LOW DIMENSIONAL MANIFOLD MODEL WITH SEMI-LOCAL PATCHES *

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

Abstract. In this paper we integrate semi-local patches and the weighted graph Laplacian [17] into the framework of the low dimensional manifold model [12]. This approach is much faster than the original LDMM algorithm. The number of iterations is typically reduced from 100 to 10 and the equations in each step are much easier to solve. This new approach is tested in image inpainting and denoising and the results are better than the original LDMM and competitive with state-of-the-art methods.

1. Introduction. In the past decade patch-based methods have achieved great success in image denoising, inpainting and other image processing problems [6, 2, 1, 4, 8, 22, 7, 24]. Among different patch-based nonlocal methods, the manifold model is attracting more and more attention, along with the development of manifold learning algorithms. The basic assumption in the manifold model is that the patches concentrate around a low dimensional smooth manifold. This assumption is verified in studies in image processing and computer vision [14, 15, 16, 9, 3, 13].

Recently the low dimensional manifold model (LDMM) was proposed for image processing [12], explicitly using the dimension of the patch manifold as a regularization to reconstruct the image. Using an elegant formula in differential geometry, LDMM is formulated as an optimization problem of minimizing the H^1 norm of the coordinate functions of the patch manifold. The key step in the algorithm is to solve a Laplace-Beltrami equation on the patch set. This equation is solved by the point integral method [10, 11, 18, 19]. LDMM has been shown to have very good performance in image processing, especially in inpainting problems. On the other hand, the computational cost of the original LDMM in [12] is relatively high. There are mainly two flaws which increase the computational load. First, the linear systems to be solved from the point integral method are usually not symmetric and not very efficient to solve numerically. Secondly, in LDMM, the image is recovered iteratively. To get a good reconstruction, the number of iterations is usually over 100. In this paper we use semi-local patches to replace nonlocal patches and use the weighted graph Laplacian to replace the point integral method. Based on these two improvements, the computational cost is reduced significantly and the results in image inpainting and denoising become even better.

The semi-local patches are combinations of nonlocal patches and local coordinates,

$$(\bar{\mathcal{P}}\mathbf{u})(x) = [(\mathcal{P}\mathbf{u})(x), \lambda x],$$

where $\mathbf{u} \in \mathbb{R}^{m\times n}$ is an image, $x \in \overline{\Omega} = \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}$ is the index of the pixel, $\mathcal{P}\mathbf{u}(x)$ is the patch at x which is defined precisely in Section 2. The weight λ is used to get different locality in the semi-local patch. The semi-local patches have

*Research supported by DOE-SC0013838 and NSF DMS-1118971. Z. Shi was partially supported by NSFC Grant 11371220, 11671005.

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

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

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

been applied to many image processing problems [14, 23, 20, 21, 5]. They give a way to balance the local and nonlocal effect. We find that with the proper choice of weight λ, semi-local patches could significantly decrease the number of iterations in LDMM and give better results.

In the image inpainting problem, we use the weighted graph Laplacian instead of the point integral method to discretize the optimization problem. The weighted graph Laplacian (WGL) is a method to compute smooth interpolation on a point cloud, which was recently developed in [17]. The key idea in the weighted graph Laplacian is to separate the energy function into two parts: one in the labeled set and the other in the unlabeled set. In WGL, we put a weight in front of the labeled set to correctly enforce the constraints on the labeled set.

(1.1)
$$\min_u \sum_{\boldsymbol{x}\in P\setminus S}\left(\sum_{\boldsymbol{y}\in P} w(\boldsymbol{x},\boldsymbol{y})(u(\boldsymbol{x})-u(\boldsymbol{y}))^2\right)+\frac{|P|}{|S|}\sum_{\boldsymbol{x}\in S}\left(\sum_{\boldsymbol{y}\in P} w(\boldsymbol{x},\boldsymbol{y})(u(\boldsymbol{x})-u(\boldsymbol{y}))^2\right),$$

with the constraint

$$u(\boldsymbol{x})=g(\boldsymbol{x}),\quad \boldsymbol{x}\in S,$$

where P is the point cloud and S is the labeled set, $w(\boldsymbol{x},\boldsymbol{y})$ is a given weight function and g is the given values on S. The equation in WGL is symmetric and positive definite, which is much easier to solve than that in the point integral method.

Using the semi-local patches and WGL, we can reduce the number of iterations in LDMM from 100 to 10. In each step, the linear system is easier to solve. Thus, the total computational time is reduced tremendously. Also the result is better than those in [12] in both image inpainting and denoising.

The rest of the paper is organized as follows. For the completeness of this paper, we briefly introduce LDMM and WGL in Sections 2 and 3 respectively. The semi-local patches are discussed in Section 4. We apply the new LDMM method to image inpainting and denoising in Section 5 and Section 6. Finally, some conclusions are made in Section 7.

2. Low Dimensional Manifold Model. Recently, the low dimensional manifold model (LDMM) was proposed for image processing, which gives very good results especially for subsampled image reconstruction [12]. The main idea in LDMM is using the dimension of the patch manifold associated with the image as a regularization to recover the image such that the associated patch manifold has the lowest dimension.

Considering a discrete image $\mathbf{u}\in\mathbb{R}^{m\times n}$, for any $x\in\overline{\Omega}=\{1,2,\dots,m\}\times\{1,2,\dots,n\}$, we define a patch $(\mathcal{P}\mathbf{u})(x)$ as a 2D rectangle of size $s_1\times s_2$ of the original image $\mathbf{u}$, and the pixel x is the top-left corner of that rectangle. The collection of all patches is called the patch set of image $\mathbf{u}$ and is denoted as $P(\mathbf{u})$:

$$P(\mathbf{u})=\{(\mathcal{P}\mathbf{u})(x): x\in\{1,2,\dots,m\}\times\{1,2,\dots,n\}\}\subset\mathbb{R}^d,\quad d=s_1\times s_2. \tag{2.1}$$

For a given image $\mathbf{u}$, the patch set $P(\mathbf{u})$ gives a point cloud in $\mathbb{R}^d$. It is observed that this point cloud is usually close to a smooth manifold embedded in $\mathbb{R}^d$. This underlying smooth manifold is called the *patch manifold* associated with $\mathbf{u}$, denoted as $\mathcal{M}(\mathbf{u})$. Many studies have revealed that for many natural images, the dimension of the patch manifold is usually very low [14, 15, 16, 9, 3, 13]. Thus we can use the

dimension of the patch manifold as a regularization to recover the image:

$$\min_{\substack{\mathbf{u}\in\mathbb{R}^{m\times n},\\ \mathcal{M}\subset\mathbb{R}^d}} \dim(\mathcal{M}), \qquad \text{subject to:} \quad \mathbf{b}=\mathbf{\Phi u}+\varepsilon, \qquad P(\mathbf{u})\subset\mathcal{M}, \tag{2.2}$$

where $\dim(\mathcal{M})$ is the dimension of the manifold $\mathcal{M}$, $\mathbf{b}$ is the observation, $\mathbf{\Phi}$ is a measurement operator, ε represents the noise perturbation.

In LDMM, the dimension of the patch manifold is calculated using the following formula:

PROPOSITION 2.1. *Let $\mathcal{M}$ be a smooth submanifold isometrically embedded in $\mathbb{R}^d$. For any $\boldsymbol{x}=(x_1,x_2,\cdots,x_d)\in\mathcal{M}$,*

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

where $\alpha_i(\boldsymbol{x})=x_i$ is the coordinate function.

Using the above proposition, the optimization problem (2.2) can be rewritten as

$$\min_{\substack{\mathbf{u}\in\mathbb{R}^{m\times n},\\ \mathcal{M}\subset\mathbb{R}^d}} \sum_{i=1}^{d}\|\nabla_{\mathcal{M}}\alpha_i\|^2_{L^2(\mathcal{M})}+\mu\|\mathbf{b}-\mathbf{\Phi u}\|_2^2, \qquad \text{subject to:} \quad P(\mathbf{u})\subset\mathcal{M}. \tag{2.3}$$

where μ is a parameter in the penalty term.

In [12], the above optimization problem is solved iteratively as follows:

- With a guess of the manifold $\mathcal{M}^k$ and a guess of the image $\mathbf{u}^k$ satisfying $P(\mathbf{u}^k)\subset\mathcal{M}^k$, compute the coordinate functions $\alpha_i^{k+1}, i=1,\cdots,d$ and $\mathbf{u}^{k+1}$,

  $$(\mathbf{u}^{k+1},\alpha_1^{k+1},\cdots,\alpha_d^{k+1})=\arg\min_{\substack{\mathbf{u}\in\mathbb{R}^{m\times n},\\ \alpha_1,\cdots,\alpha_d\in H^1(\mathcal{M}^k)}} \sum_{i=1}^{d}\|\nabla_{\mathcal{M}^k}\alpha_i\|^2_{L^2(\mathcal{M}^k)}+\mu\|\mathbf{b}-\mathbf{\Phi u}\|_2^2, \tag{2.4}$$
  $$\text{subject to:} \quad \alpha_i((\mathcal{P}\mathbf{u}^k)(x))=(\mathcal{P}_i\mathbf{u})(x), \quad x\in\overline{\Omega},$$

  where $(\mathcal{P}_i\mathbf{u})(x)$ is the ith element of patch $(\mathcal{P}\mathbf{u})(x)$.
- Update $\mathcal{M}$ by setting

  $$\mathcal{M}^{k+1}=\left\{(\alpha_1^{k+1}(\boldsymbol{x}),\cdots,\alpha_d^{k+1}(\boldsymbol{x})):\boldsymbol{x}\in\mathcal{M}^k\right\}. \tag{2.5}$$
- repeat these two steps until convergence.

Based on Bregman iteration and a standard variational approach, the key step to solve (2.4) is solving a Poisson equation on a point cloud. in [12], this equation is solved by the point integral method [10, 11, 18, 19]. LDMM using the point integral method has been shown to be very effective especially in image inpainting [12]. However, the computational cost of the original LDMM in [12] is relatively high. First, the linear system arising from the point integral method is usually not symmetric. It is not efficient to solve numerically. Secondly, in LDMM, we recover the image iteratively, based on completely nonlocal patches. We need many iterations, typically over 100, to get good reconstruction. To reduce the computational cost of LDMM, we propose two procedures in this paper. First, we use the weighted graph Laplacian [17] to replace the point integral method. The linear system given by WGL is symmetric and positive definite which is much easier to solve. Secondly, we use semi-local patches instead of nonlocal patches. Our numerical experiments show that semi-local patches could reduce the number of iterations tremendously, from over 100 to around 10.

3. Weighted Graph Laplacian. The weighted graph Laplacian (WGL) is a method to compute smooth interpolation on a point cloud, which was recently developed in [17]. The key idea in the weighted graph Laplacian is to separate the energy function in the graph Laplacian into two parts: one in the labeled set and the other in the unlabeled set. When the sample rate is low, the term in the unlabeled set overwhelms the other term, so that the continuity in the labeled set is sacrificed. To assure the continuity, one simple method is to add a weight to balance these two terms. One natural choice of the weight is the inverse of the sample rate, which gives the following optimization problem

(3.1)
$$\min_u \sum_{\boldsymbol{x}\in P\setminus S}\left(\sum_{\boldsymbol{y}\in P} w(\boldsymbol{x},\boldsymbol{y})(u(\boldsymbol{x})-u(\boldsymbol{y}))^2\right)+\frac{|P|}{|S|}\sum_{\boldsymbol{x}\in S}\left(\sum_{\boldsymbol{y}\in P} w(\boldsymbol{x},\boldsymbol{y})(u(\boldsymbol{x})-u(\boldsymbol{y}))^2\right),$$

with the constraint

$$u(\boldsymbol{x})=g(\boldsymbol{x}),\quad \boldsymbol{x}\in S,$$

where P is the point cloud and S is the labeled set, $w(\boldsymbol{x},\boldsymbol{y})$ is a given weight function and g is the given values on S.

The corresponding Euler-Lagrange equation of (3.1) is a linear system

$$\begin{aligned}\sum_{\boldsymbol{y}\in P}\left(w(\boldsymbol{x},\boldsymbol{y})+w(\boldsymbol{y},\boldsymbol{x})\right)\left(u(\boldsymbol{x})-u(\boldsymbol{y})\right)&\\ +\left(\frac{|P|}{|S|}-1\right)\sum_{\boldsymbol{y}\in S} w(\boldsymbol{y},\boldsymbol{x})(u(\boldsymbol{x})-u(\boldsymbol{y}))=0,\quad \boldsymbol{x}\in P\setminus S,&\\ u(\boldsymbol{x})=g(\boldsymbol{x}),\quad \boldsymbol{x}\in S.&\end{aligned}$$

This linear system is symmetric and positive definite. In addition, in WGL, a large positive term is added to the diagonal of the coefficient matrix, which makes the conjugate gradient method converge in fewer step.

4. Semi-local Patches. The semi-local patches are obtained by adding local coordinates to the nonlocal patches in Section 2 with a weight λ, i.e.

$$(\bar{\mathcal{P}}\mathbf{u})(x)=[(\mathcal{P}\mathbf{u})(x),\lambda x]\in\mathbb{R}^{d+2}, \tag{4.1}$$

and the semi-local patch set is

$$\bar{P}\mathbf{u}=\{(\bar{\mathcal{P}}\mathbf{u})(x): x\in\overline{\Omega}=\{1,2,\dots,m\}\times\{1,2,\dots,n\}\}. \tag{4.2}$$

where $\mathbf{u}\in\mathbb{R}^{m\times n}$ is an image, $x\in\overline{\Omega}=\{1,2,\dots,m\}\times\{1,2,\dots,n\}$ is the index of the pixel. The weight λ is used to get different locality in the semi-local patch. If $\lambda=0$, the semi-local patch is same as the nonlocal patch. If $\lambda\to\infty$, the patches are completely determined by local coordinates.

The semi-local patches have been successfully applied in many image processing problems [14, 23, 20, 21, 5]. In the geometrical point of view, if $\lambda\to\infty$, the patch set $\bar{P}(f)$ is parametrized by a local 2D coordinate, $x\mapsto\bar{\mathcal{P}}(\mathbf{u})(x)$. However, this parameterization is globally not injective and typically leads to high curvature variations and self-intersections. With this parametrization, the dimension of the manifold is very

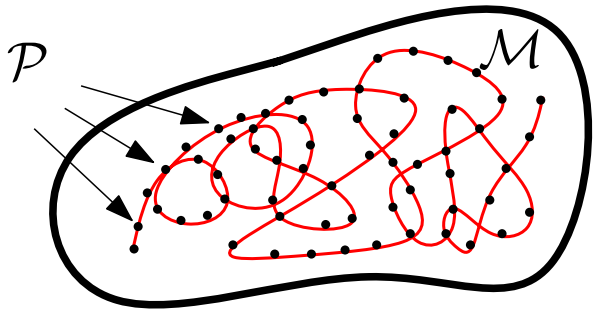

FIG. 1. *Diagram of patch set $\mathcal{P}$ (black points), local 2D parameterization (red curve) and global patch manifold parameterization.*

low, while the regularity is poor. If $\lambda = 0$, the underlying patch manifold may have higher dimension, however the manifold becomes smoother. Fig. 1 gives a diagram which shows these two different parameterizations. The idea in this paper is to find a good compromise between the dimension and the regularity of the patch manifold by choosing a proper weight λ. With a proper λ, our experiments show that LDMM is accelerateed significantly.

5. Image Inpainting. In this section, we apply LDMM with semi-local patches to image inpainting problems. In image inpainting we want to recover the image $\mathbf{u} \in \mathbb{R}^{m\times n}$ from its subsamples $\mathbf{u}|_\Omega$, where $\Omega \subset \overline{\Omega} = \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}$ is the subsampled domain. In this paper we consider the noise free case. Basically, we follow the algorithm in Section 2. The difference is that we use semi-local patches to construct weight functions w and the weighted graph Laplacian to discretize the optimization problem.

The key step in the algorithm of LDMM, as shown in Section 2, is to solve the following optimization problem:

$$\begin{aligned} (5.1)\qquad \min_{\mathbf{u}\in\mathbb{R}^{m\times n}} \quad & \sum_{i=1}^{d} \|\nabla_{\mathcal{M}}\alpha_i\|^2_{L^2(\mathcal{M})}, \\ \text{subject to:} \quad & \alpha_i(\bar{\mathcal{P}}(\mathbf{u}^k)(x)) = (\mathcal{P}_i\mathbf{u})(x), \quad x \in \overline{\Omega}, \quad i = 1, \cdots, d, \\ & \mathbf{u}(x) = \mathbf{b}(x), \qquad\quad x \in \Omega \subset \overline{\Omega}. \end{aligned}$$

where $\bar{\mathcal{P}}(\mathbf{u}^k)(x)$ is the semi-local patch of $\mathbf{u}^k$ at x, $(\mathcal{P}_i\mathbf{u})(x)$ is the ith element of patch $(\mathcal{P}\mathbf{u})(x)$.

Based on the weighted graph Laplacian, to get a correct solution, we need to put an extra weight on the sampled set. Notice that the sample set is different for different α_i. Applying WGL, we get the following discretized optimization problem:

$$\begin{aligned} (5.2)\qquad \min_{\mathbf{u}\in\mathbb{R}^{m\times n}} \quad & \sum_{i=1}^{d} \left(\sum_{x\in\overline{\Omega}\setminus\Omega_i} \sum_{y\in\overline{\Omega}} \overline{w}(x,y)((\mathcal{P}_i\mathbf{u})(x) - (\mathcal{P}_i\mathbf{u})(y))^2 \right. \\ & \left. + \frac{mn}{|\Omega|} \sum_{x\in\Omega_i} \sum_{y\in\overline{\Omega}} \overline{w}(x,y)((\mathcal{P}_i\mathbf{u})(x) - (\mathcal{P}_i\mathbf{u})(y))^2 \right), \end{aligned}$$

Algorithm 1 Image Inpainting

Require: A subsample image $\mathbf{u}|_\Omega$.
Ensure: A recovered image $\mathbf{u}$.

Generate initial image $\mathbf{u}^0$.

while not converge **do**

1. Generate semi-local patch set $\bar{P}(\mathbf{u}^k)$ from current image $\mathbf{u}^k$ with the weight λ in the semi-local patch.
2. Compute the weight function

$$\overline{w}^k(x,y) = w((\bar{\mathcal{P}}\mathbf{u}^k)(x), (\bar{\mathcal{P}}\mathbf{u}^k)(y)), \quad x, y \in \overline{\Omega}.$$

3. Update the image $\mathbf{u}^{k+1}$ by solving equation (5.4) with $\overline{w}^k(x,y)$.
4. $k \leftarrow k+1$.

end while

$\mathbf{u} = \mathbf{u}^k$.

with the constraints

$$\mathbf{u}(x) = \mathbf{b}(x), \qquad x \in \Omega \subset \overline{\Omega}.$$

$\Omega_i = \{x \in \overline{\Omega} : \quad (\mathcal{P}_i\mathbf{u})(x) \text{ is sampled}\}$ is the sampled set of α_i and

$$\overline{w}(x,y) = w((\bar{\mathcal{P}}\mathbf{u})(x), (\bar{\mathcal{P}}\mathbf{u})(y)), \tag{5.3}$$

where w is the given weight function.

Using a standard variational approach, we get the Euler-Lagrange equation of the above optimization problem:

$$\begin{aligned} \sum_{i=1}^{d} \mathcal{P}_i^* \left(\sum_{y\in\overline{\Omega}} (\overline{w}(x,y) + \overline{w}(y,x)) \left((\mathcal{P}_i\mathbf{u})(x) - (\mathcal{P}_i\mathbf{u})(y)\right) \right) \\ + \beta \sum_{i=1}^{d} \mathcal{P}_i^* \left(\sum_{y\in\Omega_i} \overline{w}(y,x) \left((\mathcal{P}_i\mathbf{u})(x) - (\mathcal{P}_i\mathbf{u})(y)\right) \right) = 0, \quad x \in \overline{\Omega}\backslash\Omega, \\ \mathbf{u}(x) = \mathbf{b}(x), \quad x \in \Omega, \end{aligned} \tag{5.4}$$

where $\mathcal{P}_i^*$ is the adjoint operator of $\mathcal{P}_i$ and $\beta = \frac{mn}{|\Omega|} - 1$.

Next, we will alpply Algorithm 1 to some image inpainting problems and show some results. In our calculations below, we take the weight $w(\boldsymbol{x}, \boldsymbol{y})$ as following:

$$w(\boldsymbol{x}, \boldsymbol{y}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{y}\|^2}{\sigma(\boldsymbol{x})^2}\right). \tag{5.5}$$

where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^{d+2}$ are semi-local patches, $\sigma(\boldsymbol{x})$ is chosen to be the distance between $\boldsymbol{x}$ and its 20th nearest neighbor, To make the weight matrix sparse, we truncate the weight to the 50 nearest neighbors. The patch size is 10×10. For each patch, the nearest neighbors are obtained by using an approximate nearest neighbor (ANN)

search algorithm. We use a k-d tree approach as well as an ANN search algorithm to reduce the computational cost. The linear system in Algorithm 1 is solved by the conjugate gradient method.

In the semi-local patches, the local coordinate is normalized to have the same amplitude as the image intensity,

$$(\bar{\mathcal{P}}\mathbf{u})(x) = [(\mathcal{P}\mathbf{u})(x), \lambda\bar{x}]$$

with

$$\bar{x} = \left(\frac{x_1\|(f|_\Omega)\|_\infty}{m}, \frac{x_2\|(f|_\Omega)\|_\infty}{n}\right),$$

$x = (x_1, x_2)$ and m, n are the sizes of the image.

Large λ makes it easier to compute the weight function w, because the searching of the nearest neighbors is confined to a local area. On the other hand, small λ gives more accurate results, since more global information is used. Based on this observation, in the computation with 10% subsamples, we gradually reduce λ in the iteration, more specially, λ is determined in the following way,

$$\lambda^{k+1} = \max(\lambda^k - 1, 3),$$

and the initial λ is set to be 10. In the computation with 5% subsamples, we fix λ to be 3, since more global information is required to get good reconstruction with lower sample rate. In all the computations, the number of iterations is fixed to be 10.

PSNR defined as follows is used to measure the accuracy of the results

$$\text{PSNR}(f, f^*) = -20\log_{10}(\|f - f^*\|/255), \tag{5.6}$$

where f^* is the ground truth.

Fig. 2 displays the results with 10% subsamples. GL and WGL are the methods presented in [17]. In all the examples, LDMM gives the best results. Comparing with WGL, the improvement in PSNR is not significant. However, we can see more differences visually. In the results of WGL, some times, there exist some spurious structures. As shown in Fig. 3, near the garlic, in the reconstruction of WGL, there is a white line which does not exist in the original image. In the result of LDMM, this white line is removed. We can also see this phenomenon in the other examples: for instance, the right-top corner in the image of boat, the sky near the roof in the image of castle.

In the results with 5% subsamples, the improvement of LDMM is more significant as shown in Fig. 4. We also study the effect of the weight λ in the semi-local patches. In Fig. 5, we show the reconstructions from 5% subsamples with different λ, $\lambda = 3, 10$ respectively. It turns out that different images have different preferences. As shown in Fig. 5, the image of Barbara prefers smaller λ, while the image of boat likes larger λ. This also makes sense intuitively. There is a large area with repeated patterns in the image of Barbara. This suggests the existence of a global structure, and we need more global information to reconstruct the image. Contrary to the image of Barbara, there is no repeated pattern in the image of boat. Too much global information may introduce some artificially repeated patterns which would corrupt the image. So far, we do not have a very precise procedure to select λ. In principle, reconstruction of repeated patterns prefers small λ, while reconstruction of local structures likes large λ.

FIG. 2. *Results of subsample image restoration from 10% subsamples.*

With the weighted graph Laplacian and the semi-local patches, we reduce the computational time of LDMM significantly. For instance, to reconstruct the image of Barbara, 256×256 from 10% subsamples, the original LDMM in [12] takes about 20 minutes, while the new algorithm takes about 5 minutes. In addition, we found that in the new algorithm, over 60% of computational time is spent in assembling the coefficient matrix in (5.4). There are many loops in assembling the coefficient matrix and our code is written in matlab. We may be able to reduce the computational time

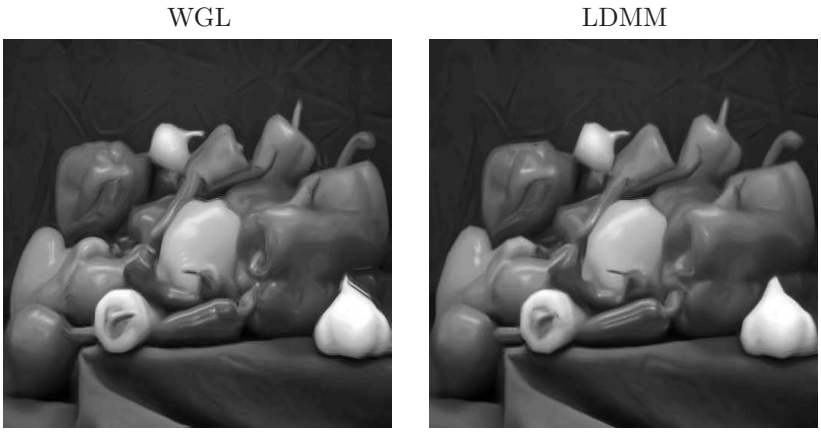

FIG. 3. *Reconstruction of the image of peppers in WGL and LDMM.*

further after optimizing this part of the code.

In addition to the computational time being reduced, the results are also better than those of the LDMM with nonlocal patches in [12]. From a geometrical point of view, the dimension becomes smaller in semi-local patches as explained in Section 4.

6. Image Denoising. In this section, we apply the LDMM to image denoising. Comparing with the image inpainting, in denoising, the constraints are different,

$$\mathbf{b}(x) = \mathbf{u}(x) + \varepsilon(x), \quad x \in \overline{\Omega} = \{1, 2, \ldots, m\} \times \{1, 2, \ldots, n\}, \tag{6.1}$$

where $\mathbf{u} \in \mathbb{R}^{m\times n}$ is the original image, $\mathbf{b} \in \mathbb{R}^{m\times n}$ is the noised image, ε is the noise. In this paper, we assume ε is the Gaussian noise.

In image denoising, the labeled set is same as the whole patch set. The weighted graph Laplacian reduces to the graph Laplacian. So, we go back to the original point integral method to solve the optimization problem.

Following the algorithm of LDMM in [12], we need to solve following optimization problem:

$$\begin{aligned} &\min_{\substack{\mathbf{u}\in\mathbb{R}^{m\times n},\\ \alpha_1,\cdots,\alpha_d}} \quad \sum_{i=1}^{d} \|\nabla_{\mathcal{M}}\alpha_i\|^2_{L^2(\mathcal{M})} + \gamma \sum_{x\in\overline{\Omega}} |\mathbf{b}(x) - \mathbf{u}(x)|^2, \\ &\text{subject to:} \quad \alpha_i(\bar{\mathcal{P}}(\mathbf{u}^k)(x)) = (\mathcal{P}_i\mathbf{u})(x), \quad x \in \overline{\Omega}, \quad i = 1, \cdots, d. \end{aligned} \tag{6.2}$$

where γ is a penalty weight related to the variance of the noise.

Or equivalently, we can impose the penalty on the patches,

$$\begin{aligned} &\min_{\substack{\mathbf{u}\in\mathbb{R}^{m\times n},\\ \alpha_1,\cdots,\alpha_d}} \quad \sum_{i=1}^{d} \|\nabla_{\mathcal{M}}\alpha_i\|^2_{L^2(\mathcal{M})} + \gamma \sum_{i=1}^{d}\sum_{x\in\overline{\Omega}} |\alpha_i(\bar{\mathcal{P}}(\mathbf{u}^k)(x)) - (\mathcal{P}_i\mathbf{b})(x)|^2, \\ &\text{subject to:} \quad \alpha_i(\bar{\mathcal{P}}(\mathbf{u}^k)(x)) = (\mathcal{P}_i\mathbf{u})(x), \quad x \in \overline{\Omega}, \quad i = 1, \cdots, d. \end{aligned} \tag{6.3}$$

Next, we split $\mathbf{u}$ and $\alpha_1, \cdots, \alpha_d$, and solve them separately.

- Compute $\alpha_i^{k+1},\ i = 1, \cdots, d$ by solving

$$\min_{\alpha_1,\cdots,\alpha_d} \quad \sum_{i=1}^{d} \|\nabla_{\mathcal{M}} \alpha_i\|^2_{L^2(\mathcal{M})} + \gamma \sum_{i=1}^{d} \sum_{x\in\overline{\Omega}} |\alpha_i(\bar{\mathcal{P}}(\mathbf{u}^k)(x)) - (\mathcal{P}_i \mathbf{b})(x)|^2 \tag{6.4}$$

- Update the image $\mathbf{u}^{k+1}$ by solving the least-squares problem

$$(\mathcal{P}_i \mathbf{u})(x) = \alpha_i(\bar{\mathcal{P}}(\mathbf{u}^k)(x)), \quad i = 1, \cdots, d, \quad x \in \overline{\Omega}. \tag{6.5}$$

In (6.4), $\alpha_1, \cdots, \alpha_d$ are decoupled, we can solve them one by one. For each α_i, we need to solve following type of problem:

$$\min_{u\in H^1(\mathcal{M})} \quad \|\nabla_{\mathcal{M}} u\|^2_{L^2(\mathcal{M})} + \gamma \sum_{\boldsymbol{y}\in P} |u(\boldsymbol{y}) - v(\boldsymbol{y})|^2, \tag{6.6}$$

where u can be any α_i, $\bar{P} = \bar{P}(\mathbf{u}^k)$ is the semi-local patch set of $\mathbf{u}^k$ and $v(\boldsymbol{y})$ is a given function on $\bar{P}(\mathbf{u}^k)$ corresponding to $(\mathcal{P}_i \mathbf{b})(x)$.

By a standard variational approach, we know that the solution of (6.6) can be obtained by solving the following PDE

$$\begin{cases} -\Delta_{\mathcal{M}} u(\boldsymbol{x}) + \gamma \sum_{\boldsymbol{y}\in\bar{P}} \delta(\boldsymbol{x}-\boldsymbol{y})(u(\boldsymbol{y}) - v(\boldsymbol{y})) = 0, \quad \boldsymbol{x} \in \mathcal{M} \\ \dfrac{\partial u}{\partial \mathbf{n}}(\boldsymbol{x}) = 0, \quad \boldsymbol{x} \in \partial\mathcal{M}. \end{cases} \tag{6.7}$$

where $\partial\mathcal{M}$ is the boundary of $\mathcal{M}$ and $\mathbf{n}$ is the out normal of $\partial\mathcal{M}$. If $\mathcal{M}$ has no boundary, $\partial\mathcal{M} = \emptyset$.

Using the point integral method, the above Laplace-Beltrami equation is discretized as follows [12],

$$\sum_{\boldsymbol{y}\in\bar{P}} w(\boldsymbol{x},\boldsymbol{y})(u(\boldsymbol{x}) - u(\boldsymbol{y})) + \bar{\gamma} \sum_{\boldsymbol{y}\in\bar{P}} w(\boldsymbol{x},\boldsymbol{y})(u(\boldsymbol{y}) - v(\boldsymbol{y})) = 0, \quad \boldsymbol{x} \in \bar{P}, \tag{6.8}$$

$\bar{\gamma}$ is a parameter related to γ. We only need to set $\bar{\gamma}$ in the computation.

Using this discretization, we get an iterative algorithm (Algorithm 2) for image denoising.

In the rest of this section, we will show some results on image denoising. We use the same weight given in (5.5). The patch size is 10×10. The number of iterations is 2. The weight λ in the semi-local patches is 3. In our tests, the noise ε is Gaussian noise with the standard deviation $\sigma = 100$. $\bar{\gamma} = 0.2$ in (6.8).

The results are shown in Fig. 7. BM3D is the famous method of block-matching with 3D collaborative filtering [6], which is known to give state-of-the-art result in image denoising. As we can see in Fig. 7, the results of BM3D is better in PSNR, although the difference is pretty small. Visually, we can see that LDMM seems to reconstruct the edges and textures sharper than BM3D as shown in Fig. 6. This difference can also be found in the result of the image of the house. LDMM seems to recover the edges of the house smoother and sharper, although the PSNR is lower than that of BM3D.

Actually, the philosophy in LDMM and BM3D is very similar. In Algorithm 2, there are also grouping, filtering and aggregation step as those in BM3D. The main difference is that in LDMM, we use a Laplace-Beltrami operator as a filter directly

Algorithm 2 Image Denoising

Require: A noisy image $\mathbf{b}$.
Ensure: A denoised image $\mathbf{u}$.

Let $\mathbf{u}^0 = \mathbf{b}$.

while not converge **do**

1. Generate semi-local patch set $\bar{P}(\mathbf{u}^k)$ from current image $\mathbf{u}^k$ with the weight λ in the semi-local patch.
2. Compute the weight function $w(\boldsymbol{x}, \boldsymbol{y}), \quad \boldsymbol{x}, \boldsymbol{y} \in \bar{P}(\mathbf{u}^k)$.
3. Compute $\alpha_i,\ i = 1, \cdots, d$ by solving (6.8) with $P = \bar{P}(\mathbf{u}^k)$, $v = (\mathcal{P}_i \mathbf{b})(x)$.
3. Update the image $\mathbf{u}^{k+1}$ by

$$\mathbf{u}^{k+1} = \left(\sum_{i=1}^{d} \mathcal{P}_i^* \mathcal{P}_i \right)^{-1} \left(\sum_{i=1}^{d} \mathcal{P}_i^* \alpha_i^{k+1} \right)$$

where $\mathcal{P}_i^*$ is the adjoint operator of $\mathcal{P}_i$.
4. $k \leftarrow k + 1$.

end while

$\mathbf{u} = \mathbf{u}^k$.

on the patches while hard thresholding and Wiener filtering is using in BM3D in the transform domain. LDMM provides a general framework for patch-based nonlocal methods. We can also use other operators, such as nonlocal total variation, to replace the Laplace-Beltrami operator in LDMM. In many problems of image processing, nonlocal TV, which corresponds to an L_1 regularization, has been proven to be very effective. In our subsequent research, we are planing to plug nonlocal TV into LDMM. The new model should have even better performance.

7. Conclusion and Future Work. In this paper, we improve the LDMM using the semi-local patches and the weighted graph Laplacian. The improvement is two fold. First, the computational time is reduced significantly. Second, the results in image inpainting and denoising are also better than those in the original LDMM.

LDMM provides a general framework for the nonlocal methods of image processing. We are working to integrate different methods, such as nonlocal TV, ∞-Laplacian, within the framework of LDMM. We are also trying to apply LDMM to different kinds of image processing problem in addition to inpainting and denoising, such as deblurring.

REFERENCES

[1] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition(CVPR)*, pages 60–65, 2005.

[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] G. Carlsson, T. Ishkhanov, V. de Silva, and A. Zomorodian. On the local behavior of spaces of natural images. *International Journal of Computer Vision*, 76:1–12, 2008.

[4] P. Chatterjee and P. Milanfar. Patch-based near-optimal image denoising. *Image Processing, IEEE Transactions on*, pages 1635–1649, 2012.

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24:603–619, 2002.

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Trans. Image Process.*, 16:2080–2095, 2007.

[7] G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Model. Simul.*, 7:1005–1028, 2008.

[8] A. Kheradmand and P. Milanfar. A general framework for regularized, similarity-based image restoration. *Image Processing, IEEE Transactions on*, pages 5136–5151, 2014.

[9] A. B. Lee, K. S. Pedersen, and D. Mumford. The nonlinear statistics of high-contrast patches in natural images. *International Journal of Computer Vision*, 54:83–103, 2003.

[10] Z. Li and Z. Shi. A convergent point integral method for isotropic elliptic equations on point cloud. *SIAM: Multiscale Modeling Simulation*, 14:874–905, 2016.

[11] 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*.

[12] S. Osher, Z. Shi, and W. Zhu. Low dimensional manifold model for image processing. *Technical Report, CAM report 16-04, UCLA*, 2016.

[13] J. A. Perea and G. Carlsson. A klein-bottle-based dictionary for texture representation. *International Journal of Computer Vision*, 107:75–97, 2014.

[14] G. Peyré. Image processing with non-local spectral bases. *Multiscale Model. Simul.*, 7:703–730, 2008.

[15] G. Peyré. Manifold models for signals and images. *Computer Vision and Image Understanding*, 113:248–260, 2009.

[16] G. Peyré. A review of adaptive image representations. *IEEE Journal of Selected Topics in Signal Processing*, 5:896–911, 2011.

[17] Z. Shi, S. Osher, and W. Zhu. Weighted graph laplacian and image inpainting. *Technical Report, CAM report 16-61, UCLA*, 2016.

[18] Z. Shi and J. Sun. Convergence of the point integral method for poisson equation on point cloud. *arXiv:1403.2141*.

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

[20] S. M. Smith and J. M. Brady. Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 23:45–78, 1997.

[21] A. Spira, R. Kimmel, and N. Sochen. A short time beltrami kernel for smoothing images and manifolds. *IEEE Trans. Image Processing*, 16:1628–1636, 2007.

[22] H. Talebi and P. Milanfar. Global image denoising. *Image Processing, IEEE Transactions on*, pages 755–768, 2014.

[23] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *Proceedings of the Sixth International Conference on Computer Vision*, page 839, 1998.

[24] R. Yin, T. Gao, Y. M. Lu, and I. Daubechies. A tale of two bases: Local-nonlocal regularization on image patches with convolution framelets. *arXiv:1606.01377*, 2016.

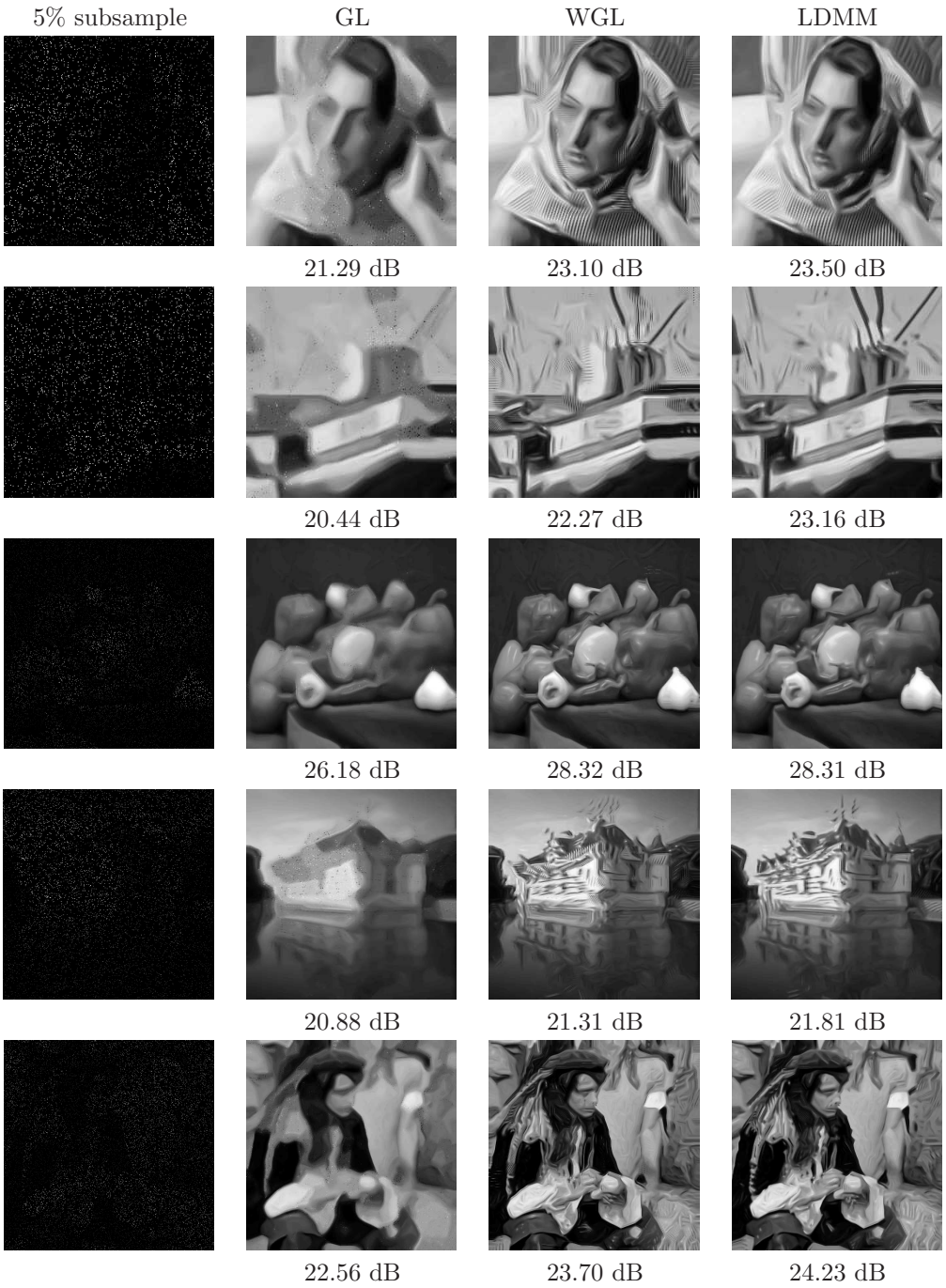

FIG. 4. *Results of subsample image restoration from 5% subsamples.*

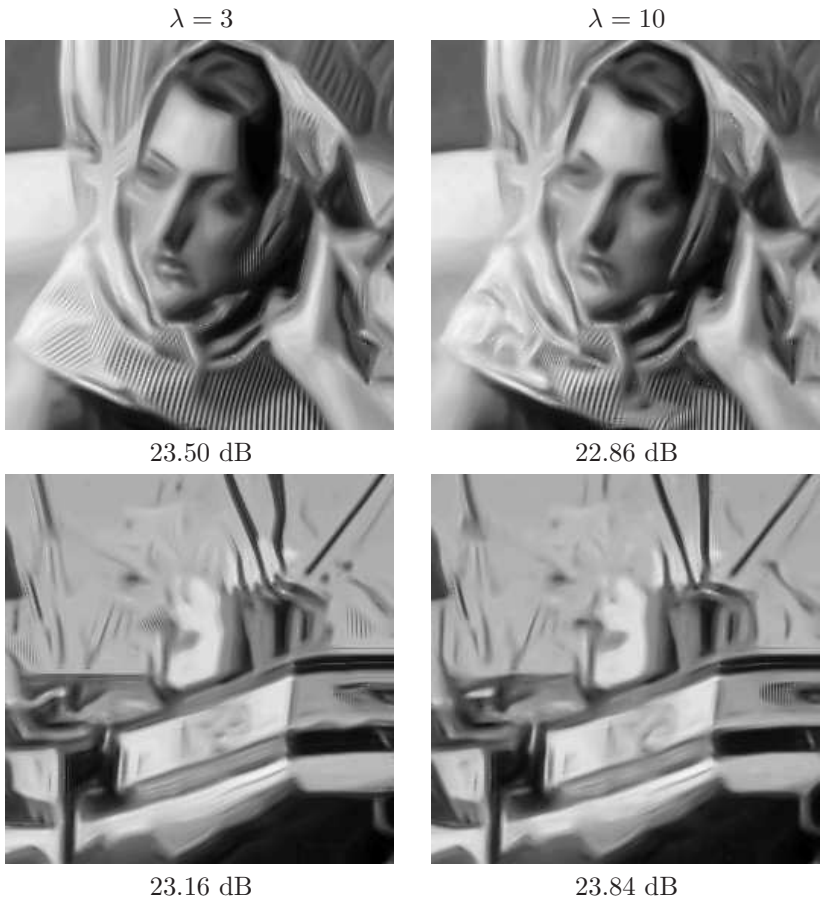

FIG. 5. *Results of subsample image restoration with different weight in the semi-local patches.*

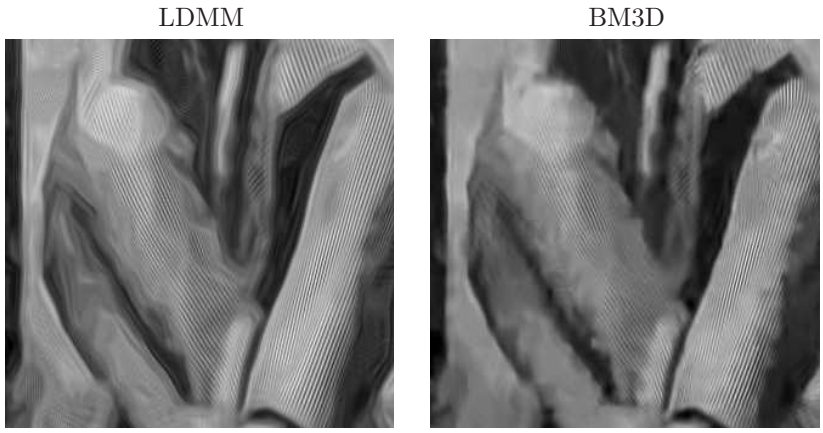

FIG. 6. *Fragment of the denoised image of Barbara using LDMM and BM3D.*

FIG. 7. *Results of image denoising.*