{10}	**2.37**	7.64	5.96	8.79	9.23	6.35
MSE_μ	**2.07**	5.43	4.47	6.08	6.09	4.53

$\sigma = 500$	DA	DU	AT	HS	SL	MM
MSE_1	**3.9**	4.3	4.3	4.4	4.3	4.1
MSE_2	**3.3**	4.4	4.2	4.4	4.3	4.1
MSE_3	**3.2**	5.1	4.4	4.9	5.0	4.8
MSE_4	**3.1**	5.0	4.8	5.1	5.1	4.1
MSE_5	**3.7**	6.8	6.0	7.0	7.2	5.4
MSE_6	**4.2**	8.7	7.6	8.7	8.9	6.8
MSE_7	**4.8**	10.5	8.3	10.9	10.4	7.9
MSE_8	**5.2**	11.1	9.2	11.7	11.3	8.4
MSE_9	**5.6**	11.9	9.7	12.7	12.7	9.3
MSE_{10}	**5.6**	11.8	9.6	12.9	12.8	9.4
MSE_μ	**4.3**	8.0	6.8	8.3	8.2	6.4

$$\min_{\hat{D}} \; (1-\lambda)\|D - \hat{D}\|_{\mathbf{F}} + \lambda\|\hat{D}\|_{\mathbf{S}}. \tag{3}$$

A solution to (3) exists because it is a special case of (2). We assume that λ is well-chosen such that the residual error is small and the columns of $\hat{D}$ are simultaneously sparse; this is not an unreasonable assumption (see Section IV).

Note that when the band images I_i have many pixels, their wavelet representations have many detail coefficients, matrix D has thousands of elements, and (3) is too large to solve. To circumvent this issue, we use a divide-and-conquer strategy: D is partitioned into a set of submatrices, each of which is estimated individually using SeDuMi; these estimated submatrices are then assembled to form $\hat{D}$.

4. Each column of $\hat{D}$ contains the denoised detail coefficients of image I_i. Extract the columns of $\hat{D}$ to yield $\{\hat{d}_i\}$. Reform a set of 2-D wavelet coefficients using the I_i's scaling coefficients and the denoised detail coefficients $\hat{d}_i$. Perform a 2-D IDWT on this data, yielding $\hat{I}_i$, a denoised version of I_i.

Junk band recovery algorithm.

1. Let $\{G_i\}_{i=1}^{K}$ and $\{J_i\}_{i=1}^{T}$ be sets of user designated good bands and junk bands, respectively. The first set contains bands with high SNR, the latter is comprised of those with low SNR that classification researchers normally discard from their data cubes.
2. Compute $G_\mu = \sum_{i=1}^{K} G_i$. Averaging K high SNR observations of essentially the same scene results in a G_μ with negligible noise. Let $g_\mu = \text{vec}(\text{detailCoeffs}(W_J(G_\mu)))$; this vector will have an obvious sparsity profile since G_μ has negligible noise.
3. Compute $d_i = \text{vec}(\text{detailCoeffs}(W_J(J_i)))$, i.e., perform a 2-D DWT on each of the J_i, arranging their detail coefficients into columns. Let $D = [d_1, \ldots, d_T]$.
4. Let $G = [D, g_\mu]$, i.e., append g_μ to the matrix D.
5. Our goal is to find a matrix $\hat{D}$ with two properties—its columns should be close to those of D, and each column of $\hat{D}$ should have a sparsity profile similar to g_μ's. We find such a matrix by solving the following problem:

$$\min_{\hat{G}} \; (1-\lambda)\|G - \hat{G}\|_{\mathbf{F}} + \lambda\|\hat{G}\|_{\mathbf{S}}. \tag{4}$$

If G is too large to compute all entries of $\hat{G}$ at once, we apply the divide-and-conquer strategy from the first algorithm to (4).

6. Extract $\hat{D}$ from the resulting $\hat{G}$. Assuming λ is properly chosen, the columns of $\hat{D}$ will have g_μ's sparsity pattern.
7. Analogous to the first algorithm, for each i, recover a denoised version of junk band $\hat{J}_i$ by doing a 2-D DWT on J_i's original scaling coefficients and its denoised detail coefficients (the ith column of $\hat{D}$).

IV. EXPERIMENTAL RESULTS

In this section, we demonstrate the effectiveness of our algorithms by performing several experiments and discussing their results.

In the first experiment, we add white Gaussian noise to high SNR real data. We test the effectiveness of our proposed denoising algorithm and compare its performance to denoising via soft thresholding of wavelet coefficients. Performance is judged by analyzing MSE between the denoised and original images. The second experiment uses real data and provides a set of good bands and junk bands for use in the junk band recovery algorithm. The performance of the algorithm is judged qualitatively by analyzing the recovered junk bands.

Experiment data. We use a subset of an AVIRIS data cube available from [9]. Our data cube is comprised of 220 band images, each 64 x 64 pixels in size. Each pixel, p, of each band image represents surface reflectivity; $p \in \{0, 1, \ldots, 10000\}$.

Denoising experiment. To test our denoising algorithm, we perturb the first ten bands of the data cube (all of which have high SNR) with zero-mean AWGN with a known standard deviation σ. We then run our algorithm (using a specified value for λ) and measure its performance by computing the MSE between the denoised bands and the corresponding original bands. We use MSE as our metric because of its wide use in the denoising community for comparing different algorithms [2].

We also denoise the images using standard Donoho-Johnstone wavelet thresholding techniques [5], [6]. The images are denoised individually (on a band-by-band basis) and an MSE is computed for each image. The performance of various global soft thresholding techniques is evaluated.

Denoising results. Tables I and II list the MSEs of various algorithms when the noise perturbing the cube has a standard deviation (σ) of 250 and 500, respectively. In each table, our denoising algorithm is denoted by DA. The other labels correspond to global soft thresholding techniques: DU, AT, HS, SL, and MM stand for Donoho's VisuShrink universal threshold ([6]), Atkinson's threshold ([1]), "heursure," "sqtwolog," and "minimaxi," respectively. Implementations of the latter three thresholding algorithms are available
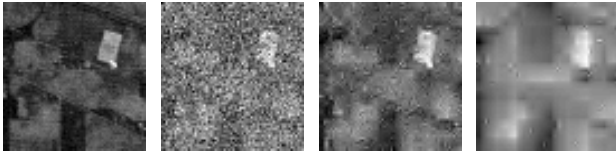
Fig. 1. Denoising Experiment. From left to right: the original, noisy ($\sigma = 500$), DA-denoised ($MSE_4 = 3.1e4$), and MM-denoised ($MSE_4 = 4.1e4$) images of band 4.

from the Wavelet Toolbox within MATLAB. MSE_i is the MSE between the denoised version of the noisy ith band and the original ith band; MSE_μ is the average MSE over all ten bands.

Looking at Tables I and II, one sees that our algorithm outperforms all global soft thresholding techniques in terms of MSE.

For the results in Table I (when $\sigma = 250$), the MSE between each *noisy* band image and its corresponding original image should be approximately $6.25 \cdot 10^4$ (or 6.25 if such a value appeared in the table). We see from the results that some of the $MSE/10^4$ values resulting from global soft thresholding techniques (DU, HS, SL, and MM) are greater than 6.25, meaning that these algorithms actually *worsen* the MSE of some bands. Compared to all the other techniques, our algorithm (DA) results in lower MSEs for each band image (i.e., removes more noise from each band image), outperforming the others.

For the results in Table II (when $\sigma = 500$), the MSE between each noisy and original band image should be approximately $25 \cdot 10^4$ (or 25.0 in the table). Since all of the values in Table II are much less than 25.0, this means that all of the denoising techniques are useful, significantly reducing the noise added to each image. Our technique results in lower MSE values and outperforms the others.

In Figure 1, the original band 4 image, its noisy version ($\sigma = 500$), the DA-denoising result, and the MM-denoising result are displayed from left to right. The DA result has an MSE of 3.1e4, whereas the best global soft thresholding result, MM, has an MSE of 4.1e4. In this case, it is clear that as the MSE increases from 3.1e4 to 4.1e4, the quality of the denoised image quickly degrades.

In addition to global thresholding, the author performed multi-level soft thresholding of several noisy image bands using an interactive wavelet denoising tool in MATLAB and was unable to produce better looking imagery than the DA results.

Additional comments. In the sense of fairness, we discuss the downsides of our algorithm. First, there is currently no automated method to find a good value for the regularization parameter, λ. Thus, the author experimented with different choices of λ, ultimately setting it equal to 0.153 and 0.160 for the $\sigma = 250$ and $\sigma = 500$ experiments, respectively. Second, our algorithm takes several minutes to run on a personal computer, whereas the soft thresholding methods take only seconds.

Junk band recovery experiment. For this experiment, we identify bands 104 through 113 as junk and bands 114 through 135 as good. The top row of Figure 2 shows junk bands 104, 108, 110, and 111. We run the junk band recovery algorithm and analyze the results qualitatively.

Junk band recovery results. After running the algorithm, each of the junk bands is denoised to some extent. Referring to Figure 2, band 104 seems less noisy. Band 108's original version is almost totally obscured by noise, whereas the recovered version has noticeable edges and features. The denoised junk bands might now be useful in classification systems.
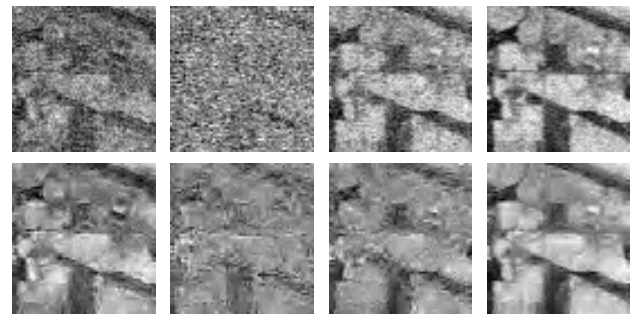

Fig. 2. Junk Band Recovery Experiment. Top row (left to right): junk bands 104, 108, 110, and 110. Bottom row: the results of the junk band recovery algorithm for the same four bands.

V. CONCLUSIONS

We have effectively denoised hyperspectral data and recovered junk bands by imposing constraints on the wavelet representations of hyperspectral images, exploiting the inherent correlation across band images via an optimization program. Our denoising algorithm outperforms widely used global and multi-level wavelet thresholding techniques in an MSE and a qualitative sense.

This work poses several interesting questions: Is it possible to avoid decomposing the optimization problem into many subproblems? Can the algorithm's performance be improved by using a different type of wavelet? Is there a fast, automated method to determine λ? Most importantly, will recovering junk bands lead to significant improvements in classifier performance?

REFERENCES

[1] I. Atkinson, F. Kamalabadi, and D. L. Jones. Wavelet-Based Hyperspectral Image Estimation. *Proc. IGARSS Conf.*, 1993.

[2] S. G. Chang, B. Yu, and M. Vetterli. Adaptive Wavelet Thresholding for Image Denoising and Compression. *IEEE Trans. Image Proc.*, 9(9):1532–1546, Sept. 2000.

[3] S. S. Chen, D. L. Donoho, and M. A. Saunders. Atomic Decomposition by Basis Pursuit. *SIAM Review*, 43(1):129–159, 2001.

[4] G. Davis, S. Mallat, and M. Avellaneda. Greedy adaptive approximation. *Constr. Approx.*, 13:57–98, 1997.

[5] D. L. Donoho. De-noising by soft-thresholding. *IEEE Trans. Inform. Theory*, 41:613–627, May 1995.

[6] D. L. Donoho and I. M. Johnstone. Ideal spatial adaptation via wavelet shrinkage. *Biometrika*, 81:425–455, 1994.

[7] S. Kaewpijit, J. LeMoigne, and T. El-Ghazawi. A Wavelet-based PCA Reduction for Hyperspectral Imagery. *Proc. IGARSS Conf.*, 2002.

[8] NASA Jet Propulsion Laboratory. Airborne Visible/Infrared Imaging Spectrometer. Online. http://aviris.jpl.nasa.gov/.

[9] D. Landgrebe and L. Biehl. MultiSpec Software. Online. http://dynamo.ecn.purdue.edu/~biehl/MultiSpec/.

[10] D. M. Malioutov, M. Çetin, and A. S. Willsky. Sparse Signal Reconstruction Perspective for Source Localization with Sensor Arrays. *IEEE Trans. Sig. Proc.*, 53(8):3010–3022, Aug. 2005. Part II.

[11] S. G. Mallat. A Theory for Multiresolution Signal Decomposition: A Wavelet Representation. *IEEE Trans. Pattern Analysis and Machine Intel.*, 11(7), July 1989.

[12] C. A. Shah, P. Watanachaturaporn, and P. K. Varshney. Some Recent Results on Hyperspectral Image Classification. *IEEE Workshop on Adv. in Tech. for Analy. of Remotely Sensed Data*, Oct. 2003.

[13] R. B. Smith. Introduction to Hyperspectral Imaging. Online. http://www.microimages.com/getstart/pdf/hyprspec.pdf.

[14] J. Sturm. SeDuMi: Self-Dual-Minimisation package for MATLAB. Online. http://sedumi.mcmaster.ca/.

[15] M. Vetterli and J. Kovačević. *Wavelets and Subband Coding*. Sig. Proc. Series. Prentice Hall, Englewood Cliffs, NJ, 1995.