LOW DIMENSIONAL MANIFOLD MODEL IN HYPERSPECTRAL IMAGE RECONSTRUCTION [*]

ZUOQIANG SHI [†], WEI ZHU [‡], AND STANLEY OSHER [§]

Abstract. We present the application of a low dimensional manifold model (LDMM) on hyperspectral image (HSI) reconstruction. An important property of hyperspectral images is that the patch manifold, which is sampled by the three-dimensional blocks in the data cube, is generally of a low dimensional nature. This is a generalization of low-rank models in that hyperspectral images with nonlinear mixing terms can also fit in this framework. The point integral method (PIM) is used to solve a Laplace-Beltrami equation over a point cloud sampling the patch manifold in LDMM. Both numerical simulations and theoretical analysis show that the sample points constraint is correctly enforced by PIM. The framework is demonstrated by experiments on the reconstruction of both linear and nonlinear mixed hyperspectral images with a significant number of missing voxels and several entirely missing spectral bands.

1. Introduction.

Hyperspectral imagery is an important domain in remote sensing with numerous applications [3]. A hyperspectral image (HSI) is a set of images of the same geographic location taken at up to 200 different wavelengths. When such data of high spatial-spectral dimensionality are collected, it is not uncommon that they are degraded. For instance, some of the voxels, or sometimes several spectral bands, are missing due to the malfunctions of the hyperspectral cameras. Typically, the measuring process can be formulated as:

$$y = \Phi_I f + \varepsilon, \tag{1.1}$$

where f and y are the "original" and "observed" HSI data cube, ε is the additive noise, and Φ_I is the projection operator corresponding to the missing data. More specifically,

$$\Phi_I f(x) = \begin{cases} f(x) & ,\boldsymbol{x} \in I \\ 0 & ,\boldsymbol{x} \notin I. \end{cases} \tag{1.2}$$

An important task in HSI analysis is to recover the original data cube from the incomplete data. It is an ill-posed problem to recover f from (1.1), and some prior knowledge of the original data cube f must be exploited.

One commonly used prior information of HSI is that the three-dimensional (3D) data cube is of a low-rank nature under the linear mixing model (LMM) [1]. More specifically, the spectral signature of each pixel is assumed to be a linear combination of a few constituent spectra, called endmemers. Under such an assumption, sparse representation techniques have been used to reconstruct the original HSI [4, 9, 18, 19]. However, such approaches all assume the low-rank property of the hyperspectral images, which results from the LMM. Despite the simplicity of LMM, the linear mixing assumption has been shown to be physically inaccurate in certain situations [5].

[*]Research supported by DOE-SC0013838 and NSF DMS-1118971. Z. Shi was partially supported by NSFC Grant 11371220.

[†]Yau Mathematical Sciences Center, Tsinghua University, Beijing, China, 100084. *Email: zqshi@math.tsinghua.edu.cn.*

[‡]Department of Mathematics, University of California, Los Angeles, CA 90095, USA, *Email: weizhu731@math.ucla.edu*

[§]Department of Mathematics, University of California, Los Angeles, CA 90095, USA, *Email: sjo@math.ucla.edu*

Graph based methods with various regularizations can also be applied to recover f from (1.1). A well-studied and commonly used model is the total variation (TV) method introduced by Rudin, Osher, and Fatemi in 1992 [14]:

$$\min_f \|\nabla f\|_{L^1} + \lambda \|y - \Phi_I f\|_2^2, \tag{1.3}$$

where $\|\nabla f\|_{L^1}$ is the total variation semi-norm. The total variation method has the advantage of recovering images while keeping the edges intact. However, this is a local method in the sense that the gradient of a pixel is computed by its immediate adjacent pixels. It is known that local image processing methods fail to achieve satisfactory results for images with repetitive structures. Therefore nonlocal methods are usually preferred to process images with textures. Buades, Coll, and Morel introduced nonlocal means as a filter for image denoising [2]. Zhou and Schölkopf also constructed a theory of nonlocal operators in their work [20]. Gilboa and Osher [6] later formalized a systematic framework for nonlocal image processing. In essence, nonlocal methods replace the local gradient with its nonlocal counterpart $\nabla_w u \in L^2(\Omega, L^2(\Omega))$:

$$\nabla_w u(x)(y) = \sqrt{w(x,y)}(u(x) - u(y)), \tag{1.4}$$

where $w(x,y)$ is the weight between x and y, and is defined as the similarity between two patches around x and y. The nonlocal total variation (NLTV) model for problem (1.1) becomes:

$$\min_f \|\nabla_w f\|_{L^1} + \lambda \|y - \Phi_I f\|_2^2. \tag{1.5}$$

Nonlocal methods have been shown to achieve better results than local methods in image processing problems, and they have also been applied to HSI classification [8, 11, 21]. However, nonlocal methods do not perform very well on image reconstruction problems with a significant number of uniformly distributed missing pixels (over 90%) [13], and the reason is that the sample points constraint is not correctly enforced in nonlocal methods.

In [13], the authors proposed a low dimensional manifold model (LDMM) for general image processing problems. In the core of the model, the dimension of the patch manifold is utilized as a regularization to recover the image. The key step of the model is to solve the following Laplace-Beltrami equation over the manifold $\mathcal{M}$:

$$\begin{cases} -\Delta_{\mathcal{M}} u(\boldsymbol{x}) + \mu \sum_{\boldsymbol{y} \in \Omega} \delta(\boldsymbol{x} - \boldsymbol{y})(u(\boldsymbol{y}) - v(\boldsymbol{y})) = 0, & \boldsymbol{x} \in \mathcal{M} \\ \dfrac{\partial u}{\partial \mathbf{n}}(\boldsymbol{x}) = 0, & \boldsymbol{x} \in \partial\mathcal{M}, \end{cases} \tag{1.6}$$

where Ω is the patch set spanned by the image, and $\mathcal{M}$ is the underlying patch manifold embedded in the Euclidean space. The above equation (1.6) is solved by the point integral method (PIM)[10, 15, 16]. The PIM approximates (1.6) with an integral equation instead of discretizing the Laplace operator using the traditional graph Laplacian, which was found to be inconsistent due to the boundary conditions. LDMM achieves excellent results, especially in image inpainting problems where the number of missing pixels is large.

In this paper, we extend the idea in [13] and apply LDMM to the reconstruction of HSI with a large number of missing voxels and several entirely missing spectral bands. The low dimensional nature of the patch manifold of HSI is exploited and

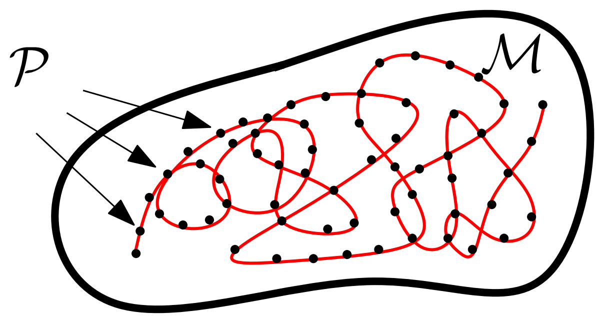

FIG. 1. *Diagram of patch set $\mathcal{P}$ (black points), trivial parameterization (red curve) and patch manifold $\mathcal{M}$.*

good recovery results have been achieved on both linearly and nonlinearly mixed HSI. LDMM can be used as a reconstruction method for severely degraded HSI, and it can also be utilized in the data collection process in the first place (i.e. purposefully measuring only a very small portion of the data).

The paper is organized as follows: In Section 2 we define the patch manifold of HSI and analyze its low dimensionality. In Section 3, we propose the low dimensional manifold model, and discuss the numerical implementation of LDMM including the point integral method, discretization, etc. Different choices of the patch manifold and proper initializations are discussed in Section 4. Numerical results are presented in Section 5. Finally, we draw conclusions in Section 6.

2. Patch Manifold.

In this section, we describe the patch manifold of HSI and analyze its dimensionality. Assume we have a hyperspectral image $f \in \mathbb{R}^{m\times n\times B}$, where $m \times n$ is the spatial dimension of the image and B is the length of the spectral signatures. For any $(i, j) \in \{1, 2, \dots, m\} \times \{1, 2, \dots, n\}$, we define a patch $p_{ij}(f)$ as a 3D block of size $s_1 \times s_2 \times B$ of the original data cube f, and the pixel (i, j) is the top-left corner of the rectangle of size $s_1 \times s_2$. The patch set $\mathcal{P}(f)$ is defined as the collection of all patches:

$$\mathcal{P}(f) = \{p_{ij}(f) : (i, j) \in \{1, 2, \dots, m\} \times \{1, 2, \dots, n\}\} \subset \mathbb{R}^d, \quad d = s_1 \times s_2 \times B. \tag{2.1}$$

The patch set $\mathcal{P}(f)$ has a trivial 2D parameterization which is given as $(i, j) \mapsto p_{ij}(f)$. In this sense, the patch set is locally a 2D sub-manifold embedded in $\mathbb{R}^d$. However, this parameterization is globally not injective and typically leads to high curvature variations and self-intersections in real applications.

For a given hyperspectral image f, the patch set $\mathcal{P}(f)$ gives a point cloud in $\mathbb{R}^d$. It is found that this point cloud is usually close to a smooth manifold $\mathcal{M}(f)$ embedded in $\mathbb{R}^d$. This underlying smooth manifold is called the *patch manifold* associated with f, denoted as $\mathcal{M}(f)$. Fig. 1 gives a diagram shows the relation between patch set, trival parameterization and patch manifold.

Besides the manifold structure, it is found that for real-life hyperspectral images, the patch manifold $\mathcal{M}(f)$ is always of low dimensionality. In the following analysis, we assume that f is a discrete sampling of a continuous function $\tilde{f} : [0, 1]^3 \to \mathbb{R}$. More specifically, $f(i, j, k) = \tilde{f}(x_i, y_j, t_k)$, where $(x_i, y_j, t_k) = (i\Delta_x, j\Delta_y, k\Delta_t)$.

Under the linear mixing model (LMM), there is a small collection of constituent elements (endmembers) $e_l \in L^2([0, 1]), l = 1, 2, \dots, K$ that generate the entire image

$\tilde{f} \in L^2([0,1]^3)$. More specifically,

$$\tilde{f}(x,y,t) = \sum_{l=1}^{K} \beta_l(x,y) e_l(t), \quad \beta_l(x,y) \geq 0. \tag{2.2}$$

Then the patch $p_{(x,y)}f \in \mathbb{R}^d$ can be written down as:

$$p_{(x,y)}f(i,j,k) = \tilde{f}(x+x_i, y+y_j, t_k) = \sum_{l=1}^{K} \beta_l(x+x_i, y+y_j) e_l(t_k). \tag{2.3}$$

In practice, we usually choose the size of the patches $s_1 \times s_2$ small enough to be consistent with the spatial resolution of the HSI. So locally inside the patch, $\beta_l(x+x_i, y+y_j)$ can be approximated by Taylor expansion:

$$p_{(x,y)}f(i,j,k) \approx \sum_{l=1}^{K} \left(\beta_l(x,y) + \frac{\partial \beta_l}{\partial x}(x,y) \cdot x_i + \frac{\partial \beta_l}{\partial y}(x,y) \cdot y_j \right) e_l(t_k) \tag{2.4}$$

$$= \sum_{l=1}^{K} \left(\beta_l(x,y) + i \frac{\partial \beta_l}{\partial x}(x,y)\Delta_x + j \frac{\partial \beta_l}{\partial y}(x,y)\Delta_y \right) e_l(k\Delta_t). \tag{2.5}$$

Therefore the underlying patch manifold $\mathcal{M}(f)$ can be approximated by a manifold of dimension $3K$.

If we are taking into account the more intimate mixture by considering the bilinear mixing model (BMM), then $\tilde{f}$ can be expressed as:

$$\tilde{f}(x,y,t) = \sum_{u=1}^{K} \beta_u(x,y) e_u(t) + \sum_{u=1}^{p-1} \sum_{v=u+1}^{p} \gamma_{uv}(x,y) \beta_u(x,y) \beta_v(x,y) e_u(t) \cdot e_v(t). \tag{2.6}$$

Again, the same analysis as above shows that $\mathcal{M}(f)$ under BMM is of dimension $(3K + \frac{3}{2}(K^2 - K))$

Of course, more complicated mixing models yield higher dimensions of the patch manifolds, but the intrinsic dimension of the manifold is still smaller than d, which is the dimension of the Euclidean space in which the manifold is embedded. Moreover, in reality, the intimate mixture between multiple endmembers can usually be neglected. This means that it is a natural idea to use the low dimensionality of the patch manifold as a prior knowledge to reconstruct the HSI.

We make one remark before ending this section. In practice there are different ways to choose the patch size $s_1 \times s_2 \times s_3$ to cater to different problems at hand. In our discussion above, s_1 and s_2 are chosen to be consistent with the spatial resolution, and s_3 is chosen to be B, which is the length of the third dimension of the original data cube. But sometimes s_3 can be chosen to be smaller than B. More details about the patch sizes are discussed in Section 4.

3. Low Dimensional Manifold Model.

In this paper, we focus more on the "inpainting" part of the reconstruction. That is, we assume the data is not contaminated by noise and the incomplete data are generated by the following process:

$$y = \Phi_I f, \tag{3.1}$$

where Φ_I is defined in (1.2). Based on the discussion in the previous section it is natural to utilize the dimension of the patch manifold as a regularization in the following variational problem:

$$\min_{\substack{f\in\mathbb{R}^{m\times n\times B},\\ \mathcal{M}\subset\mathbb{R}^d}} \quad \dim(\mathcal{M}), \qquad \text{subject to:} \quad y = \Phi_I f, \quad \mathcal{P}(f)\subset\mathcal{M}. \tag{3.2}$$

However, (3.2) is not well defined in the sense that we have not yet derived an explicit formula to evaluate $\dim(\mathcal{M})$. Fortunately, there is a very simple formula to calculate the dimension of a smooth manifold [13]:

PROPOSITION 3.1. *Let $\mathcal{M}$ be a smooth submanifold isometrically embedded in $\mathbb{R}^d$. For any $\boldsymbol{x}\in\mathcal{M}$,*

$$\dim(\mathcal{M}) = \sum_{j=1}^{d} \|\nabla_{\mathcal{M}}\alpha_j(\boldsymbol{x})\|^2,$$

where $\alpha_i, i = 1,\ldots,d$ are the coordinate functions on $\mathcal{M}$, i.e.

$$\alpha_i(\boldsymbol{x}) = x_i, \quad \forall \boldsymbol{x} = (x_1,\ldots,x_d)\in\mathcal{M}.$$

Hence (3.2) can be transformed into the following LDMM:

$$\min_{\substack{f\in\mathbb{R}^{m\times n\times B},\\ \mathcal{M}\subset\mathbb{R}^d}} \quad \sum_{i=1}^{d}\|\nabla_{\mathcal{M}}\alpha_i\|^2_{L^2(\mathcal{M})} + \delta_{\Phi_I^{-1}(y)}(f), \qquad \text{subject to:} \quad \mathcal{P}(f)\subset\mathcal{M}, \tag{3.3}$$

where $\Phi_I^{-1}(y)$ is the pre-image of y, and $\delta_{\Phi_I^{-1}(y)}$ is the indicator function. More specifically:

$$\Phi_I^{-1}(y) = \{f : \Phi_I(f) = y\}, \quad \delta_{\Phi_I^{-1}(y)}(f) = \begin{cases} 0 & , f\in\Phi_I^{-1}(y) \\ \infty & , f\notin\Phi_I^{-1}(y). \end{cases} \tag{3.4}$$

We point out that (3.2) and (3.3) are not exactly the same, since the regularizer in (3.3) is actually the L^1 norm of the local dimension over the manifold $\mathcal{M}$. This is actually favorable in the context of HSI reconstruction for two reasons. First, the patch manifold $\mathcal{M}$ is not necessarily a smooth manifold, and taking the integral of local dimension of $\mathcal{M}$ can address the situation when $\mathcal{M}$ is only piecewise smooth. Second, for high-resolution HSI, there might be several materials that are not mixed with the others, which means the patch manifold $\mathcal{M}$ has zero local dimension at those points (patches spanned by "pure pixels"); the L^1 regularizer of the local dimension, which is known to promote sparsity, correctly uses this prior knowledge.

3.1. Numerical Implementation. The numerical procedure to solve (3.3) is similar to that in [13], and we write it down here again for readers not familiar with LDMM. It is hard to solve problem (3.3) directly with respect to both $\mathcal{M}$ and the image f. So we use an iterative method instead. In every iteration, we first fix the manifold $\mathcal{M}$ and use the metric defined by $\mathcal{M}$ to update the image f; then the manifold is updated by enforcing the fact that the patch set of the new image f samples the new manifold $\mathcal{M}$.

In practice, we find it a lot easier to perturb the coordinate function α_i as well. More specifically, we have the following iterative procedure:

- Assume we have the n-th iterates f^n and $\mathcal{M}^n$ satisfying $\mathcal{P}(f^n) \subset \mathcal{M}^n$.
- Fix the manifold $\mathcal{M}^n$, and update f^{n+1} and the perturbed coordinate functions $\alpha_i^{n+1}, i = 1, \ldots, d$ by solving:

(3.5)
$$(f^{n+1}, \alpha_1^{n+1}, \ldots, \alpha_d^{n+1}) = \arg \min_{\substack{f \in \mathbb{R}^{m \times n \times B}, \\ \alpha_1, \ldots, \alpha_d \in H^1(\mathcal{M}^n)}} \sum_{i=1}^{d} \|\nabla_{\mathcal{M}^n} \alpha_i\|^2_{L^2(\mathcal{M}^n)} + \delta_{\Phi_I^{-1}(y)}(f),$$
$$\text{subject to:} \quad \boldsymbol{\alpha}(\mathcal{P}(f^n)) = \mathcal{P}(f)$$

where $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_d)^T$ is the perturbed coordinate function. The constraint means that the coordinate function $\boldsymbol{\alpha}$ maps the original patch set $\mathcal{P}(f^n)$ into the patch set of the new image $\mathcal{P}(f^{n+1})$.
- Update $\mathcal{M}$ by setting

$$\mathcal{M}^{n+1} = \boldsymbol{\alpha}(\mathcal{M}^n) = \left\{ (\alpha_1^{n+1}(\boldsymbol{x}), \ldots, \alpha_d^{n+1}(\boldsymbol{x}))^T : \boldsymbol{x} \in \mathcal{M}^n \right\}. \tag{3.6}$$

- Repeat the process until convergence.

We first point out the legitimacy of perturbing the coordinate function $\boldsymbol{\alpha}$. As discussed in Proposition 3.1, $\boldsymbol{\alpha}$ should be a function defined on $\mathcal{M}$ which maps $\boldsymbol{x}$ into itself, so it makes no sense at first glance to make $\boldsymbol{\alpha}$ a variable. But it is not hard to observe that when the iteration converges, $\mathcal{M}^{n+1}$ is very close to $\mathcal{M}^n$ and so is $\mathcal{P}(f^{n+1})$ to $\mathcal{P}(f^n)$. Therefore $\boldsymbol{\alpha}$ is getting closer and closer to the identity. Moreover, perturbed $\boldsymbol{\alpha}$ helps enforce the constraint $\mathcal{P}(f^{n+1}) \subset \mathcal{M}^{n+1}$ as can be seen from (3.5) and (3.6).

Notice that (3.6) is easy to implement, whereas (3.5) is a much more complicated constrained optimization problem. We use the split Bregman iteration [12, 7] to enforce the linear constraint and update α_i and f in (3.5) sequentially.

- Given k-th iterates $\boldsymbol{\alpha}^{n+1,k}$, $f^{n+1,k}$ and d^k.
- Update $\alpha_i^{n+1,k+1}$, $i = 1, \ldots, d$ with fixed $f^{n+1,k}$ and d^k

(3.7)
$$\begin{aligned}&(\alpha_1^{n+1,k+1}, \ldots, \alpha_d^{n+1,k+1}) \\ &= \arg \min_{\alpha_1, \ldots, \alpha_d \in H^1(\mathcal{M}^n)} \sum_{i=1}^{d} \|\nabla \alpha_i\|^2_{L^2(\mathcal{M}^n)} + \mu \|\boldsymbol{\alpha}(\mathcal{P}(f^n)) - \mathcal{P}(f^{n+1,k}) + d^k\|^2_{\mathrm{F}},\end{aligned}$$

where both the patch set $\mathcal{P}(f^{n+1,k})$ and the image of the patch set under the perturbed coordinate functions $\boldsymbol{\alpha}(\mathcal{P}(f^n))$ are treated as matrices in the space $\mathbb{R}^{d \times N}$. More specifically, every column of $\mathcal{P}(f)$ is one patch in the patch set $\mathcal{P}(f)$, and

$$\boldsymbol{\alpha}(\mathcal{P}(f^n)) = \begin{pmatrix} \alpha_1(\mathcal{P}(f^n)) \\ \alpha_2(\mathcal{P}(f^n)) \\ \vdots \\ \alpha_d(\mathcal{P}(f^n)) \end{pmatrix} \in \mathbb{R}^{d \times N}, \quad N = |\mathcal{P}(f^n)|.$$

- Update $f^{n+1,k+1}$ with fixed $\boldsymbol{\alpha}^{n+1,k+1}$ and d^k

(3.8)
$$f^{n+1,k+1} = \arg \min_{f \in \mathbb{R}^{m \times n \times B}} \quad \delta_{\Phi_I^{-1}(y)}(f) + \mu \|\boldsymbol{\alpha}^{n+1,k+1}(\mathcal{P}(f^n)) - \mathcal{P}(f) + d^k\|^2_{\mathrm{F}}.$$

- Update d^{k+1},

$$d^{k+1} = d^k + \boldsymbol{\alpha}^{n+1,k+1}(\mathcal{P}(f^n)) - \mathcal{P}(f^{n+1,k+1}).$$

We first show how to solve (3.8). This is a constrained least-square problem:

$$f^{n+1,k+1} = \arg \min_{\substack{f\in\mathbb{R}^{m\times n\times B},\\ \Phi_I(f)=y}} \quad \|\boldsymbol{\alpha}^{n+1,k+1}(\mathcal{P}(f^n)) - \mathcal{P}(f) + d^k\|_{\mathrm{F}}^2, \tag{3.9}$$

which has an analytic solution:

$$f^{n+1,k+1}(\boldsymbol{x}) = \left\{ \begin{array}{cc} y(\boldsymbol{x}), & \boldsymbol{x}\in I, \\ (\mathcal{P}^*\mathcal{P})^{-1}(\mathcal{P}^*(\boldsymbol{\alpha}^{n+1,k+1}(\mathcal{P}(f^n)) + d^k))(\boldsymbol{x}), & \boldsymbol{x}\notin I, \end{array}\right. \tag{3.10}$$

where $\mathcal{P}^*$ is the adjoint operator of $\mathcal{P}$. It is worth mentioning that $\mathcal{P}^*\mathcal{P}$ is a diagonal operator, so (3.10) can be solved efficiently.

Next, we talk about how to solve (3.7). Notice that in (3.7), $\alpha_i^{n+1,k+1},\ i=1,\dots,d$ are decoupled and can be solved separately,

$$\alpha_i^{n+1,k+1} = \arg \min_{\alpha_i\in H^1(\mathcal{M}^n)} \quad \|\nabla\alpha_i\|^2_{L^2(\mathcal{M}^n)} + \mu\|\alpha_i(\mathcal{P}(f^n)) - \mathcal{P}_i(f^{n+1,k}) + d_i^k\|^2, \tag{3.11}$$

where $\mathcal{P}_i(f^n)$ is the i-th row of the matrix $\mathcal{P}(f^n)$.

To sum up the discussion above, we have a continuous version of LDMM as in Algorithm 1.

The key step in Algorithm 1 is to solve optimization problems of the form:

$$\min_{u\in H^1(\mathcal{M})} \quad \|\nabla_{\mathcal{M}}u\|^2_{L^2(\mathcal{M})} + \mu\sum_{\boldsymbol{y}\in\Omega}|u(\boldsymbol{y}) - v(\boldsymbol{y})|^2, \tag{3.12}$$

where u can be any α_i, $\mathcal{M}=\mathcal{M}^n$, $\Omega=\mathcal{P}(f^n)$ and $v(\boldsymbol{y})$ is a given function on Ω.

By standard variational methods, we know problem (3.12) is equivalent to the following PDE:

$$\left\{ \begin{array}{ll} -\Delta_{\mathcal{M}}u(\boldsymbol{x}) + \mu\displaystyle\sum_{\boldsymbol{y}\in\Omega}\delta(\boldsymbol{x}-\boldsymbol{y})(u(\boldsymbol{y}) - v(\boldsymbol{y})) = 0, & \boldsymbol{x}\in\mathcal{M} \\ \dfrac{\partial u}{\partial \mathbf{n}}(\boldsymbol{x}) = 0, & \boldsymbol{x}\in\partial\mathcal{M}. \end{array}\right. \tag{3.13}$$

Notice that problem (3.13) is a Laplace equation over an unstructured point cloud $\mathcal{P}(f^n)$ sampling the manifold $\mathcal{M}$, which can be solved correctly and efficiently by the point integral method (PIM) [10, 15, 16].

3.1.1. Point Integral Method. In order to solve (3.13) over a point cloud, we need to discretize the Laplace-Beltrami operator and the δ-function. One might be tempted to use the graph-Laplacian to discretize $\Delta_{\mathcal{M}}$, but the consistency of the graph-Laplacian does not hold true when the boundary $\partial\mathcal{M}$ of $\mathcal{M}$ is nonempty [10, 15, 16, 13]. Moreover, the δ-functions in (3.13) are all supported on the sample point set $\mathcal{P}(f^n)$, so it is difficult to discretize them directly. Therefore, it seems natural to convolve (3.13) with a test function R_t:

$$\int_{\mathcal{M}} -\Delta_{\mathcal{M}}u(\boldsymbol{y})R_t(\boldsymbol{x},\boldsymbol{y}) + \mu\sum_{\boldsymbol{z}\in\Omega}\delta(\boldsymbol{y}-\boldsymbol{z})(u(\boldsymbol{z}) - v(\boldsymbol{z}))R_t(\boldsymbol{x},\boldsymbol{y})\mathrm{d}\boldsymbol{y} = 0, \tag{3.14}$$

Algorithm 1 LDMM Algorithm - Continuous version

```
Require: Initial guess of the image $f^0$, $d^0 = 0$.
Ensure: Restored image $f$.
1: while not converge do
2:   while not converge do
3:     With fixed manifold $\mathcal{M}^n$, for $i = 1, \cdots, d$, solve
         $\alpha_i^{n+1,k+1} = \arg\min_{\alpha_i \in H^1(\mathcal{M}^n)} \|\nabla_{\mathcal{M}^n}\alpha_i\|^2_{L^2(\mathcal{M}^n)} + \mu\|\alpha_i(\mathcal{P}(f^n)) - \mathcal{P}_i(f^{n+1,k}) + d_i^k\|^2.$
4:     Update $f^{n+1,k+1}$,
         $f^{n+1,k+1} = \arg\min_{f \in \mathbb{R}^{m\times n\times B}} \delta_{\Phi_I^{-1}(y)}(f) + \mu\|\boldsymbol{\alpha}^{n+1,k+1}(\mathcal{P}(f^n)) - \mathcal{P}(f) + d^k\|^2_{\mathrm{F}}.$
5:     Update $d^{k+1}$,
         $d^{k+1} = d^k + \boldsymbol{\alpha}^{n+1,k+1}(\mathcal{P}(f^n)) - \mathcal{P}(f^{n+1,k+1}).$
6:   end while
7:   Set $f^{n+1} = f^{n+1,k}$, $\boldsymbol{\alpha}^{n+1} = \boldsymbol{\alpha}^{n+1,k}$.
8:   Update $\mathcal{M}$
         $\mathcal{M}^{n+1} = \left\{(\alpha_1^{n+1}(\boldsymbol{x}), \cdots, \alpha_d^{n+1}(\boldsymbol{x})) : \boldsymbol{x} \in \mathcal{M}^n\right\}.$
9: end while
```

where $t > 0$ is a positive parameter, and R_t is the Gaussian:

$$R_t(\boldsymbol{x}, \boldsymbol{y}) = C_t \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{y}|^2}{4t}\right). \tag{3.15}$$

C_t is the normalizing factor for the Gaussian.

Then we can use the point integral method to solve (3.14). The key observation in PIM is that the convolution of $\Delta_{\mathcal{M}} u$ and a Gaussian can be approximated by:

$$\int_{\mathcal{M}} \Delta_{\mathcal{M}} u(\boldsymbol{y}) R_t(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{y} \approx -\frac{1}{t}\int_{\mathcal{M}} (u(\boldsymbol{x}) - u(\boldsymbol{y})) R_t(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{y} + 2\int_{\partial\mathcal{M}} \frac{\partial u(\boldsymbol{y})}{\partial \mathbf{n}} R_t(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\tau_{\boldsymbol{y}}. \tag{3.16}$$

More rigorously, we have the following theorem:

Theorem 3.2. *If $u \in C^3(\mathcal{M})$ is a function on $\mathcal{M}$, then we have for any $\boldsymbol{x} \in \mathcal{M}$,*

$$\|r(u)\|_{L^2(\mathcal{M})} = O(t^{1/4}), \tag{3.17}$$

where

$$r(u) = \int_{\mathcal{M}} \Delta u(\boldsymbol{y}) R_t(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{y} + \frac{1}{t}\int_{\mathcal{M}} (u(\boldsymbol{x}) - u(\boldsymbol{y})) R_t(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\boldsymbol{y} - 2\int_{\partial\mathcal{M}} \frac{\partial u(\boldsymbol{y})}{\partial \mathbf{n}} R_t(\boldsymbol{x}, \boldsymbol{y}) \mathrm{d}\tau_{\boldsymbol{y}}.$$

The detailed proof can be found in [15].

Combining (3.13)(3.14)(3.16), we have the following integral equation to approximate (3.13)

$$\int_{\mathcal{M}} (u(\boldsymbol{x}) - u(\boldsymbol{y}))R_t(\boldsymbol{x}, \boldsymbol{y})\mathrm{d}\boldsymbol{y} + \mu t \sum_{\boldsymbol{y}\in\Omega} R_t(\boldsymbol{x}, \boldsymbol{y})(u(\boldsymbol{y}) - v(\boldsymbol{y})) = 0. \tag{3.18}$$

Notice that only point value information of u is needed in the above equation instead of $\Delta_{\mathcal{M}}$, and we no longer have to discretize the δ-functions.

3.1.2. Discretization. Next we discuss how to discretize (3.18) over the point cloud $\Omega = \mathcal{P}(f^n)$. Let $\mathcal{P}(f^n) = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$, and we assume $\mathcal{P}(f^n)$ samples the patch manifold $\mathcal{M}$ uniformly at random. It is easy to discretize (3.18) into the following linear system:

$$\frac{|\mathcal{M}|}{N}\sum_{j=1}^{N} R_t(\boldsymbol{x}_i, \boldsymbol{x}_j)(u(\boldsymbol{x}_i) - u(\boldsymbol{x}_j)) + \mu t \sum_{j=1}^{N} R_t(\boldsymbol{x}_i, \boldsymbol{x}_j)(u(\boldsymbol{x}_j) - v(\boldsymbol{x}_j)) = 0, \tag{3.19}$$

where $|\mathcal{M}|$ is the volume of $\mathcal{M}$.

(3.19) can be rewritten in the following matrix form:

$$(\boldsymbol{L} + \bar{\mu}\boldsymbol{W})\boldsymbol{u} = \bar{\mu}\boldsymbol{W}\boldsymbol{v}, \tag{3.20}$$

where $\boldsymbol{u} = (u(\boldsymbol{x}_1), \ldots, u(\boldsymbol{x}_N))^T$,$\boldsymbol{v} = (v(\boldsymbol{x}_1), \ldots, v(\boldsymbol{x}_N))^T$, and $\bar{\mu} = \frac{\mu t N}{|\mathcal{M}|}$. $\boldsymbol{L}$ is defined as the difference between $\boldsymbol{D}$ and $\boldsymbol{W}$:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}, \tag{3.21}$$

where $\boldsymbol{W} = (w_{ij})$ is the weight matrix with $w_{ij} = R_t(\boldsymbol{x}_i, \boldsymbol{x}_j)$. $\boldsymbol{D} = \mathrm{diag}(d_i)$ is the degree matrix with $d_i = \sum_{j=1}^{N} w_{ij}$. So now we know how to update $\alpha_i^{n+1,k+1}$ in Algorithm 1.

Next, we discuss about the computation of the weight matrix $\boldsymbol{W}$. Since we need to solve the large-scale linear system (3.20), it is preferable if $\boldsymbol{W}$ is a sparse matrix. We use the approximate nearest neighbor (ANN) scheme to efficiently compute the sparse weight matrix $\boldsymbol{W}$ [21]. In the core of ANN search scheme, a balanced k-dimensional tree is built, and an upper bound on the number of distance comparisons are placed to approximately compute the nearest neighbors of every point in $\mathcal{P}(f^n)$. Interested readers are referred to [21] for a detailed explanation of ANN search scheme.

To summarize the discussion in this section, we write down Algorithm 2 for HSI reconstruction.

4. Different Choices for Patch Size and Proper Initializations.

As discussed in Section 2, there is a lot of freedom in choosing the patch size $s_1 \times s_2 \times s_3$. Usually $s_1 \times s_2$ is chosen to be compatible with the spatial resolution, and smaller s_3 leads to higher accuracy and more computational time.

When dealing with the problem of HSI reconstruction, we only have the degraded image y to begin with, and the objective is to fill in the missing data. We found out that choosing $s_3 = B$ usually leads to unsatisfactory recovering results, especially when the number of the missing data is significant. To explain the reason why large s_3 leads to worse recovering results, let us assume for simplicity that $s_1 = s_2 = 1$ and $s_3 = B$. In this case, every patch is the spectral signature of a pixel, and the size of $\mathcal{P}(f)$ is $B \times N$, where N is the number of pixels in the 2D spatial dimension.

Algorithm 2 LDMM Algorithm

```
Require: Initial guess of the image $f^0$, $d^0 = 0$.
Ensure: Restored image $f$.
1: while not converge do
2:   Compute the weight matrix $\boldsymbol{W} = (w_{ij})_{1\le i,j\le N}$ from $\mathcal{P}(f^n)$, where $N = |\mathcal{P}(f^n)|$,
       $$w_{ij} = R_t(\boldsymbol{x}_i, \boldsymbol{x}_j), \quad \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathcal{P}(f^n), \quad i,j = 1,\dots,N.$$
     Assemble the matrices $\boldsymbol{L}$, $\boldsymbol{W}$ as follows:
       $$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W},\ \boldsymbol{W} = (w_{i,j}),\ i,j = 1,\cdots,N.$$
3:   for $k = 1 : K$ do
4:     Solve the following linear systems
         $$(\boldsymbol{L} + \bar{\mu}\boldsymbol{W})\boldsymbol{U}_k = \bar{\mu}\boldsymbol{W}\boldsymbol{V}_{k-1},$$
       where $\boldsymbol{V}_k = \left(\mathcal{P}(f^n) - d^k\right)^T$.
5:     Update $f$ by solving a constrained least square problem.
         $$f^{n+1,k}(\boldsymbol{x}) = \begin{cases} y(\boldsymbol{x}), & \boldsymbol{x} \in I, \\ (\mathcal{P}^*\mathcal{P})^{-1}(\mathcal{P}^*(\boldsymbol{U}_k^T + d^{k-1}))(\boldsymbol{x}), & \boldsymbol{x} \notin I \end{cases}$$
6:     Update $d^k$,
         $$d^k = d^{k-1} + \boldsymbol{U}_k^T - \mathcal{P}(f^{n+1}).$$
7:   end for
8:   $f^{n+1} = f^{n+1,K}$.
9: end while
```

According to the discussion in Section 3, updates for the coordinate functions α_i, which correspond to the 2D image of the i-th spectral band, are decoupled. That means the information in other spectral bands is not utilized in the update of α_i. And the only occasion when all of the information is used is when the weight matrix is updated. Therefore, we usually choose a small s_3, e.g. $s_3 \approx B/10$ for a good reconstruction result. In practice, in order to speed up the computation, we can first run a few iterations with large s_3 to get a rough inpainting result as initialization, and then use smaller s_3 to refine the result. For example, we can first choose $s_1 \times s_2 \times s_3$ to be $1 \times 1 \times B$, and then $2 \times 2 \times B$, and at last $5 \times 5 \times B/10$.

Another way to obtain a proper initialization is to utilize the low-rank matrix completion methods. In this paper, we first use the accelerated proximal gradient singular value thresholding (APG) algorithm [17] to get an initial guess of the linear mixing part of the HSI. And then we use LDMM with small patch size, $5 \times 5 \times B/10$, to pick up the more intimate nonlinear part of the image. In the numerical experiments, we found out that using low rank matrix completion as an initialization usually achieves better results than just using LDMM.

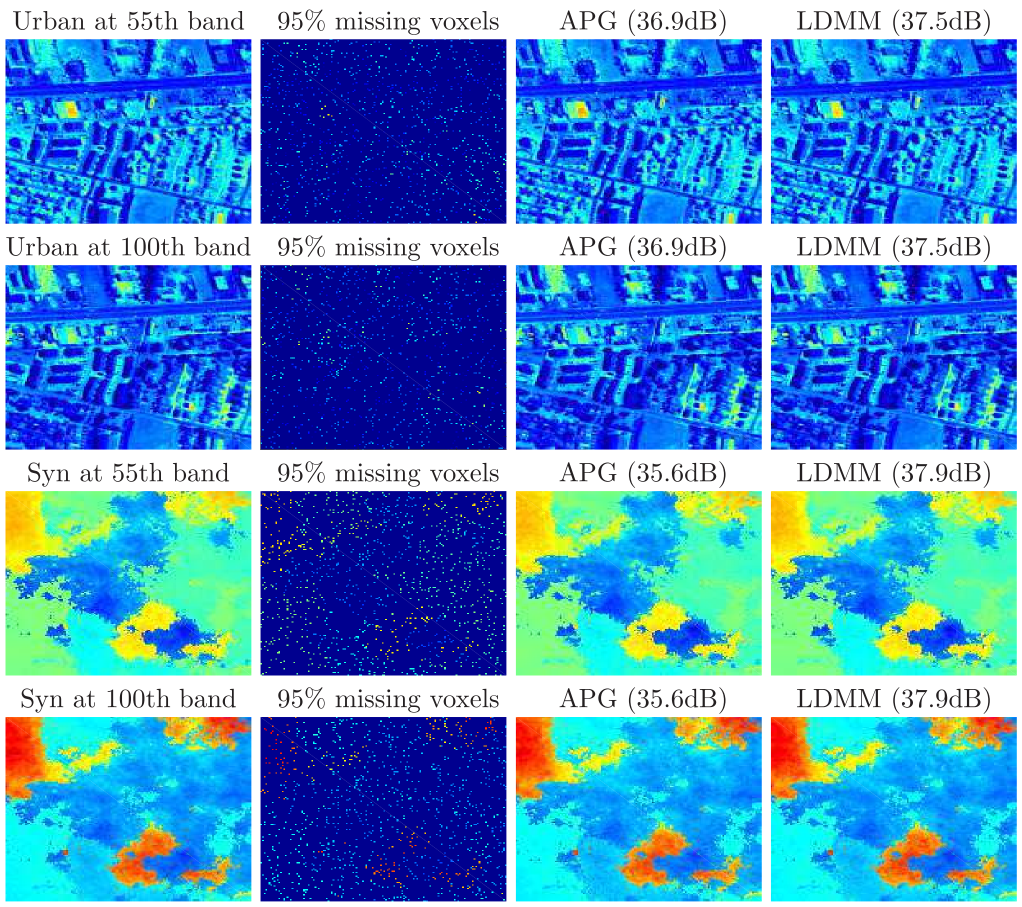

FIG. 2. *Reconstruction of the Urban and synthetic HSI with 95% missing voxels chosen uniformly at random. Figures in the third column are the results using low-rank matrix completion method APG. Figures in the fourth column are the results using LDMM with the results of APG as initialization. PSNR calculated among the entire data cube is used to quantitatively measure the accuracies of the algorithms.*

5. Numerical Results. In this section, we present the numerical results of HSI reconstruction by our proposed LDMM. All the numerical experiments are run on a Macbook Pro with 2.7 GHz Intel Core i5 CPU and 8 GB memory.

5.1. Description of the Datasets. We use two datasets, Urban and synthetic HSI, to illustrate the performance of LDMM on HSI reconstruction.

The Urban dataset is a real-world HSI from HYperspectral Digital Imagery Collection Experiment (HYDICE), which has 150×150 pixels and contains 162 spectral bands. Six classes of material (road, dirt, house, metal, tree, and grass) are mostly linearly mixed in the Urban HSI.

In the synthetic dataset, the five endmembers are randomly extracted from a real scene with a total of 162 spectral bands. The spatial dimension of the synthetic HSI is also 150×150, and the 22500 abundance vectors are generated as a sum of Gaussian fields. The dataset is generated using a bilinear mixing model:

$$y = \sum_{i=1}^{P} a_i e_i + \sum_{i=1}^{P-1} \sum_{j=i+1}^{P} \gamma_{ij} a_i a_j e_i \odot e_j + n, \tag{5.1}$$

where γ_{ij} are chosen uniformly and randomly in the interval $[0, 1]$, n is the Gaussian noise, and $e_i \odot e_j$ is the pointwise product between two endmembers.

5.2. Reconstruction of HSI with uniformly random missing pixels. The first experiment we are considering is the reconstruction of the Urban and synthetic datasets with 95% missing voxels chosen uniformly at random. In this experiment, we first use the low-rank matrix completion algorithm APG [17] to get an initial guess of the linear mixing part of the HSI, then we use LDMM with a small patch size, $5 \times 5 \times 15$, to refine the result. The reason for the choice of patch size is explained in Section 4. The parameter $\bar{\mu}$ is set to be 1, and the weight matrices are truncated to 50 nearest neighbors.

The recovering results of the Urban and synthetic data from the observation of 5% pixels using APG and LDMM are shown in Fig. 2. All spectral bands are reconstructed simultaneously, even though we only show two recovered bands for every degraded image. One can hardly tell the difference between the visual results of APG and LDMM, but PSNR defined by the following formula can be used to quantitatively measure the accuracy of the reconstruction:

$$\mathrm{PSNR}(f, f^*) = -20 \log_{10} \left(\|f - f^*\| / \max(f^*) \right), \tag{5.2}$$

where f^* is the original image. As we can see, the low-rank matrix completion method APG works pretty well on the Urban dataset (PSNR = 36.9 dB), which is essentially a linearly mixed HSI. But our LDMM using APG as initialization can further refine the result to a higher accuracy (PSNR = 37.5 dB). As for the nonlinearly mixed synthetic HSI, LDMM achieves a much better result (PSNR = 37.9 dB) than using just APG (PSNR = 35.6 dB).

Finally, we have to point out that the computational speed is a drawback of LDMM, despite its higher accuracy and adaptivity to nonlinear mixing problems. For an image of the size $150 \times 150 \times 162$, LDMM needs around 50 minutes to converge. However, our current code for LDMM is not optimized, and there are several ways to speed up the computation. For example, all the linear systems in LDMM are decoupled and can be easily parallelized. Also, to speed up the computation of the weight matrix, we can search the nearest neighbors in a local window, instead of searching amongst the entire patch set.

5.3. Reconstruction of HSI with entirely missing bands. Next, we present the result of the reconstruction of HSI with several entirely missing spectral bands. This is a more difficult problem than the previous one in the sense that the entire information of certain spectral bands is missing. We make the problem even harder by downsampling the remaining voxels significantly. In the experiments, we consider removing 16 of the 162 spectral bands chosen randomly, and we further downsample 5% of the remaining voxels. The objective is to recover the original data cube, including the removed spectral bands.

In the experiments, we first use APG on the spectral bands that are not removed, and then make inference on the missing bands by cubic spline interpolation with respect to the spectral dimension. The smoothness of the spectral signatures in HSI

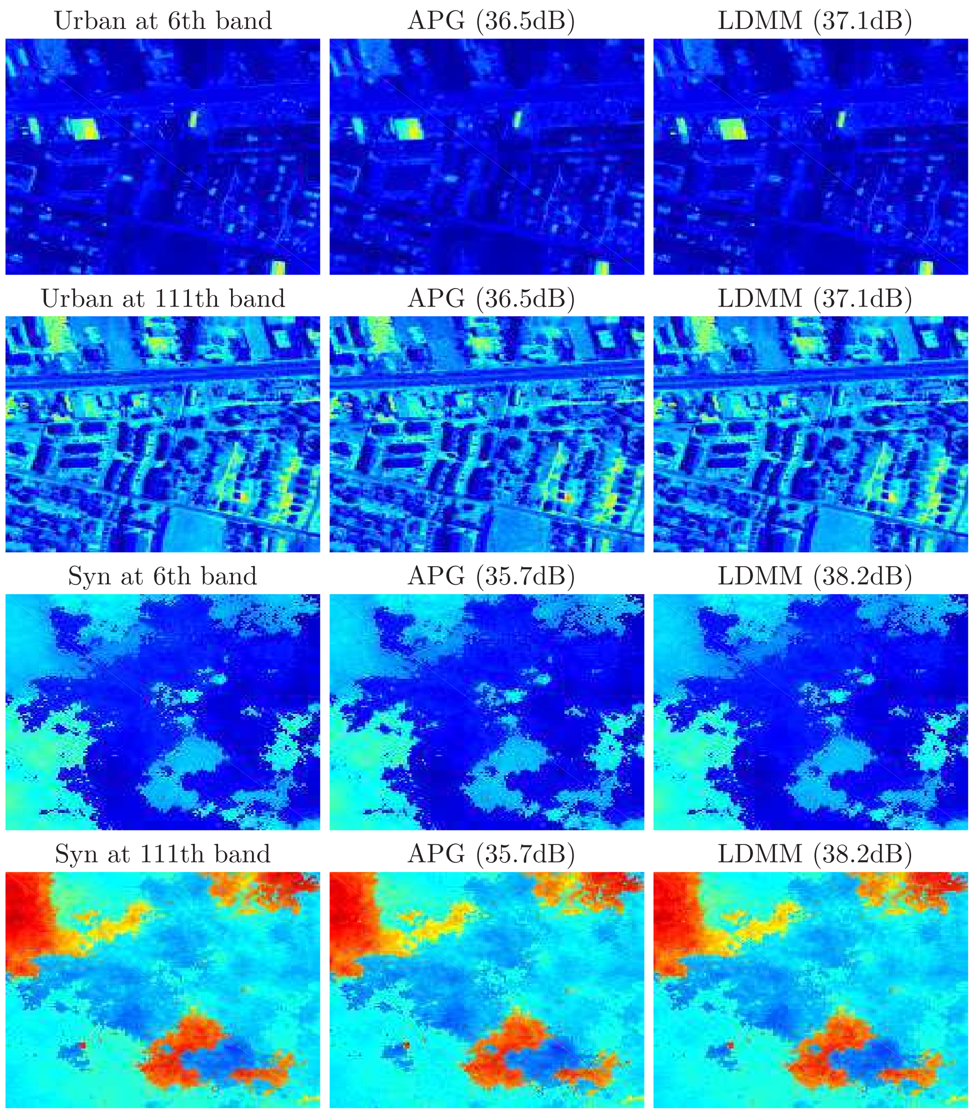

FIG. 3. *Reconstruction of the Urban and synthetic HSI with 16 entirely missing bands, and 95% missing voxels chosen uniformly at random. Figures in the third column are the results using low-rank matrix completion method APG. Figures in the fourth column are the results using LDMM with the results of APG as initialization. The spectral bands shown above are removed initially, and the PSNR is calculated among the missing bands.*

ensures the validity of the cubic spline interpolation. We then use LDMM on the entire data cube, including the missing bands, with $s_1 \times s_2 \times s_3 = 5 \times 5 \times 15$. The parameters are set to be the same as the previous experiment.

Fig. 3 shows the numerical results of the Urban and synthetic data using APG and LDMM. The spectral bands shown in the figure are removed initially, and PSNR is calculated among all the missing bands. Again, LDMM yields better results than APG on both linear and nonlinear HSI. Also, the improvement of LDMM on nonlinear HSI is more significant. We also need to point out that the higher PSNR here does not imply the results are better than the ones in Section 5.2. In this experiment, the PSNR of the synthetic dataset in the entire data cube is actually 36.3dB, which is lower than 37.9dB in the previous experiment with just missing pixels.

6. Conclusion. In this paper, we applied LDMM to the reconstruction of hyperspectral images with significant missing voxels. The low dimensionality of the patch manifold of HSI is exploited and both linear and nonlinear mixed HSI fit in this framework. In the core of the algorithm, a point integral method is used to solve a Laplace-Beltrami equation over a point cloud sampling the underlying manifold. LDMM generally produces better results on HSI inpainting problems as compared to low-rank matrix completion methods, especially on nonlinearly mixed HSI. We are now working to speed up the computational run-time of LDMM to make it a more powerful technique for general data reconstruction tasks.

Acknowledgment. The authors would like to thank Alexandre Tiard, for generating the synthetic hyperspectral image for numerical experiments.

REFERENCES

[1] J. Bioucas-Dias, A. Plaza, N. Dobigeon, M. Parente, Q. Du, P. Gader, and J. Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 5(2):354–379, 2012.

[2] A. Buades, B. Coll, and J.-M. Morel. A review of image denoising algorithms, with a new one. *Multiscale Model. Simul.*, 4:490–530, 2005.

[3] C.-I. Chang. *Hyperspectral imaging: techniques for spectral detection and classification*, volume 1. Springer Science & Business Media, 2003.

[4] A. S. Charles, B. A. Olshausen, and C. J. Rozell. Learning sparse codes for hyperspectral imagery. *IEEE Journal of Selected Topics in Signal Processing*, 5(5):963–978, 2011.

[5] N. Dobigeon, J.-Y. Tourneret, C. Richard, J. Bermudez, S. McLaughlin, and A. Hero. Nonlinear unmixing of hyperspectral images: Models and algorithms. *Signal Processing Magazine, IEEE*, 31(1):82–94, 2014.

[6] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, 2009.

[7] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM J. Imageing Sci.*, 2:323–343, 2009.

[8] H. Hu, J. Sunu, and A. L. Bertozzi. *Energy Minimization Methods in Computer Vision and Pattern Recognition: 10th International Conference, EMMCVPR 2015, Hong Kong, China, January 13-16, 2015. Proceedings*, chapter Multi-class Graph Mumford-Shah Model for Plume Detection Using the MBO scheme, pages 209–222. Springer International Publishing, Cham, 2015.

[9] R. Kawakami, Y. Matsushita, J. Wright, M. Ben-Ezra, Y. W. Tai, and K. Ikeuchi. High-resolution hyperspectral imaging via matrix factorization. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2329–2336, 2011.

[10] Z. Li, Z. Shi, and J. Sun. Point integral method for solving poisson-type equations on manifolds from point clouds with convergence guarantees. *arXiv:1409.2623.*

[11] E. Merkurjev, J. Sunu, and A. Bertozzi. Graph MBO method for multiclass segmentation of hyperspectral stand-off detection video. In *Image Processing (ICIP), 2014 IEEE International Conference on*, pages 689–693, Oct 2014.

[12] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Model. Simul.*, 4:460–489, 2005.

[13] S. Osher, Z. Shi, and W. Zhu. Low Dimensional Manifold Model for Image Processing. Technical Report UCLA CAM Report 16-04, 2016.

[14] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Phys. D*, 60:259–268, 1992.

[15] Z. Shi and J. Sun. Convergence of the point integral method for the poisson equation on manifolds i: the neumann boundary. *arXiv:1403.2141.*

[16] Z. Shi and J. Sun. Convergence of the point integral method for the poisson equation on manifolds ii: the dirichlet boundary. *arXiv:1312.4424.*

[17] K.-C. Toh and S. Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of Optimization*, 6(615-640):15, 2010.

[18] S. Valiollahzadeh and W. Yin. Hyperspectral data reconstruction combining spatial and spectral sparsity. *Rice University CAAM Technical Report TR10-29*, 2010.

[19] Z. Xing, M. Zhou, A. Castrodad, G. Sapiro, and L. Carin. Dictionary learning for noisy and incomplete hyperspectral images. *SIAM Journal on Imaging Sciences*, 5(1):33–56, 2012.

[20] D. Zhou and B. Schölkopf. *Pattern Recognition: 27th DAGM Symposium, Vienna, Austria, August 31 - September 2, 2005. Proceedings*, chapter Regularization on Discrete Spaces, pages 361–368. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[21] W. Zhu, V. Chayes, A. Tiard, S. Sanchez, D. Dahlberg, D. Kuang, A. L. Bertozzi, S. Osher, and D. Zosso. Unsupervised classification in hyperspectral imagery with nonlocal total variation and primal-dual hybrid gradient algorithm. *arXiv preprint arXiv:1604.08182*, 2016.