Chapter 10
Low Dimensional Manifold Model in Hyperspectral Image Reconstruction

Wei Zhu, Zuoqiang Shi and Stanley Osher

Abstract In this chapter, we present a low dimensional manifold model (LDMM) for hyperspectral image reconstruction. This model is based on the observation that the spatial–spectral blocks of hyperspectral images typically lie close to a collection of low dimensional manifolds. To emphasize this, we directly use the dimension of the manifold as a regularization term in a variational functional, which can be solved efficiently by alternating direction of minimization and advanced numerical discretization. Experiments on the reconstruction of hyperspectral images from sparse and noisy sampling demonstrate the superiority of LDMM in terms of both speed and accuracy.

10.1 Introduction

Hyperspectral imagery is an important domain in the field of remote sensing with numerous applications in agriculture, environmental science, and surveillance [6]. When capturing a hyperspectral image (HSI), the sensors detect the intensity of reflection at a wide range of continuous wavelengths, from the infrared to ultraviolet, to form a 3D data cube with up to thousands of spectral bands. When such data of high dimensionality are collected, the observed images are very likely degraded due to various reasons. For instance, the collected images might be extremely noisy because of limited exposure time, or some of the voxels can be missing due to the

W. Zhu (✉)
Department of Mathematics, Duke University, Durham, NC, USA
e-mail: zhu@math.duke.edu

Z. Shi
Department of Mathematical Sciences & Yau Mathematical Sciences Center,
Tsinghua University, Beijing, China
e-mail: zqshi@tsinghua.edu.cn

S. Osher
Department of Mathematics, University of California, Los Angeles, CA, USA
e-mail: sjo@math.ucla.edu

© Springer Nature Switzerland AG 2020

S. Prasad and J. Chanussot (eds.), *Hyperspectral Image Analysis*,
Advances in Computer Vision and Pattern Recognition,
https://doi.org/10.1007/978-3-030-38617-7_10

malfunctions of the hyperspectral cameras. Thus an important task in HSI analysis is to recover the original image from its noisy incomplete observation. This is an ill-posed inverse problem, and some prior knowledge of the original data must be exploited.

One widely used prior information of HSI is that the 3D data cube has a low-rank structure under the linear mixing model (LMM) [3]. More specifically, the spectral signature of each pixel is assumed to be a linear combination of a few constituent end-members. Under such an assumption, low-rank matrix completion and sparse representation techniques have been used for HSI reconstruction [7, 16, 28]. Despite the simplicity of LMM, the linear mixing assumption has been shown to be physically inaccurate in certain situations [9].

Partial differential equation (PDEs) and graph-based image processing techniques have also been applied to HSI reconstruction. The total variation (TV) method [23] has been widely used as a regularization in hyperspectral image processing [1, 13, 15, 29]. The nonlocal total variation (NLTV) [11], which computes the gradient in a nonlocal graph-based manner, has also been applied to the analysis of hyperspectral images [14, 18, 30]. However, such methods typically fail to produce satisfactory results when there is a significant number of missing voxels in the degraded HSI.

Over the past decade, patch-based manifold models have achieved great success in image processing. The key assumption in the manifold model is that image patches typically concentrate around a low dimensional smooth manifold [5, 17, 21, 22]. Based on such assumption, a low dimensional manifold model (LDMM) has been proposed for general image processing problems [20, 24], in which the dimension of the patch manifold is directly used as a regularization term in a variational functional. LDMM achieved excellent results, especially in image inpainting problems from very sparse sampling.

In this chapter, we will illustrate how LDMM can be used in HSI reconstruction. The direct extension of LDMM to higher dimensional data reconstruction has been considered in [32], but such generalization typically has poor scalability and requires huge memory storage. A considerable amount of computational burden can be reduced, however, if the special structure of hyperspectral images is utilized. Because an HSI is a collection of 2D images of the same spatial location, a single spatial similarity matrix can be shared across all spectral bands [31]. The resulting algorithm is considerably faster than its 3D counterpart: it typically takes less than two minutes given a proper initialization as compared to hours in [32].

10.2 Low Dimensional Manifold Model

We provide a detailed explanation of LDMM in HSI reconstruction, which includes the definition of the patch manifold of an HSI, the variational functional with the manifold dimension as a regularizer, and how to compute the dimension of a manifold sampled from a point cloud.

10.2.1 Patch Manifold

Let $\boldsymbol{u} \in \mathbb{R}^{m\times n\times B}$ be a hyperspectral image, where $m \times n$ and B are the spatial and spectral dimensions of the image. For any $\boldsymbol{x} \in \bar{\Omega} = [m] \times [n]$, where $[m] = \{1, 2, \ldots, m\}$, we define a patch $\mathcal{P}\boldsymbol{u}(\boldsymbol{x})$ as a 3D block of size $s_1 \times s_2 \times B$ of the original data cube $\boldsymbol{u}$, and the pixel $\boldsymbol{x}$ is the top-left corner of the rectangle of size $s_1 \times s_2$. The *patch set* $\mathcal{P}(\boldsymbol{u})$ is defined as the collection of all patches:

$$\mathcal{P}(\boldsymbol{u}) = \{\mathcal{P}\boldsymbol{u}(\boldsymbol{x}) : \boldsymbol{x} \in \bar{\Omega}\} \subset \mathbb{R}^d, \quad d = s_1 \times s_2 \times B. \tag{10.1}$$

It has been shown in [20, 32] that the point cloud $\mathcal{P}(\boldsymbol{u})$ is typically close to a collection of low dimensional smooth manifolds $\mathcal{M} = \cup_{l=1}^{L}\mathcal{M}_l$ embedded in $\mathbb{R}^d$. This collection of manifolds is called the *patch manifold* of $\boldsymbol{u}$.

Remark 10.1 We sometimes regard $\mathcal{P} : \mathbb{R}^{m\times n\times B} \to \mathbb{R}^{d\times|\bar{\Omega}|}$ as an operator that maps an HSI $\boldsymbol{u} \in \mathbb{R}^{m\times n\times B}$ to its patch set $\mathcal{P}(\boldsymbol{u}) \in \mathbb{R}^{d\times|\bar{\Omega}|}$. This point of view will be assumed throughout Sect. 10.3.

10.2.2 Model Formulation and Calculating the Manifold Dimension

Our objective is to reconstruct the unknown HSI $\boldsymbol{u}$ from its noisy and incomplete observation $\boldsymbol{b} \in \mathbb{R}^{m\times n\times B}$. Assume that for each spectral band $t \in [B]$, $\boldsymbol{b}$ is only known on a random subset $\Omega^t \subset \bar{\Omega} = [m] \times [n]$, with a sampling rate r (in our experiments $r = 5\%$ or 10%.) We also use the notation

$$\bar{\Omega}_{\text{all}} = [m] \times [n] \times [B] \quad \text{and} \quad \Omega_{\text{all}} = \cup_{t=1}^{B}\Omega^t \tag{10.2}$$

to denote the domain of the entire 3D data cube and its sampled subset. According to the analysis in Sect. 10.2.1, we can use the dimension of the patch manifold as a regularizer to reconstruct $\boldsymbol{u}$ from $\boldsymbol{b}$:

$$\begin{aligned} &\min_{\substack{\boldsymbol{u}\in\mathbb{R}^{m\times n\times B}\\ \mathcal{M}\subset\mathbb{R}^d}} \int_{\mathcal{M}} \dim(\mathcal{M}(\boldsymbol{p}))\mathrm{d}\boldsymbol{p} + \lambda \sum_{t=1}^{B} \|\boldsymbol{u}^t - \boldsymbol{b}^t\|^2_{L^2(\Omega^t)} \\ &\text{subject to:} \quad \mathcal{P}(\boldsymbol{u}) \subset \mathcal{M}, \end{aligned} \tag{10.3}$$

where $\boldsymbol{u}^t \in \mathbb{R}^{m\times n}$ is the t-th spectral band of the HSI $\boldsymbol{u} \in \mathbb{R}^{m\times n\times B}$, $\mathcal{M}(\boldsymbol{p})$ denotes the smooth manifold $\mathcal{M}_l$ to which $\boldsymbol{p}$ belongs, $\mathcal{M} = \cup_{l=1}^{L}\mathcal{M}_l$, and

$$\int_{\mathcal{M}} \dim(\mathcal{M}(\boldsymbol{p}))\mathrm{d}\boldsymbol{p} = \sum_{l=1}^{L} |\mathcal{M}_l| \dim(\mathcal{M}_l)$$

is the L^1 norm of the local dimension. Note that (10.3) is not mathematically well defined, since we do not know how to compute the manifold dimension given only a point cloud sampling the manifold. Fortunately, the following formula from differential geometry provides a simple way of calculating the dimension of a smooth manifold [20]:

Proposition 10.1 *Let $\mathcal{M}$ be a smooth submanifold isometrically embedded in $\mathbb{R}^d$. For any $\boldsymbol{p} \in \mathcal{M}$, we have*

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

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

$$\alpha_i(\boldsymbol{p}) = p_i, \quad \forall \boldsymbol{p} = (p_1, \ldots, p_d) \in \mathcal{M}.$$

Proof Since $\mathcal{M}$ is a smooth submanifold isometrically embedded in $\mathbb{R}^d$, it can be locally parametrized as

$$\boldsymbol{p} = \psi(\boldsymbol{\gamma}) : U \subset \mathbb{R}^k \to \mathcal{M} \subset \mathbb{R}^d, \tag{10.4}$$

where $k = \dim(\mathcal{M})$, $\boldsymbol{\gamma} = (\gamma^1, \ldots, \gamma^k)^T \in \mathbb{R}^k$, and $\boldsymbol{p} = (p^1, \ldots, p^d)^T \in \mathcal{M}$. With the induced metric from $\mathbb{R}^d$, we have $\partial_{i'} = (\partial_{i'}\psi^1, \ldots, \partial_{i'}\psi^d)$, and the metric tensor is

$$g_{i'j'} =< \partial_{i'}, \partial_{j'} >= \sum_{l=1}^{d} \partial_{i'}\psi^l \partial_{j'}\psi^l. \tag{10.5}$$

Let $g^{i'j'}$ denote the inverse of $g_{i'j'}$, i.e.,

$$\sum_{l'=1}^{k} g_{i'l'}\, g^{l'j'} = \delta_{i'j'} = \begin{cases} 1, & i' = j', \\ 0, & i' \neq j'. \end{cases} \tag{10.6}$$

For any function u on $\mathcal{M}$, its gradient $\nabla_{\mathcal{M}} u$ is defined as

$$\nabla_{\mathcal{M}} u = \sum_{i',j'=1}^{k} g^{i'j'} \partial_{j'} u \; \partial_{i'}. \tag{10.7}$$

When viewed as a vector in the ambient space $\mathbb{R}^d$, the j-th component of $\nabla_{\mathcal{M}} u$ in the ambient coordinates can be written as

$$\nabla^j_{\mathcal{M}} u = \sum_{i',j'=1}^{k} \partial_{i'}\psi^j g^{i'j'} \partial_{j'} u, \quad j = 1, \ldots, d. \tag{10.8}$$

Following the definition of $\nabla_{\mathcal{M}}$ in (10.8), we have

$$\begin{aligned}
\sum_{j=1}^{d} \|\nabla_{\mathcal{M}}\alpha_j\|^2 &= \sum_{i,j=1}^{d} \nabla^i_{\mathcal{M}}\alpha_j \nabla^i_{\mathcal{M}}\alpha_j \\
&= \sum_{i,j=1}^{d} \left(\sum_{i',j'=1}^{k} \partial_{i'}\psi^j g^{i'j'} \partial_{j'}\alpha_j \right) \left(\sum_{i'',j''=1}^{k} \partial_{i''}\psi^j g^{i''j''} \partial_{j''}\alpha_j \right) \\
&= \sum_{j=1}^{d} \sum_{i',j',i'',j''=1}^{k} \left(\sum_{i=1}^{d} \partial_{i'}\psi^i \partial_{i''}\psi^i \right) g^{i'j'} g^{i''j''} \partial_{j'}\alpha_j \partial_{j''}\alpha_j \\
&= \sum_{j=1}^{d} \sum_{j',i'',j''=1}^{k} \left(\sum_{i'=1}^{k} g_{i'i''} g^{i'j'} \right) g^{i''j''} \partial_{j'}\alpha_j \partial_{j''}\alpha_j \\
&= \sum_{j=1}^{d} \sum_{j',i'',j''=1}^{k} \delta_{i''j'} g^{i''j''} \partial_{j'}\alpha_j \partial_{j''}\alpha_j \\
&= \sum_{j=1}^{d} \sum_{j',j''=1}^{k} g^{j'j''} \partial_{j'}\alpha_j \partial_{j''}\alpha_j .
\end{aligned}$$

Notice that $\partial_{j'}\alpha_j = \frac{\partial}{\partial \gamma^{j'}}\alpha_j(\psi(\boldsymbol{\gamma})) = \partial_{j'}\psi^j$. We thus have

$$\begin{aligned}
\sum_{j=1}^{d} \|\nabla_{\mathcal{M}}\alpha_j\|^2 &= \sum_{j=1}^{d} \sum_{j',j''=1}^{k} g^{j'j''} \partial_{j'}\psi^j \partial_{j''}\psi^j \\
&= \sum_{j',j''=1}^{k} g^{j'j''} \left(\sum_{j=1}^{d} \partial_{j'}\psi^j \partial_{j''}\psi^j \right) \\
&= \sum_{j',j''=1}^{k} g^{j'j''} g_{j'j''} \\
&= \sum_{j'=1}^{k} \delta_{j'j'} = k = \dim(\mathcal{M}) \qquad (10.9)
\end{aligned}$$

This conclude the proof.

Based on Proposition 10.1, we can rewrite (10.3) as

$$\begin{aligned}
&\min_{\substack{\boldsymbol{u}\in\mathbb{R}^{m\times n\times B} \\ \mathcal{M}\subset\mathbb{R}^d}} \sum_{i=1}^{d_s} \sum_{t=1}^{B} \|\nabla_{\mathcal{M}}\alpha_i^t\|^2_{L^2(\mathcal{M})} + \lambda \sum_{t=1}^{B} \|\boldsymbol{u}^t - \boldsymbol{b}^t\|^2_{L^2(\Omega^t)} \\
&\text{subject to:} \quad \mathcal{P}(\boldsymbol{u}) \subset \mathcal{M}, \qquad (10.10)
\end{aligned}$$

where $d_s = s_1 \times s_2$ is the spatial dimension, α_i^t is the coordinate function that maps a point $\boldsymbol{p} = \left(p_i^t\right)_{i \in [d_s], t \in [B]} \in \mathcal{M} \subset \mathbb{R}^d$ into its (i, t)-th coordinate p_i^t. Note that (10.10) is a constrained nonconvex optimization problem with respect to $\mathcal{M}$ and $\boldsymbol{u}$, the solution of which will be explained in detail in the next section.

10.3 Two Numerical Approaches of Solving the LDMM Model

Because (10.10) is nonconvex, we attempt to solve it by alternating the direction of minimization with respect to $\boldsymbol{u}$ and $\mathcal{M}$. More specifically, given $\mathcal{M}^{(k)}$ and $\boldsymbol{u}^{(k)}$ at step k satisfying $\mathcal{P}(\boldsymbol{u}^{(k)}) \subset \mathcal{M}^{(k)}$

- With fixed $\mathcal{M}^{(k)}$, update the data $\boldsymbol{u}^{(k+1)}$ and the perturbed coordinate functions $\boldsymbol{\alpha}^{(k+1)} = \left\{[\alpha_i^t]^{(k+1)}\right\}_{i,t}$ by solving:

$$(\boldsymbol{u}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)}) = \arg\min_{\boldsymbol{u},\boldsymbol{\alpha}} \sum_{i,t} \|\nabla_{\mathcal{M}^{(k)}} \alpha_i^t\|_{L^2(\mathcal{M}^{(k)})}^2 + \lambda \sum_{t=1}^{B} \|\boldsymbol{u}^t - \boldsymbol{b}^t\|_{L^2(\Omega^t)}^2.$$
$$\text{subject to: } \boldsymbol{\alpha}(\mathcal{P}\boldsymbol{u}^{(k)}) = \mathcal{P}(\boldsymbol{u}) \qquad (10.11)$$

- Update the manifold $\mathcal{M}^{(k+1)}$ as the image of $\mathcal{M}^{(k)}$ under the perturbed coordinate function $\boldsymbol{\alpha}^{(k+1)}$

$$\mathcal{M}^{(k+1)} = \boldsymbol{\alpha}^{(k+1)}(\mathcal{M}^{(k)}). \qquad (10.12)$$

Remark 10.2 Note that $\mathcal{P}(\boldsymbol{u}^{(k+1)}) \subset \mathcal{M}^{(k+1)}$ holds because they are the images of $\mathcal{P}(\boldsymbol{u}^{(k)})$ and $\mathcal{M}^{(k)}$ under the same perturbed coordinate functions $\boldsymbol{\alpha}^{(k+1)}$. Moreover, if the above iterative procedure converges, then the adjacent iterates $\mathcal{M}^{(k)} \approx \mathcal{M}^{(k+1)}$ when k is large enough. Therefore, the perturbed coordinate function $\boldsymbol{\alpha}^{(k+1)}$ is indeed very close to identity as defined in Proposition 10.1.

Notice that (10.12) is easy to implement, whereas (10.11) is a constrained optimization problem whose numerical implementation involves the discretization of the manifold gradient operator $\nabla_{\mathcal{M}}$ over an unstructured point cloud $\mathcal{P}(\boldsymbol{u}^{(k)})$. In what follows, we provide two numerical procedures to solve the LDMM model. In the first method, problem (10.11) is further split into two subproblems, whose Euler–Lagrange equation, a Laplace–Beltrami equation over the manifold $\mathcal{M}^{(k)}$, is solved by the point integral method (PIM) [19]. In the second approach, we directly discretize the Dirichlet energy $\|\nabla_{\mathcal{M}} \alpha_i^t\|_{L^2(\mathcal{M})}^2$ in (10.11) using the weighted nonlocal

Laplacian (WNLL) [25], a practical way of finding a smooth interpolation function on a point cloud.

10.3.1 *The First Approach*

Notice that (10.11) is a convex optimization problem subject to a linear constraint, which can be further split into two simpler subproblems using the split Bregman iteration [12]. More specifically, given the l-th iterates $\boldsymbol{\alpha}^{(k+1),l}$, $\boldsymbol{u}^{(k+1),l}$, and $\boldsymbol{z}^l$:

- Update $\boldsymbol{\alpha}^{(k+1),l+1} = \left\{[\alpha_i^t]^{(k+1),l+1}\right\}_{i,t}$ with fixed $\boldsymbol{u}^{(k+1),l}$ and $\boldsymbol{z}^l$,

$$\boldsymbol{\alpha}^{(k+1),l+1} = \min_{\boldsymbol{\alpha}} \sum_{i,t} \|\nabla_{\mathcal{M}^{(k)}} \alpha_i^t\|^2_{L^2(\mathcal{M}^{(k)})} + \mu \|\boldsymbol{\alpha}(\mathcal{P}(\boldsymbol{u}^{(k)})) - \mathcal{P}(\boldsymbol{u}^{(k+1),l}) + \boldsymbol{z}^l\|_F^2, \tag{10.13}$$

  where both the patch set $\mathcal{P}(\boldsymbol{u}^{(k+1),l})$ and the image of the patch set under the perturbed coordinate functions $\boldsymbol{\alpha}(\mathcal{P}(\boldsymbol{u}^{(k)}))$ are treated as matrices in $\mathbb{R}^{d\times|\bar{\Omega}|}$.
- Update $\boldsymbol{u}^{(k+1),l+1}$ with fixed $\boldsymbol{\alpha}^{(k+1),l+1}$ and $\boldsymbol{z}^l$,

$$\begin{aligned}\boldsymbol{u}^{(k+1),l+1} &= \min_{\boldsymbol{u}} \lambda \sum_{t=1}^{B} \|\boldsymbol{u}^t - \boldsymbol{b}^t\|^2_{L^2(\Omega^t)} + \mu \|\boldsymbol{\alpha}^{(k+1),l+1}(\mathcal{P}(\boldsymbol{u}^{(k)})) - \mathcal{P}(\boldsymbol{u}) + \boldsymbol{z}^l\|_F^2 \\ &= \min_{\boldsymbol{u}} \lambda \|I_{\Omega_{\text{all}}} \boldsymbol{u} - \boldsymbol{b}\|^2_{L^2(\bar{\Omega}_{\text{all}})} + \mu \|\boldsymbol{\alpha}^{(k+1),l+1}(\mathcal{P}(\boldsymbol{u}^{(k)})) - \mathcal{P}(\boldsymbol{u}) + \boldsymbol{z}^l\|_F^2,\end{aligned} \tag{10.14}$$

  where $\Omega_{\text{all}} \subset \bar{\Omega}_{\text{all}}$ are defined in (10.2), and $I_{\Omega_{\text{all}}} : \mathbb{R}^{m\times n\times B} \to \mathbb{R}^{m\times n\times B}$ is the projection operator that sets $\boldsymbol{u}(\boldsymbol{x}, t)$ to zero for $(\boldsymbol{x}, t) \notin \Omega_{\text{all}}$, i.e.,

$$I_{\Omega_{\text{all}}} \boldsymbol{u}(\boldsymbol{x}, t) = \begin{cases} \boldsymbol{u}(\boldsymbol{x}, t) & , (\boldsymbol{x}, t) \in \Omega_{\text{all}}, \\ 0 & , (\boldsymbol{x}, t) \notin \Omega_{\text{all}}, \end{cases} \tag{10.15}$$

- Update $\boldsymbol{z}^{l+1}$,

$$\boldsymbol{z}^{l+1} = \boldsymbol{z}^l + \boldsymbol{\alpha}^{(k+1),l+1}(\mathcal{P}(\boldsymbol{u}^{(k)})) - \mathcal{P}(\boldsymbol{u}^{(k+1),l+1}). \tag{10.16}$$

Note that among (10.13), (10.14), and (10.16), the dual variable update (10.16) is trivial to implement, and the $\boldsymbol{u}$ update (10.14) has the following closed form solution:

$$\boldsymbol{u}^{(k+1),l+1} = \left(\lambda I^*_{\Omega_{\text{all}}} I_{\Omega_{\text{all}}} + \mu \mathcal{P}^* \mathcal{P}\right)^{-1} \left[\lambda I^*_{\Omega_{\text{all}}} \boldsymbol{b} + \mu \mathcal{P}^* \left(\boldsymbol{z}^l + \boldsymbol{\alpha}^{(k+1),l+1}(\mathcal{P}(\boldsymbol{u}^{(k)}))\right)\right], \tag{10.17}$$

where $I^*_{\Omega_{\text{all}}} : \mathbb{R}^{m\times n\times B} \to \mathbb{R}^{m\times n\times B}$ and $\mathcal{P}^* : \mathbb{R}^{d\times|\bar{\Omega}|} \to \mathbb{R}^{m\times n\times B}$ are the adjoint operators of $I_{\Omega_{\text{all}}}$ and $\mathcal{P}$. It is worth mentioning that $\left(\lambda I^*_{\Omega_{\text{all}}} I_{\Omega_{\text{all}}} + \mu \mathcal{P}^*\mathcal{P}\right)$ is a diagonal operator, and hence (10.17) can be solved efficiently. As for the $\boldsymbol{\alpha}$ update (10.13), one can easily check that the coordinate functions $\left\{[\alpha_i^t]^{(k+1),l+1}\right\}_{i,t}$ are decoupled, and thus (10.13) can be solved separately,

$$[\alpha_i^t]^{(k+1),l+1} = \min_{\alpha_i^t} \|\nabla_{\mathcal{M}^{(k)}} \alpha_i^t\|^2_{L^2(\mathcal{M}^{(k)})} + \mu\|\alpha_i^t(\mathcal{P}(\boldsymbol{u}^{(k)})) - \mathcal{P}_i^t(\boldsymbol{u}^{(k+1),l}) + (\boldsymbol{z}^l)_i^t\|^2, \tag{10.18}$$

where

$$\mathcal{P}_i^t(\boldsymbol{u}) = \left(\mathcal{P}_i^t \boldsymbol{u}(\boldsymbol{x})\right)_{\boldsymbol{x}\in\bar{\Omega}} \in \mathbb{R}^{|\bar{\Omega}|}, \tag{10.19}$$

and $\mathcal{P}_i^t \boldsymbol{u}(\boldsymbol{x})$ is the (i,t)-th element in the patch $\mathcal{P}\boldsymbol{u}(\boldsymbol{x})$. We next explain how to solve problem (10.18) using the point integral method.

10.3.1.1 Discretization with the Point Integral Method

Note that problem (10.18) for each individual $[\alpha_i^t]^{(k+1),l+1}$ can be cast into the following canonical form,

$$\min_{u\in H^1(\mathcal{M})} \|\nabla_{\mathcal{M}} u\|^2_{L^2(\mathcal{M})} + \mu \sum_{\boldsymbol{p}\in P} |u(\boldsymbol{p}) - v(\boldsymbol{p})|^2, \tag{10.20}$$

where u can be any α_i^t, $\mathcal{M} = \mathcal{M}^{(k)}$, $\mathcal{M} \supset P = \mathcal{P}(\boldsymbol{u}^{(k)})$, and $v(\boldsymbol{p})$ is a given function on P. It can easily be checked by standard variational methods that (10.20) is equivalent to the following Euler–Lagrange equation:

$$\begin{cases} -\Delta_{\mathcal{M}} u(\boldsymbol{p}) + \mu \displaystyle\sum_{\boldsymbol{q}\in P} \delta(\boldsymbol{p}-\boldsymbol{q})\,(u(\boldsymbol{q}) - v(\boldsymbol{q})) = 0, & \boldsymbol{p}\in\mathcal{M} \\ \hfill \dfrac{\partial u}{\partial \mathbf{n}} = 0, & \boldsymbol{p}\in\partial\mathcal{M}. \end{cases} \tag{10.21}$$

This is a Laplace–Beltrami equation over the manifold $\mathcal{M}$ sampled by the point cloud P, which can be solved efficiently using the point integral method (PIM) [19]. The key observation in PIM is the following integral approximation.

Theorem 10.1 *If $u \in C^3(\mathcal{M})$ is a function on $\mathcal{M}$, then*

$$\begin{aligned} &\left\| \int_{\mathcal{M}} \Delta_{\mathcal{M}} u(\boldsymbol{q}) R_t(\boldsymbol{p},\boldsymbol{q}) d\boldsymbol{q} - 2\int_{\partial\mathcal{M}} \frac{\partial u(\boldsymbol{q})}{\partial n} R_t(\boldsymbol{p},\boldsymbol{q}) d\tau_{\boldsymbol{q}} \right. \\ &\left. + \frac{1}{t}\int_{\mathcal{M}} (u(\boldsymbol{p}) - u(\boldsymbol{q})) R_t(\boldsymbol{p},\boldsymbol{q}) d\boldsymbol{q} \right\|_{L^2(\mathcal{M})} = O(t^{1/4}), \end{aligned} \tag{10.22}$$

where R_t is the normalized heat kernel:

$$R_t(\boldsymbol{p}, \boldsymbol{q}) = C_t \exp\left(-\frac{|\boldsymbol{p} - \boldsymbol{q}|^2}{4t}\right). \tag{10.23}$$

Theorem 10.1 suggests that the solution u of (10.21) should approximately satisfy the following integral equation:

$$\int_{\mathcal{M}} (u(\boldsymbol{p}) - u(\boldsymbol{q}))\, R_t(\boldsymbol{p}, \boldsymbol{q})\mathrm{d}\boldsymbol{q} + \mu t \sum_{\boldsymbol{q} \in P} R_t(\boldsymbol{p}, \boldsymbol{q})\, (u(\boldsymbol{q}) - v(\boldsymbol{q})) = 0. \tag{10.24}$$

If we further assume that $P = \mathcal{P}(\boldsymbol{u}^{(k)}) = \left\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_{|\bar{\Omega}|}\right\} \subset \mathbb{R}^d$ samples the manifold $\mathcal{M}$ uniformly at random, then (10.24) can be discretized into the following linear system:

$$\frac{|\mathcal{M}|}{|\bar{\Omega}|} \sum_{j=1}^{|\bar{\Omega}|} R_{t,ij}(u_i - u_j) + \mu t \sum_{j=1}^{|\bar{\Omega}|} R_{t,ij}(u_j - v_j) = 0, \tag{10.25}$$

where $u_i = u(\boldsymbol{p}_i), v_i = v(\boldsymbol{p}_i)$, and $R_{t,ij} = R_t(\boldsymbol{p}_i, \boldsymbol{p}_j)$. Equation (10.25) can be written in the matrix form

$$(\boldsymbol{L} + \bar{\mu}\boldsymbol{W})\boldsymbol{u} = \bar{\mu}\boldsymbol{W}\boldsymbol{v}, \tag{10.26}$$

where $\boldsymbol{u} = (u_1, \ldots, u_{|\bar{\Omega}|})^T, \boldsymbol{v} = (v_1, \ldots, v_{|\bar{\Omega}|})^T, \bar{\mu} = \frac{\mu t |\bar{\Omega}|}{|\mathcal{M}|}, \boldsymbol{W} = (w_{ij})_{i,j \in \{1,\ldots,|\bar{\Omega}|\}}$ is the weight matrix with $w_{ij} = R_{t,ij}$, and $\boldsymbol{L}$ is the difference between $\boldsymbol{D}$ and $\boldsymbol{W}$,

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}, \tag{10.27}$$

where $\boldsymbol{D} = diag(d_i)$ is the degree matrix with $d_i = \sum_{j=1}^{|\bar{\Omega}|} w_{ij}$. In the numerical experiments, we always truncate the weight matrix $\boldsymbol{W}$ to 20 nearest neighbors, searched efficiently using the k-d tree data structure [10]. Thus (10.26) is a sparse symmetric linear system, which can be solved by the conjugate gradient method. We summarize the numerical procedures of solving LDMM with the PIM discretization in Algorithm 1.

10.3.2 The Second Approach

We now explain our second approach of solving (10.11). This involves directly discretizing the Dirichlet energy $\|\nabla_{\mathcal{M}^{(k)}} \alpha_i^t\|^2_{L^2(\mathcal{M}^{(k)})}$ in (10.11) using the weighted nonlocal Laplacian (WNLL) [25] without splitting the update of $\boldsymbol{\alpha}$ and $\boldsymbol{u}$ [31].

Algorithm 1 LDMM for HSI reconstruction with the PIM discretization

```
Input: A noisy and incomplete observation $\boldsymbol{b}$ of an unknown hyperspectral image $\boldsymbol{u} \in \mathbb{R}^{m\times n\times B}$.
  For every spectral band $t \in [B]$, $\boldsymbol{u}$ is only partially observed on a random subset $\Omega^t$ of $\bar{\Omega} = [m] \times [n]$.
Output: Reconstructed HSI $\boldsymbol{u}$.
  Initial guess $\boldsymbol{u}^{(0)}$, $\boldsymbol{z}^0 = 0$.
  while not converge do
    1. Compute the weight matrix $\boldsymbol{W} = (w_{ij})_{1\le i,j\le|\bar{\Omega}|}$ and $\boldsymbol{L}$ from $\mathcal{P}(\boldsymbol{u}^{(k)})$,
         $w_{ij} = R_{t,ij} = R_t(\boldsymbol{p}_i, \boldsymbol{p}_j), \quad \boldsymbol{p}_i, \boldsymbol{p}_j \in \mathcal{P}(\boldsymbol{u}^{(k)}), \quad i, j = 1, \dots, |\bar{\Omega}|, \quad \boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}.$
    while not converge do
      1. Solve the linear systems for $\boldsymbol{U}^{l+1} = \boldsymbol{\alpha}^{(k+1),l+1}(\mathcal{P}(\boldsymbol{u}^{(k)})) \in \mathbb{R}^{d\times|\bar{\Omega}|}$
           $(\boldsymbol{L} + \bar{\mu}\boldsymbol{W})(\boldsymbol{U}^{l+1})^T = \bar{\mu}\boldsymbol{W}\boldsymbol{V}^l,$
         where $\boldsymbol{V}^l = \left(\mathcal{P}(\boldsymbol{u}^{(k+1),l}) - \boldsymbol{z}^l\right)^T$.
      2. Update $\boldsymbol{u}^{(k+1),l+1}$
           $\boldsymbol{u}^{(k+1),l+1} = \left(\lambda I^*_{\Omega_{\text{all}}} I_{\Omega_{\text{all}}} + \mu\mathcal{P}^*\mathcal{P}\right)^{-1} \left[\lambda I^*_{\Omega_{\text{all}}}\boldsymbol{b} + \mu\mathcal{P}^*\left(\boldsymbol{z}^l + \boldsymbol{U}^{l+1}\right)\right],$
      3. Update $\boldsymbol{z}^{l+1}$
           $\boldsymbol{z}^{l+1} = \boldsymbol{z}^l + \boldsymbol{U}^{l+1} - \mathcal{P}(\boldsymbol{u}^{(k+1),l+1}).$
      4. $l \leftarrow l + 1$.
    end while
    1. $\boldsymbol{u}^{(k+1)} \leftarrow \boldsymbol{u}^{(k+1),l}$.
    2. $k \leftarrow k + 1$.
  end while
  $\boldsymbol{u} \leftarrow \boldsymbol{u}^{(k)}$.
```

10.3.2.1 Weighted Nonlocal Laplacian

The weighted nonlocal Laplacian was proposed in [25] to find a smooth interpolation of a function on a point cloud. Suppose that $C = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \dots, \boldsymbol{c}_n\}$ is a set of points in $\mathbb{R}^d$, and let g be a function defined on a subset $S = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \dots, \boldsymbol{s}_n\} \subset C$. The objective is to extend g to C by finding a smooth function u on $\mathcal{M}$ that agrees with g when restricted to S.

A widely used method to solve the above interpolation problem is the harmonic extension model [8, 33], which seeks to minimize the following energy:

$$\mathcal{J}(u) = \|\nabla_{\mathcal{M}} u\|^2_{L^2(\mathcal{M})}, \quad \text{subject to:} \quad u(\boldsymbol{p}) = g(\boldsymbol{p}) \quad \text{on } S. \tag{10.28}$$

A common way of discretizing the manifold gradient $\nabla_{\mathcal{M}} u$ is to use its nonlocal approximation:

$$\nabla_{\mathcal{M}} u(\boldsymbol{p})(\boldsymbol{q}) \approx \sqrt{w(\boldsymbol{p},\boldsymbol{q})}\,(u(\boldsymbol{p}) - u(\boldsymbol{q}))\,,$$

where w is a positive weight function, e.g., $w(\boldsymbol{p},\boldsymbol{q}) = \exp\left(-\frac{\|\boldsymbol{p}-\boldsymbol{q}\|^2}{\sigma^2}\right)$. With this approximation, we have

$$\mathcal{J}(u) \approx \sum_{\boldsymbol{p},\boldsymbol{q}\in P} w(\boldsymbol{p},\boldsymbol{q})\,(u(\boldsymbol{p}) - u(\boldsymbol{q}))^2\,. \tag{10.29}$$

Such discretization of solving the harmonic extension model leads to the well-known graph Laplacian method [2, 4, 33]. However, a closer look into the energy $\mathcal{J}$ in (10.29) reveals that the model fails to achieve satisfactory results when the sample rate $|S|/|C|$ is low [20, 25]. More specifically, after splitting the sum in (10.29) into two terms, we have

$$\mathcal{J}(u) = \sum_{\boldsymbol{p}\in S}\sum_{\boldsymbol{q}\in C} w(\boldsymbol{p},\boldsymbol{q})\,(u(\boldsymbol{p}) - u(\boldsymbol{q}))^2 + \sum_{\boldsymbol{p}\in C\setminus S}\sum_{\boldsymbol{q}\in C} w(\boldsymbol{p},\boldsymbol{q})\,(u(\boldsymbol{p}) - u(\boldsymbol{q}))^2\,. \tag{10.30}$$

Note that the first term in (10.30) is much smaller than the second term when $|S| \ll |C|$. Therefore, the second term will be prioritized when minimizing (10.30), and the continuity of u on the sampled set S will be sacrifice. An easy remedy is to add a large weight $\mu = |C|/|S|$ in front of the first term in (10.30) to balance the two terms:

$$\mathcal{J}_{\text{WNLL}}(u) = \mu\sum_{\boldsymbol{p}\in S}\sum_{\boldsymbol{q}\in C} w(\boldsymbol{p},\boldsymbol{q})\,(u(\boldsymbol{p}) - u(\boldsymbol{q}))^2 + \sum_{\boldsymbol{p}\in C\setminus S}\sum_{\boldsymbol{q}\in C} w(\boldsymbol{p},\boldsymbol{q})\,(u(\boldsymbol{p}) - u(\boldsymbol{q}))^2\,. \tag{10.31}$$

It is readily checked that $\mathcal{J}_{\text{WNLL}}$ generalizes the graph Laplacian $\mathcal{J}$ in the sense that $\mathcal{J}_{\text{WNLL}} = \mathcal{J}$ when $|S| = |C|$. The generalized energy functional $\mathcal{J}_{\text{WNLL}}$ is called the weighted nonlocal Laplacian.

We point out that such intuition can be made precise by deriving (10.31) through the point integral method [19]. The interested reader can refer to [25] for the details.

10.3.2.2 Numerical Discretization

We now explain how to solve the optimization problem (10.11) using the weighted nonlocal Laplacian (10.31). Using the terminology introduced in Sect. 10.3.2.1, the functions to be interpolated in (10.11) are α_i^t, the point cloud C is $\mathcal{P}(\boldsymbol{u}^{(k)})$, and the sampled set for α_i^t is

$$S_i^t = \left\{\mathcal{P}\boldsymbol{u}^{(k)}(\boldsymbol{x}) : \mathcal{P}_i\boldsymbol{u}^{(k)}(\boldsymbol{x}) \text{ is sampled}\right\} \subset C.$$

Based on the discussion of WNLL in Sect. 10.3.2.1, we can discretize the Dirichlet energy $\|\nabla_{\mathcal{M}^{(k)}}\alpha_i^t\|^2_{L^2(\mathcal{M}^{(k)}}$ as

$$\begin{aligned}\|\nabla_{\mathcal{M}^{(k)}}\alpha_i^t\|^2_{L^2(\mathcal{M}^{(k)}} = &\frac{|\bar{\Omega}|}{|\Omega_i^t|}\sum_{\boldsymbol{x}\in\Omega_i^t}\sum_{\boldsymbol{y}\in\bar{\Omega}}\bar{w}(\boldsymbol{x},\boldsymbol{y})\left(\alpha_i^t(\mathcal{P}\boldsymbol{u}^{(k)}(\boldsymbol{x}))-\alpha_i^t(\mathcal{P}\boldsymbol{u}^{(k)}(\boldsymbol{y}))\right)^2\\ &+\sum_{\boldsymbol{x}\in\bar{\Omega}\backslash\Omega_i^t}\sum_{\boldsymbol{y}\in\bar{\Omega}}\bar{w}(\boldsymbol{x},\boldsymbol{y})\left(\alpha_i^t(\mathcal{P}\boldsymbol{u}^{(k)}(\boldsymbol{x}))-\alpha_i^t(\mathcal{P}\boldsymbol{u}^{(k)}(\boldsymbol{y}))\right)^2, \quad (10.32)\end{aligned}$$

where

$$\Omega_i^t = \left\{\boldsymbol{x}\in\bar{\Omega} : \mathcal{P}_i^t\boldsymbol{u}^{(k)}(\boldsymbol{x}) \text{ is sampled}\right\}$$

is a spatially translated version of Ω^t, $|\bar{\Omega}|/|\Omega_i^t| = 1/r$ is the inverse of the sampling rate, and $\bar{w}(\boldsymbol{x},\boldsymbol{y}) = w(\mathcal{P}\boldsymbol{u}^{(k)}(\boldsymbol{x}),\mathcal{P}\boldsymbol{u}^{(k)}(\boldsymbol{y}))$ is the similarity between the patches, with

$$w(\boldsymbol{p},\boldsymbol{q}) = \exp\left(-\frac{\|\boldsymbol{p}-\boldsymbol{q}\|^2}{\sigma(\boldsymbol{p})\sigma(\boldsymbol{q})}\right), \quad (10.33)$$

where $\sigma(\boldsymbol{p})$ is the normalizing factor. Combining the WNLL discretization (10.32) and the constraint in (10.11), the update of $\boldsymbol{u}$ in (10.11) can be discretized as

$$\begin{aligned}\min_{\boldsymbol{u}}\ \lambda\sum_{t=1}^{B}\|\boldsymbol{u}^t-\boldsymbol{b}^t\|^2_{L^2(\Omega^t)}+\sum_{i,t}\Bigg[&\sum_{\boldsymbol{x}\in\bar{\Omega}\backslash\Omega_i^t}\sum_{\boldsymbol{y}\in\bar{\Omega}}\bar{w}(\boldsymbol{x},\boldsymbol{y})\left(\mathcal{P}_i^t\boldsymbol{u}(\boldsymbol{x})-\mathcal{P}_i^t\boldsymbol{u}(\boldsymbol{y})\right)^2\\ &+\frac{1}{r}\sum_{\boldsymbol{x}\in\Omega_i^t}\sum_{\boldsymbol{y}\in\bar{\Omega}}\bar{w}(\boldsymbol{x},\boldsymbol{y})\left(\mathcal{P}_i^t\boldsymbol{u}(\boldsymbol{x})-\mathcal{P}_i^t\boldsymbol{u}(\boldsymbol{y})\right)^2\Bigg]. \quad (10.34)\end{aligned}$$

Remark 10.3 Unlike our first approach of solving (10.11) detailed in Sect. 10.3.1, we do not explicitly update the perturbed coordinate function $\boldsymbol{\alpha}$. The reason is that the value of $\boldsymbol{\alpha}$ on the point cloud $\mathcal{P}(\boldsymbol{u}^{(k)})$ is already implicitly determined for a given $\boldsymbol{u}$, and this is enough to discretize the Dirichlet energy $\|\nabla_{\mathcal{M}^{(k)}}\alpha_i^t\|^2_{L^2(\mathcal{M}^{(k)})}$ on the manifold $\mathcal{M}^{(k)}$.

Note that (10.34) is decoupled with respect to the spectral coordinate t, and for any given $t\in[B]$, we only need to solve the following problem:

$$\min_{\boldsymbol{u}^t} \quad \lambda\|\boldsymbol{u}^t - \boldsymbol{b}^t\|^2_{L^2(\Omega^t)} + \sum_{i=1}^{d_s} \left[\sum_{\boldsymbol{x}\in\bar{\Omega}\setminus\Omega_i^t} \sum_{\boldsymbol{y}\in\bar{\Omega}} \bar{w}(\boldsymbol{x}, \boldsymbol{y}) \left(\mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{x}) - \mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{y})\right)^2 \right.$$
$$\left. + \frac{1}{r} \sum_{\boldsymbol{x}\in\Omega_i^t} \sum_{\boldsymbol{y}\in\bar{\Omega}} \bar{w}(\boldsymbol{x}, \boldsymbol{y}) \left(\mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{x}) - \mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{y})\right)^2 \right], \tag{10.35}$$

where $\mathcal{P}_i : \mathbb{R}^{m\times n} \to \mathbb{R}^{m\times n}$ satisfies $\mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{x}) = \mathcal{P}_i^t \boldsymbol{u}(\boldsymbol{x})$. A standard variational technique shows that (10.35) is equivalent to the following Euler–Lagrange equation:

$$\left[\mu \sum_{i=1}^{d_s} \mathcal{P}_i^* I_{\Omega_i^t} \boldsymbol{h}_i^t(\boldsymbol{x}) + \sum_{i=1}^{d_s} \mathcal{P}_i^* \boldsymbol{g}_i^t(\boldsymbol{x}) + \lambda I_{\Omega^t} \left(\boldsymbol{u}^t - \boldsymbol{b}^t\right) \right](\boldsymbol{x}) = 0, \quad \forall \boldsymbol{x} \in \bar{\Omega} \tag{10.36}$$

where $\mu = 1/r - 1$, $\mathcal{P}_i^*$ is the adjoint operator of $\mathcal{P}_i$, I_{Ω^t} is the projection operator that sets $\boldsymbol{u}^t(\boldsymbol{x})$ to zero for $\boldsymbol{x} \notin \Omega^t$, i.e.,

$$I_{\Omega^t} \boldsymbol{u}^t(\boldsymbol{x}) = \begin{cases} \boldsymbol{u}^t(\boldsymbol{x}) & , \boldsymbol{x} \in \Omega^t, \\ 0 & , \boldsymbol{x} \notin \Omega^t, \end{cases} \tag{10.37}$$

and

$$\begin{cases} \boldsymbol{h}_i^t(\boldsymbol{x}) = \sum\limits_{\boldsymbol{y}\in\bar{\Omega}} \bar{w}(\boldsymbol{x}, \boldsymbol{y}) \left(\mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{x}) - \mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{y})\right) \\ \boldsymbol{g}_i^t(\boldsymbol{x}) = \sum\limits_{\boldsymbol{y}\in\bar{\Omega}} 2\bar{w}(\boldsymbol{x}, \boldsymbol{y}) \left(\mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{x}) - \mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{y})\right) + \mu \sum\limits_{\boldsymbol{y}\in\Omega_i^t} \bar{w}(\boldsymbol{x}, \boldsymbol{y}) \left(\mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{x}) - \mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{y})\right) \end{cases}$$

We use the notation $\boldsymbol{x}_{\widehat{j}}$ to denote the j-th component (in the spatial domain) after $\boldsymbol{x}$ in a patch. Assuming a periodic padding is used when patches exceed the spatial domain of HSI, one can easily verify that

$$\begin{cases} \mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{x}) = \boldsymbol{u}^t(\boldsymbol{x}_{\widehat{i-1}}), \\ \mathcal{P}_i^* \boldsymbol{u}^t(\boldsymbol{x}) = \boldsymbol{u}^t(\boldsymbol{x}_{\widehat{1-i}}). \end{cases}$$

With such notations, it follows that

$$\begin{aligned} \mathcal{P}_i^* I_{\Omega_i^t} \boldsymbol{h}_i^t(\boldsymbol{x}) =& \left[I_{\Omega_i^t} \boldsymbol{h}_i^t\right](\boldsymbol{x}_{\widehat{1-i}}) \\ =& I_{\Omega^t} \left[\boldsymbol{h}_i^t(\boldsymbol{x}_{\widehat{1-i}})\right] \\ =& I_{\Omega^t} \left[\sum_{\boldsymbol{y}\in\bar{\Omega}} \bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y}_{\widehat{1-i}}) \left(\mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{x}_{\widehat{1-i}}) - \mathcal{P}_i \boldsymbol{u}^t(\boldsymbol{y}_{\widehat{1-i}})\right) \right] \end{aligned}$$

$$=I_{\Omega^t}\left[\sum_{\boldsymbol{y}\in\bar{\Omega}} \bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y}_{\widehat{1-i}})\left(\boldsymbol{u}^t(\boldsymbol{x})-\boldsymbol{u}^t(\boldsymbol{y})\right)\right], \tag{10.38}$$

and

$$\begin{aligned}
\mathcal{P}_i^*\boldsymbol{g}_i^t(\boldsymbol{x}) =& \boldsymbol{g}_i^t(\boldsymbol{x}_{\widehat{1-i}}) \\
=& \sum_{\boldsymbol{y}\in\bar{\Omega}} 2\bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y})\left(\mathcal{P}_i\boldsymbol{u}^t(\boldsymbol{x}_{\widehat{1-i}})-\mathcal{P}_i\boldsymbol{u}^t(\boldsymbol{y})\right) \\
&+\mu\sum_{\boldsymbol{y}\in\Omega_i^t} \bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y})\left(\mathcal{P}_i\boldsymbol{u}^t(\boldsymbol{x}_{\widehat{1-i}})-\mathcal{P}_i\boldsymbol{u}^t(\boldsymbol{y})\right) \\
=& \sum_{\boldsymbol{y}\in\bar{\Omega}} 2\bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y}_{\widehat{1-i}})\left(\mathcal{P}_i\boldsymbol{u}^t(\boldsymbol{x}_{\widehat{1-i}})-\mathcal{P}_i\boldsymbol{u}^t(\boldsymbol{y}_{\widehat{1-i}})\right) \\
&+\mu\sum_{\boldsymbol{y}\in\Omega^t} \bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y}_{\widehat{1-i}})\left(\mathcal{P}_i\boldsymbol{u}^t(\boldsymbol{x}_{\widehat{1-i}})-\mathcal{P}_i\boldsymbol{u}^t(\boldsymbol{y}_{\widehat{1-i}})\right) \\
=& \sum_{\boldsymbol{y}\in\bar{\Omega}} 2\bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y}_{\widehat{1-i}})\left(\boldsymbol{u}^t(\boldsymbol{x})-\boldsymbol{u}^t(\boldsymbol{y})\right) \\
&+\mu\sum_{\boldsymbol{y}\in\Omega^t} \bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y}_{\widehat{1-i}})\left(\boldsymbol{u}^t(\boldsymbol{x})-\boldsymbol{u}^t(\boldsymbol{y})\right).
\end{aligned} \tag{10.39}$$

Therefore, we can rewrite (10.36) as

$$\begin{aligned}
&\sum_{i=1}^{d_s}\left[\sum_{\boldsymbol{y}\in\bar{\Omega}} 2\bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y}_{\widehat{1-i}})\left(\boldsymbol{u}^t(\boldsymbol{x})-\boldsymbol{u}^t(\boldsymbol{y})\right)+\mu\sum_{\boldsymbol{y}\in\Omega^t} \bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y}_{\widehat{1-i}})\left(\boldsymbol{u}^t(\boldsymbol{x})-\boldsymbol{u}^t(\boldsymbol{y})\right)\right] \\
&\quad+\mu I_{\Omega^t}\left[\sum_{\boldsymbol{y}\in\bar{\Omega}}\sum_{i=1}^{d_s} \bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y}_{\widehat{1-i}})\left(\boldsymbol{u}^t(\boldsymbol{x})-\boldsymbol{u}^t(\boldsymbol{y})\right)\right]+\lambda I_{\Omega^t}\left(\boldsymbol{u}^t-\boldsymbol{b}^t\right)=0,\ \forall\boldsymbol{x}\in\bar{\Omega}.
\end{aligned} \tag{10.40}$$

After assembling the weight matrices $\bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y}_{\widehat{1-i}})$ into

$$\tilde{w}(\boldsymbol{x}, \boldsymbol{y})=\sum_{i=1}^{d_s}\bar{w}(\boldsymbol{x}_{\widehat{1-i}}, \boldsymbol{y}_{\widehat{1-i}}), \tag{10.41}$$

it follows that (10.40) is equivalent to

$$2\sum_{y\in\bar{\Omega}}\tilde{w}(\boldsymbol{x},\boldsymbol{y})\left(\boldsymbol{u}^t(\boldsymbol{x})-\boldsymbol{u}^t(\boldsymbol{y})\right)+\mu\sum_{y\in\Omega^t}\tilde{w}(\boldsymbol{x},\boldsymbol{y})\left(\boldsymbol{u}^t(\boldsymbol{x})-\boldsymbol{u}^t(\boldsymbol{y})\right)$$
$$+\mu I_{\Omega^t}\left[\sum_{y\in\bar{\Omega}}\tilde{w}(\boldsymbol{x},\boldsymbol{y})\left(\boldsymbol{u}^t(\boldsymbol{x})-\boldsymbol{u}^t(\boldsymbol{y})\right)\right]+\lambda I_{\Omega^t}\left(\boldsymbol{u}^t-\boldsymbol{b}^t\right)=0,\quad\forall\boldsymbol{x}\in\bar{\Omega} \tag{10.42}$$

Note that (10.42) is a sparse linear system for $\boldsymbol{u}^t$ in $\mathbb{R}^{mn}$, but unlike (10.26), the coefficient matrix is not symmetric because of the projection operator I_{Ω^t}. We thus use the generalized minimal residual method (GMRES) to solve the systems (10.42). The numerical procedures of solving LDMM with the WNLL discretization is summarized in Algorithm 2.

Algorithm 2 LDMM for HSI reconstruction with the WNLL discretization

```
Input: A noisy and incomplete observation $\boldsymbol{b}$ of an unknown hyperspectral image $\boldsymbol{u}\in\mathbb{R}^{m\times n\times B}$.
  For every spectral band $t\in[B]$, $\boldsymbol{u}$ is only partially observed on a random subset $\Omega^t$ of $\bar{\Omega}=[m]\times[n]$.
Output: Reconstructed HSI $\boldsymbol{u}$.
  Initial guess $\boldsymbol{u}^{(0)}$.
  while not converge do
    1. Extract the patch set $\mathcal{P}\boldsymbol{u}^{(k)}$ from $\boldsymbol{u}^{(k)}$.
    2. Compute the similarity matrix on the spatial domain
         $\overline{w}(\boldsymbol{x},\boldsymbol{y})=w(\mathcal{P}\boldsymbol{u}^{(k)}(\boldsymbol{x}),\mathcal{P}\boldsymbol{u}^{(k)}(\boldsymbol{y})),\quad\boldsymbol{x},\boldsymbol{y}\in\overline{\Omega}.$
    3. Assemble the new similarity matrix
         $\tilde{w}(\boldsymbol{x},\boldsymbol{y})=\sum_{i=1}^{d_s}\bar{w}(\boldsymbol{x}_{\widehat{1-i}},\boldsymbol{y}_{\widehat{1-i}})$
    4. For every spectral band $t$, Update $(\boldsymbol{u}^t)^{(k+1)}$ as the solution of (10.42) using GMRES.
    5. $k\leftarrow k+1$.
  end while
  $\boldsymbol{u}=\boldsymbol{u}^{(k)}$.
```

10.3.3 A Comparison of the Two Approaches

We first compare the computational cost of the two approaches. The most time-consuming part of both algorithms is solving the $|\bar{\Omega}|$-dimensional sparse linear systems. For each iteration in the inner loop of Algorithm 1, one needs to solve $d=s_1\times s_2\times B$ linear systems. On the other hand, one only needs to solve B linear systems in each iteration of Algorithm 2. Moreover, unlike Algorithm 1, Algorithm 2

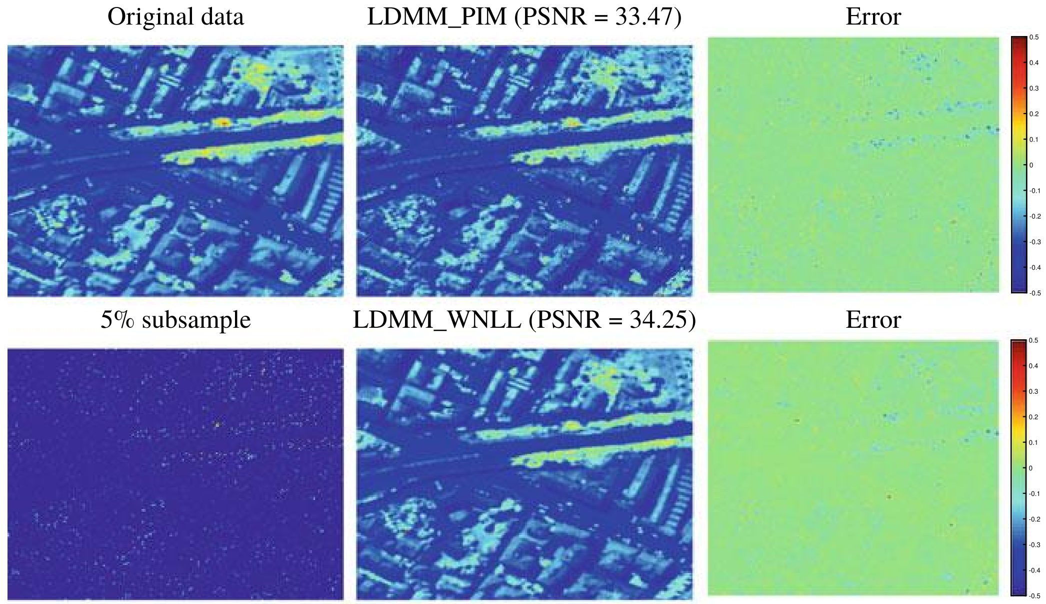

Fig. 10.1 Reconstruction of the Pavia Center dataset from its 5% noise-free subsample. The first column displays the original image and its 5% random subsample at one spectral band. The remaining two columns display the reconstructed images and the error (the difference between the original data and the reconstruction) using LDMM with the PIM (first row) and the WNLL (second row) discretization

does not have an inner loop. The reason is that the weight assembly step (10.41) in Algorithm 2 combines $s_1 \times s_2$ equations in the spatial patch domain into only one equation, and the WNLL discretization enforces the constraint in (10.11) directly without a further splitting. Therefore, Algorithm 2 is much more computationally efficient as compared to Algorithm 1.

We also compare the numerical accuracy of the two algorithms in the reconstruction of hyperspectral images from their 5% noise-free random subsamples. In the experiments, we set the spatial patch size $s_1 \times s_2$ to 2×2, and Figs. 10.1 and 10.2 present the performance of the two algorithms on the Pavia Center and Pavia University datasets. The peak signal-to-noise ratio,

$$\mathrm{PSNR} = 10 \log_{10} \left(\frac{\|\boldsymbol{u}^*\|_\infty}{\mathrm{MSE}} \right), \tag{10.43}$$

is used to evaluate the reconstruction, where $\boldsymbol{u}^*$ is the ground truth, and MSE is the mean squared error. As can be seen from Figs. 10.1 and 10.2, even though both algorithms lead to remarkable results of HSI reconstruction, LDMM with the WNLL discretization has a slight edge over PIM in terms of accuracy as well. Due to the advantage of WNLL in both computational efficiency and reconstruction accuracy, we will report the results of LDMM with WNLL discretization only in the following experiments, even though PIM has more theoretical guarantee on the consistency of the discretization (c.f. Theorem 10.1).

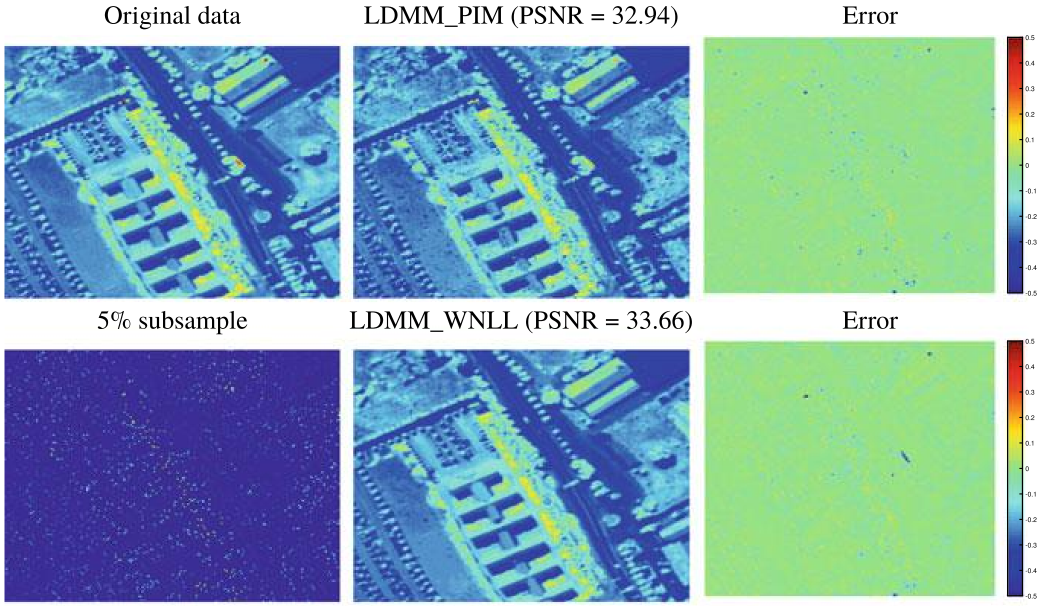

Fig. 10.2 Reconstruction of the Pavia University dataset from its 5% noise-free subsample. The first column displays the original image and its 5% random subsample at one spectral band. The remaining two columns display the reconstructed images and the error (the difference between the original data and the reconstruction) using LDMM with the PIM (first row) and the WNLL (second row) discretization

10.4 Numerical Experiments

In this section, we present the numerical results on the following datasets: Pavia University, Pavia Center, Indian Pine, and San Diego Airport. All images have been cropped in the spatial dimension to 200 × 200 for easy comparison. The objective of the experiment is to reconstruct the original HSI from 5% random subsample (10% random subsample for noisy data). As discussed in Sect. 10.3.3, we choose LDMM with the WNLL discretization (Algorithm 2) as the default method because of its computational efficiency and superior numerical accuracy.

10.4.1 Experimental Setup

Empirically, we discovered that it is easier for LDMM to converge if we use a reasonable initialization. In our experiments, we always use the result of the low-rank matrix completion algorithm APG [26] as an initialization, and run three iterations of manifold update for LDMM. The peak signal-to-noise ratio (PSNR) defined in (10.43) is used to evaluate the reconstruction accuracy. All experiments were run on a Linux machine with 8 Intel core i7-7820X 3.6 GHz CPUs and 64 GB of RAM. Codes and datasets are available for download at http://services.math.duke.edu/~zhu/software/HSI_LDMM_public.tar.gz.

10.4.2 Reconstruction from Noise-Free Subsample

We first present the results of the reconstruction of hyperspectral images from their 5% noise-free random subsamples. Table 10.1 displays the computational time and accuracy of the low-rank matrix completion initialization (APG) and LDMM with different spatial patch sizes. Unlike high-resolution RGB images, the spatial resolution of the hyperspectral images considered in this chapter is typically much lower, and thus we limited the choice of spatial patch size to only 1 × 1 and 2 × 2. It is clear from the table that LDMM significantly improves the accuracy of the APG initialization with comparable extra computational time. Figures 10.3 and 10.4 provide a visual illustration of the results. It can be observed from the figures that the reconstructions of the subsampled hyperspectral images by LDMM are spatially much smoother than the APG initialization because of the low-dimensionality regularization on the patch manifold. What is also quite interesting from the error map of LDMM in Fig. 10.3 is the relatively poorer reconstruction of the several "rare" objects in the scene, e.g., the two airplanes on the upper right corner of the image. The poor reconstruction of these "anomalies" in the scene is due to the fact that the patch manifolds of these rare objects are not well-resolved with only limited samples. This observation does mean that LDMM is less robust when reconstructing images with many sparsely sampled distinct objects. However, on the other hand, it also suggests that LDMM might be used as an "anomaly detection" algorithm by purposefully subsampling the original image and identifying the anomalies as the objects that are least well reconstructed by LDMM. This usage of LDMM as an hyperspectral anomaly detection algorithm has been studied recently in [27].

Table 10.1 Reconstruction of the HSIs from their noise-free 5% subsamples. LDMM (1 × 1) and LDMM (2 × 2) stand for LDMM with spatial patch size of 1 × 1 and 2 × 2. The reported time of LDMM does not include that of the AGP initialization

	APG		LDMM (1 × 1)		LDMM (2 × 2)	
	PSNR	Time (s)	PSNR	Time (s)	PSNR	Time (s)
Indian Pine	26.80	13	32.09	8	**34.08**	22
Pavia Center	32.61	17	**34.54**	11	34.25	31
Pavia University	31.51	13	33.38	11	**33.66**	29
San Diego Airport	32.43	23	40.33	16	**44.21**	46

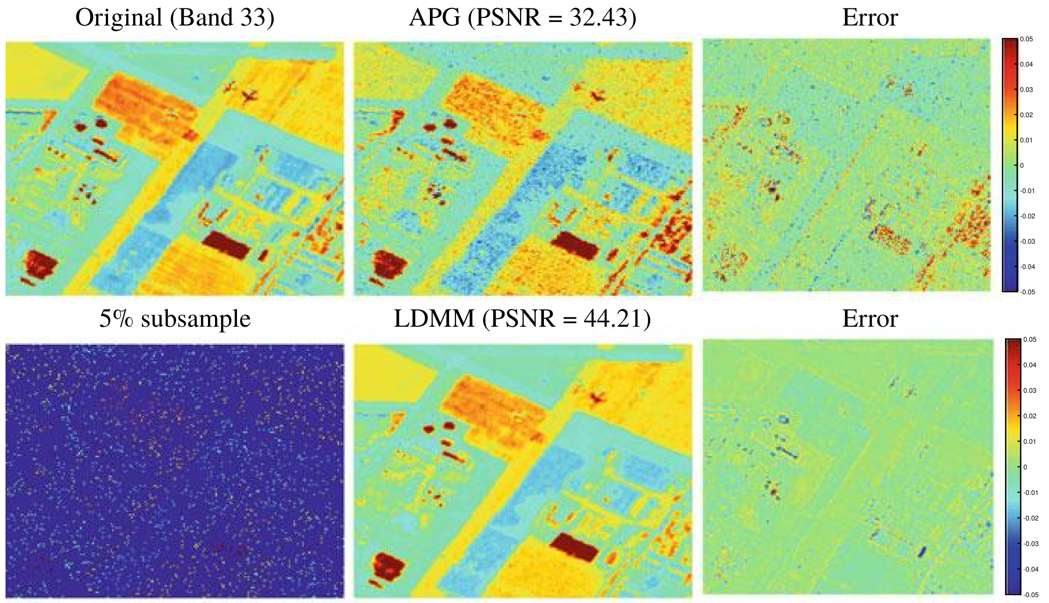

Fig. 10.3 Reconstruction of the San Diego Airport dataset from its 5% noise-free subsample. Note that the error is displayed with a scale 1/20 of the original data to visually amplify the difference

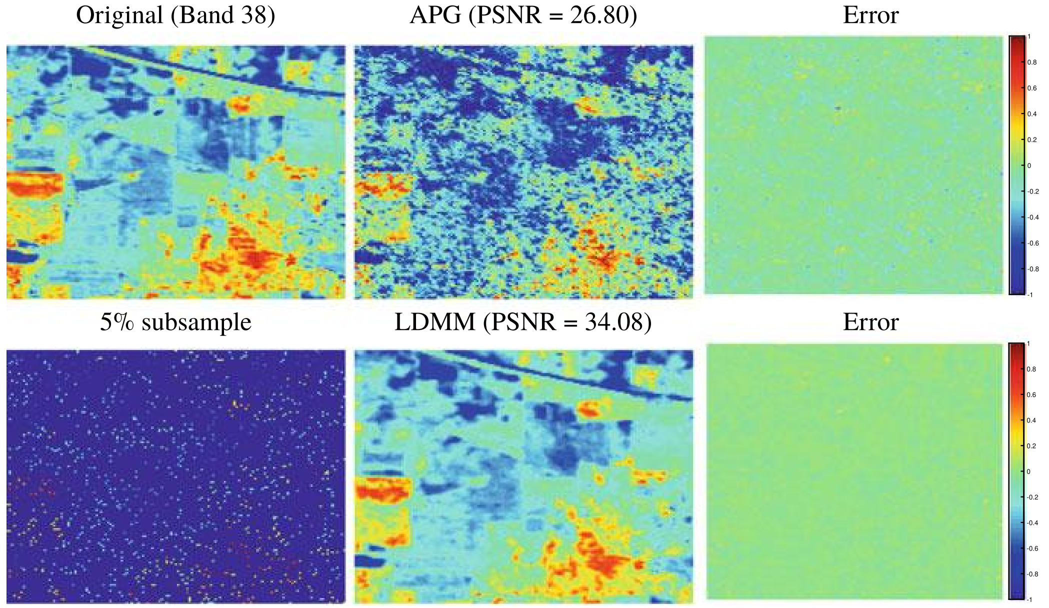

Fig. 10.4 Reconstruction of the Indian Pine dataset from its 5% noise-free subsample

10.4.3 Reconstruction from Noisy Subsample

We next show the results of the reconstruction of hyperspectral images from their 10% noisy subsample. A Gaussian noise with a standard deviation of 0.05 is first added to the original image, and 90% of the voxels are later removed from the data cube. We report the accuracy and computational time of the experiments in Table 10.2. Note that when noise is present, LDMM with 2 × 2 patches typically produce better results than that with 1 × 1 patches. This is due to the stronger spatial regularization by choosing a larger patch size. Visual illustrations of the reconstruction are displayed in Figs. 10.5 and 10.6. It can be observed from the figures that, even with the

Table 10.2 Reconstruction of the noisy HSIs from their 10% subsamples. LDMM (1 × 1) and LDMM (2 × 2) stand for LDMM with spatial patch size of 1 × 1 and 2 × 2. The reported time of LDMM does not include that of the AGP initialization

	APG		LDMM (1 × 1)		LDMM (2 × 2)	
	PSNR	Time (s)	PSNR	Time (s)	PSNR	Time (s)
Indian Pine	31.56	18	**34.03**	54	34.02	56
Pavia Center	30.22	47	30.55	82	**31.61**	82
Pavia University	29.88	38	30.26	77	**31.40**	86
San Diego Airport	33.90	69	39.17	186	**41.31**	231

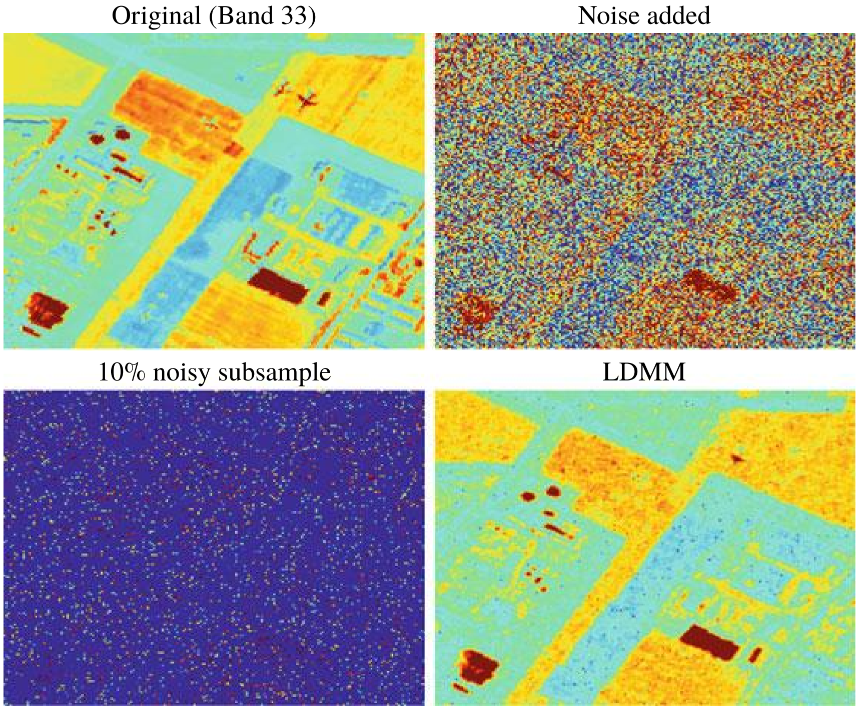

Fig. 10.5 Reconstruction of the San Diego Airport dataset from its 10% noisy subsample

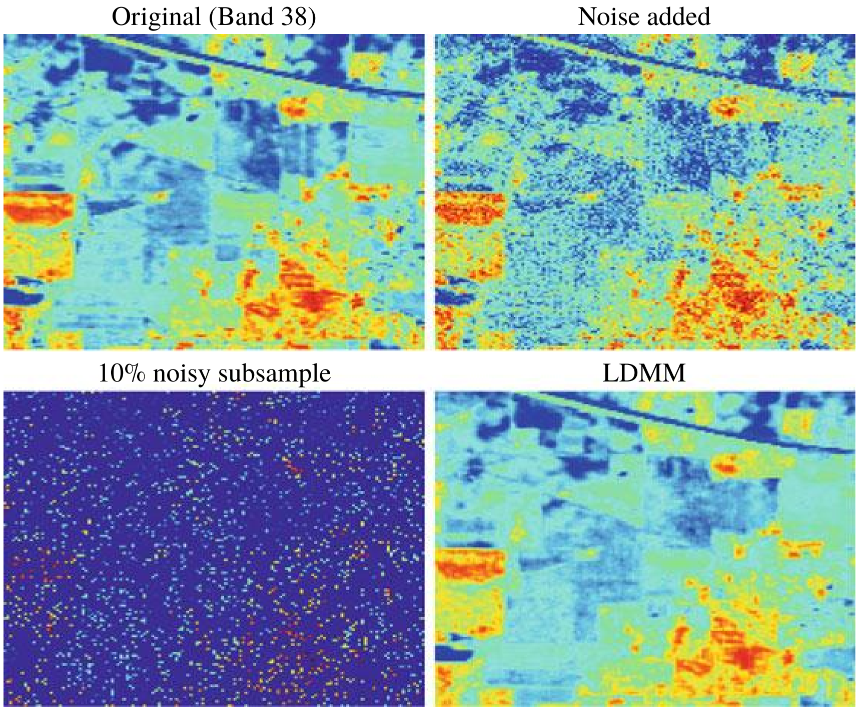

Fig. 10.6 Reconstruction of the Indian Pine dataset from its 10% noisy subsample

presence of significant noise in the remaining subsampled voxels, LDMM is able to achieve a reasonable reconstruction of the original HSI, especially for objects that are abundantly represented in the image.

10.5 Conclusion

We explained in this chapter the low dimensional manifold model for the reconstruction of hyperspectral images from noisy and incomplete observations with a significant number of missing voxels. LDMM is based on the assumption that the 3D patches in a hyperspectral image tend to sample a collection of low dimensional manifolds. As a result, we directly use the dimension of the patch manifold as a regularizer in a variational functional, which can be solved using either the point integral method or the weighted nonlocal Laplacian. Because of the special data structure of hyperspectral images, the same similarity matrix can be shared across all spectral bands, which significantly reduces the computational burden. Numerical experiments show that the proposed algorithm is both accurate and efficient for HSI reconstruction from its noisy and incomplete observation.

Acknowledgements This work was supported by STROBE: A National Science Foundation Science & Technology Center, under Grant No. DMR 1548924 as well as DOE-DE-SC0013838 and NSFC 11671005.

References

1. Aggarwal HK, Majumdar A (2016) Hyperspectral image denoising using spatio-spectral total variation. IEEE Geosci Remote Sens Lett 13(3):442–446
2. Bertozzi AL, Flenner A (2012) Diffuse interface models on graphs for classification of high dimensional data. Multiscale Model Simul 10(3):1090–1118
3. Bioucas-Dias J, Plaza A, Dobigeon N, Parente M, Du Q, Gader P, Chanussot J (2012) Hyperspectral unmixing overview: geometrical, statistical, and sparse regression-based approaches. IEEE J Sel Top Appl Earth Obs Remote Sens 5(2):354–379
4. Bühler T, Hein M (2009) Spectral clustering based on the graph p-Laplacian. In: Proceedings of the 26th annual international conference on machine learning. ACM, pp 81–88
5. Carlsson G, Ishkhanov T, de Silva V, Zomorodian A (2008) On the local behavior of spaces of natural images. Int J Comput Vis 76(1):1–12
6. Chang C-I (2003) Hyperspectral imaging: techniques for spectral detection and classification, vol 1. Springer Science & Business Media
7. Charles AS, Olshausen BA, Rozell CJ (2011) Learning sparse codes for hyperspectral imagery. IEEE J Sel Top Signal Process 5(5):963–978
8. Chung FR, Graham FC (1997) Spectral graph theory, vol 92. American Mathematical Society
9. Dobigeon N, Tourneret J-Y, Richard C, Bermudez J, McLaughlin S, Hero A (2014) Nonlinear unmixing of hyperspectral images: models and algorithms. IEEE Signal Process Mag 31(1):82–94
10. Friedman JH, Bentley JL, Finkel RA (1976) An algorithm for finding best matches in logarithmic time. ACM Trans Math Softw, 3(SLAC-PUB-1549-REV. 2):209–226
11. Gilboa G, Osher S (2009) Nonlocal operators with applications to image processing. Multiscale Model Simul 7(3):1005–1028
12. Goldstein T, Osher S (2009) The split bregman method for l1-regularized problems. SIAM J Imaging Sci 2(2):323–343
13. He W, Zhang H, Zhang L, Shen H (2016) Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration. IEEE Trans Geosci Remote Sens 54(1):178–188
14. Hu H, Sunu J, Bertozzi AL (2015) Multi-class graph Mumford-Shah model for plume detection using the MBO scheme. In: Energy minimization methods in computer vision and pattern recognition: 10th international conference, EMMCVPR 2015, Hong Kong, China, January 13–16, 2015. Proceedings. Springer International Publishing, Cham, pp 209–222
15. Iordache MD, Bioucas-Dias JM, Plaza A (2012) Total variation spatial regularization for sparse hyperspectral unmixing. IEEE Trans Geosci Remote Sens 50(11):4484–4502
16. Kawakami R, Matsushita Y, Wright J, Ben-Ezra M, Tai YW, Ikeuchi K (2011) High-resolution hyperspectral imaging via matrix factorization. In: 2011 IEEE conference on computer vision and pattern recognition (CVPR), pp 2329–2336
17. Lee AB, Pedersen KS, Mumford D (2003) The nonlinear statistics of high-contrast patches in natural images. Int J Comput Vis 54(1–3):83–103
18. Li J, Yuan Q, Shen H, Zhang L (2015) Hyperspectral image recovery employing a multidimensional nonlocal total variation model. Signal Process 111:230–248
19. Li Z, Shi Z, Sun J (2017) Point integral method for solving poisson-type equations on manifolds from point clouds with convergence guarantees. Commun Comput Phys 22(1):228–258
20. Osher S, Shi Z, Zhu W (2017) Low dimensional manifold model for image processing. SIAM J Imaging Sci 10(4):1669–1690
21. Peyré G (2008) Image processing with nonlocal spectral bases. Multiscale Model Simul 7(2):703–730
22. Peyré G (2011) A review of adaptive image representations. IEEE J Sel Top Signal Process 5(5):896–911
23. Rudin LI, Osher S, Fatemi E (1992) Nonlinear total variation based noise removal algorithms. Phys D 60:259–268
24. Shi Z, Osher S, Zhu W (2017) Generalization of the weighted nonlocal Laplacian in low dimensional manifold model. J Sci Comput

25. Shi Z, Osher S, Zhu W (2017) Weighted nonlocal Laplacian on interpolation from sparse data. J Sci Comput 73(2):1164–1177
26. Toh K-C, Yun S (2010) An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. Pac J Optim 6(615–640):15
27. Wu Z, Zhu W, Chanussot J, Xu Y, Osher S (2019) Hyperspectral anomaly detection via global and local joint modeling of background. IEEE Trans Signal Process 67(14):3858–3869
28. Xing Z, Zhou M, Castrodad A, Sapiro G, Carin L (2012) Dictionary learning for noisy and incomplete hyperspectral images. SIAM J Imaging Sci 5(1):33–56
29. Yuan Q, Zhang L, Shen H (2012) Hyperspectral image denoising employing a spectral-spatial adaptive total variation model. IEEE Trans Geosci Remote Sens 50(10):3660–3677
30. Zhu W, Chayes V, Tiard A, Sanchez S, Dahlberg D, Bertozzi AL, Osher S, Zosso D, Kuang D (2017) Unsupervised classification in hyperspectral imagery with nonlocal total variation and primal-dual hybrid gradient algorithm. IEEE Trans Geosci Remote Sens 55(5):2786–2798
31. Zhu W, Shi Z, Osher S (2018) Scalable low dimensional manifold model in the reconstruction of noisy and incomplete hyperspectral images. In: 9th workshop on hyperspectral image and signal processing: evolution in remote sensing (WHISPERS)
32. Zhu W, Wang B, Barnard R, Hauck CD, Jenko F, Osher S (2018) Scientific data interpolation with low dimensional manifold model. J Comput Phys 352:213–245
33. Zhu X, Ghahramani Z, Lafferty JD (2003) Semi-supervised learning using Gaussian fields and harmonic functions. In: Proceedings of the 20th international conference on machine learning (ICML-03), pp 912–919