Impulsive Noise and Gaussian-Impulsive Mixed Noise Removal by Low Dimensional Manifold Model [*]

Yifan Chen[†] Banghua Zhu[‡] Zuoqiang Shi[§]

Abstract

In this paper, we apply the low dimensional manifold model (LDMM) to remove the impulsive noise from a single image. Like many nonlocal methods, LDMM is a patch-based model in which a patch set is constructed by grouping similar pixels. In LDMM, these patches are assumed to lie on a low dimensional manifold. We remove the noise by fitting the image onto an underlying manifold and minimize its dimension. The key step of LDMM is to solve the Laplace-Beltrami equations in the point cloud formed by these patches. We use the novel point integral method (PIM) to tackle it. By iteratively solving these Partial Differential Equations and updating the manifold, LDMM based method can restore corrupted image precisely. Massive experiments on both gray and color noisy images show that the proposed algorithm gives state-of-the-art results especially in the case of high-density impulsive noise. We also apply our algorithm to remove Gaussian-impulsive mixed noise and get very competitive results.

1 Introduction

During acquisition and transmission, images are always corrupted by noise. In real camera systems, the noise has various sources and can be quite complex. In order to estimate the original image from its noise corrupted observation, different noise models are proposed. Additive white Gaussian noise (AWGN) and impulsive noise (IN) are among the most common noise models. Generally, thermal motion of electrons in camera sensors and circuits could lead to AWGN, and malfunctioning pixels in camera sensors, faulty memory locations in hardware or bit errors in transmission will result in IN, which means AWGN and IN are pervasive in real camera images. Many papers have been published on removing impulsive noise [1, 15, 22, 12, 32, 4, 36, 5, 28, 9, 25]. The multiple sources of noise has also led to a variety of Gaussian-impulsive mixed noise removal algorithms [35, 21, 19, 14, 33, 24, 11, 20].

In order to well separate the noise from the images, we need a good model of the natural images. Recently, non-local methods are attracting more and more attention. In non-local methods, small patches instead of the pixels are used to model the images. Usually, the patches are overlapped with each other, so the collection of them has more redundancy than a single image. Many nonlocal methods, such as Adaptive Median Filter [12], BM3D [6], Weighted Encoding with Sparse Nonlocal Regularization (WESNR) [14], have been proven to be very effective in image denoising.

LDMM is also a patch-based method. It was first proposed in [23] to deal with general image processing problems. Later, it was also successfully applied in hyper spectral image reconstruction [29] and scientific

[*] Research supported by NSFC Grant 11671005.

[†] Department of Mathematical Science, Tsinghua University, chenyifan14@mails.tsinghua.edu.cn

[‡] Department of Electrical Engineering, Tsinghua University, 13aeon.v01d@gmail.com

[§] Department of Mathematical Science & Yau Mathematical Sciences Center, Tsinghua University, zqshi@mail.tsinghua.edu.cn

data interpolation [37]. In LDMM, the patches of the image are assumed to lie in an underlying smooth manifold, which is the so-called patch manifold. A lot of research reveals that the dimension of the patch manifold is much lower than its ambient Eucleadian space. Based on this observation, the dimension of the patch manifold seems to be a good regularization for image restoration. In LDMM, the image is reconstructed by looking for the patch manifold with the smallest dimension. Using a simple formula in differential geometry and the split Bregman iteration [10], an effective algorithm has been developed in LDMM.

In this paper, we generalize LDMM to deal with the image with Gaussian-impulsive mixed noise. The main idea is to use LDMM to model the image and use the sparse structure to handle the impulsive noise. Let $g = f + n + r$ be the noisy image, where f is the original image, n and r are Gaussian and impulsive noise, respectively. Formally, the model we used in this paper can be written as

$$\min_{f,r,\mathcal{M}} \text{dim}\mathcal{M} + \lambda\|f - g - r\|_2^2 + \mu\|r\|_0, \ \text{ s.t. } \mathcal{P}f \subset \mathcal{M} \tag{1}$$

where $\mathcal{P}f$ denotes the patch set of f which will be defined in the subsequent section.

One key observation in LDMM is that the dimension of the manifold has a simple formula to compute (4). Using this formula, we can rewrite the formal model (1) to a mathematical model. The model is solved by Point Integral Method [18] and split Bregman iteration. The details of the algorithm will be given in the subsequent sections.

1.1 Related work

LDMM is closely related to non-local methods. After the nonlocal method was first introduced in the nonlocal mean (NLM) method [2] to remove Gaussian noise, many other nonlocal methods were developed and achieved great success in image denoising, including the celebrated BM3D method [6] which establishes a benchmark in Gaussian noise removal. In the past few years, nonlocal Bayesian method [16] was proposed to incorporate the information of the local tangent space to get better performance to remove Gaussian noise.

In some sense, LDMM is a generalization of the well-known low rank model. Based on the low rank model, robust PCA [3] is proved to be a powerful method to separate the low rank background and sparse perturbation. This model is very effective for video data or hyper spectral data since they fit the low rank model very well [13, 8]. However, the image usually does not inherit low rank structure. LDMM seems to be a good generalization to model a single image.

Besides LDMM, there are other methods to exploit the low dimensional property of the patch manifold. One approach is to estimate the dimension of the tangent space using neighbor patches. Many methods were developed based on this model [14, 33, 34].

The rest of the paper is organized as follows. A brief introduction to LDMM and point integral method is given in Section 2. The noise removal model and algorithm is introduced in Section 3. In Section 4, extensive numerical experiments are conducted to demonstrate the performance of the proposed method. Discussion and conclusions are made in Section 5.

2 Low Dimensional Manifold Model

2.1 Patch manifold and Dimension Analysis

We now introduce the mathematical model of the patch manifold. For simplicity, we only consider gray images here. The argument can be easily applied to color images.

A gray image can be viewed as a 2D matrix $f \in \mathbb{R}^{m\times n}$ with entries between 0 to 255. We take the sub-matrix of f and vectorize it to get the patch. Here each sub-matrix is of size $s_1 \times s_2$. Let $x \in \Omega =$

$\{1, 2, ..., m\} \times \{1, 2, ..., n\}$ be the coordinate of the top-left pixel of the sub-matrix on f. This sub-matrix is denoted by M_x. We adopt a replicate padding for f to make every M_x well-defined. The $s_1 \times s_2$ patch at x is then defined as its vectorization, i.e. $\mathcal{P}f(x)$ =vec(M_x) where $\mathcal{P}$ is an operator from an image to its patch set such that

$$\mathcal{P}f = \{\mathcal{P}f(x) : x \in \Omega\} \subset \mathbb{R}^d, \quad d = s_1 \times s_2. \tag{2}$$

The operator $\mathcal{P}_i$ is an automorphism of $\mathbb{R}^{m\times n}$ with $\mathcal{P}_i f(x) = (\text{vec}(M_x))_i$.

From the geometrical point of view, the patch set is a point cloud in $\mathbb{R}^d$. A lot of research reveals that the patch set of the natural images always concentrate around a low dimensional smooth manifold embedded in $\mathbb{R}^d$ [26, 27]. Intuitively if two parts of a picture are very similar, the corresponding patches will be close to each other in $\mathbb{R}^d$. Therefore, $\mathcal{P}f$ consists of many clusters and each cluster is locally "flat". One natural model is to assume that each cluster samples a low dimension smooth manifold. In this sense, patch manifold may not be conventional manifold defined in mathematics. It is more likely to be the collection of many low dimensional manifolds. The dimension of $\mathcal{M}$ is defined as the integration of the dimension of its smooth pieces, i.e.

$$\text{dim}\mathcal{M} = \int_{\mathcal{M}} \text{dim}\mathcal{M}(\mathbf{p})\text{d}\mathbf{p}. \tag{3}$$

where $\text{dim}\mathcal{M}(\mathbf{p})$ is the dimension of $\mathcal{M}$ at $\mathbf{p}$.

In differential geometry, We have a simple fomula to compute the dimension pointwisely,

$$\text{dim}\mathcal{M}(\mathbf{p}) = \sum_{i=1}^{d} ||\nabla_{\mathcal{M}}\alpha_i(\mathbf{p})||_2^2 \tag{4}$$

where α_i is the i-th coordinate function of $\mathcal{M}$, i.e. $\alpha_i(\mathbf{p}) = \mathbf{p}_i$.

LDMM is based on the observation that the dimension of $\mathcal{M}$ is small and uses this as *a prior* information for regularization. Based on standard variational argument, $\|\nabla_{\mathcal{M}}\alpha_i(\mathbf{p})\|_2^2$ gives Laplace-Beltrami operator in manifold. However, in LDMM, we do not know the closed form of the patch manifold. What we have is just one sample of $\mathcal{M}$ which is the collection of the patches. Then we need to discretize Laplace-Beltrami operator on point cloud in high dimensional space. Here we use the point integral method (PIM) [18] to discretize the Laplace-Beltrami operator.

2.2 Point Integral Method

In this section, we briefly introduce the PIM method to solve Laplace-Beltrami equation on $\mathcal{M}$ only using the point cloud $\mathcal{P}f$. We refer to [18, 17] for comprehensive introductions of the point integral method.

Consider the following Laplace-Beltrami equation:

$$\begin{cases} -\Delta_{\mathcal{M}} u(x) = v(x), & x \in \mathcal{M} \\ \quad \dfrac{\partial u}{\partial \mathbf{n}}(x) = 0, & x \in \partial\mathcal{M} \end{cases} \tag{5}$$

where v is a given function, $\partial\mathcal{M}$ is the boundary of $\mathcal{M}$, $\mathbf{n}$ is the out normal of $\partial\mathcal{M}$.

The key ingredient of PIM is an integral approximation of the Laplace-Beltrami operator

$$\int_{\mathcal{M}} \Delta u(y)\bar{R}_t(x, y)\text{dy} = -\frac{1}{t}\int_{\mathcal{M}} (u(x) - u(y))R_t(x, y)\text{dy} + 2\int_{\partial\mathcal{M}} \frac{\partial u(y)}{\partial \mathbf{n}}\bar{R}_t(x, y)\text{ds} + O(t^{1/4})$$

where $t > 0$ is a small parameter, $R : \mathbb{R}^+ \to \mathbb{R}^+$ is a positive integrable function and

$$R_t(x, y) = R\left(\frac{\|x - y\|^2}{4t}\right). \tag{6}$$

$\bar{R}$ is the primitive function of R, i.e.

$$\bar{R}(r) = \int_r^\infty R(s)\mathrm{ds} \quad \text{and} \quad \bar{R}_t(x,y) = \bar{R}\left(\frac{\|x-y\|^2}{4t}\right) \tag{7}$$

We usually set $R(r) = e^{-r}$, which leads to

$$R_t(x,y) = \bar{R}_t(x,y) = \exp\left(\frac{\|x-y\|^2}{4t}\right). \tag{8}$$

Using above integral approximation and dropping the $O(t^{1/4})$ error term we get an integral equation without any differential operators inside:

$$\int_{\mathcal{M}} (u(x)-u(y))R_t(x,y)\mathrm{dy} - t\int_{\mathcal{M}} \bar{R}_t(x,y)v(y)\mathrm{d}y = 0. \tag{9}$$

This integral equation is very easy to be discretized on point cloud $\mathcal{P}f$. For simplicity, we assume that $\mathcal{P}$ is uniformly distributed. Then the integral equation becomes a linear system by directly replacing the integral by summation over $\mathcal{P}$. The solution of the linear system provides an approximate solution of the original Laplace-Beltrami equation. The details are deferred to the algorithm section.

3 Mixed Noise Removal

In this section we introduce the model for mixed noise removal. Denote $g = f + n + r$ to be the corrupted image, where f is the clean image, n and r are Gaussian and impulsive noise respectively. Our goal is to recover f from g.

There are two types of impulsive noise: random-valued impulsive noise and salt-and-pepper noise. In the random-valued impulsive noise, the intensity of the noise is uniformly distributed in $[0, 255]$ and the position is selected at random. Here, we consider the case that only a small portion of the pixels are corrupted. Thus we add a sparse penalty on the noise part. For the Gaussian noise, it is well known that l^2 penalty could handle it very well. Combining with the low dimensional regularization of the image, we get an model as follows:

$$\begin{aligned} \min_{f,r,\mathcal{M}} \quad & \sum_{i=1}^d ||\nabla_{\mathcal{M}}\alpha_i||^2_{L^2(\mathcal{M})} + \lambda\|f-g-r\|_2^2 + \mu\|r\|_0, \\ & \text{subject to: } \mathcal{P}f \subset \mathcal{M}. \end{aligned}$$

We want to remark that although we have sparse assumption in above model, our method is shown to be able to handle very heavy impulsive noise. In the numerical experiments, we test the case that 90% of the pixels are corrupted, in which case the sparsity argument does not hold any more. However we still get reasonable recovery from the corrupted image.

In salt-and-pepper noise, the intensity of the noise is either 0 or 255. We can detect the position of the noise and deal with the noise removal as an inpainting problem. Let Λ denotes its positions and $\mathrm{P}_{\Lambda^c}f$ is the projection of f into the region without impulsive noise. In this case, our model becomes

$$\begin{aligned} \min_{f,\mathcal{M}} \quad & \sum_{i=1}^d ||\nabla_{\mathcal{M}}\alpha_i||^2_{L^2(\mathcal{M})} + \lambda\|\mathrm{P}_{\Lambda^c}(f-g)\|_2^2, \\ & \text{subject to: } \mathcal{P}f \subset \mathcal{M} \end{aligned}$$

In this paper, we use the adaptive median filter (AMF) to detect the salt and pepper noise.

3.1 Algorithm

To solve the optimization problems in the previous section, we first transform them into a patch-based formulation.

Denote

$$\begin{aligned}
\|r\|_0 &= \frac{1}{d}\sum_{i=1}^{d}\|\mathcal{P}_i r\|_0 \\
\|\mathrm{P}_{\Lambda^c}(f-g)\|_2^2 &= \frac{1}{d}\sum_{i=1}^{d}\|\mathrm{P}^{(i)}(\mathcal{P}_i f-\mathcal{P}_i g)\|_2^2 \qquad (10)\\
\|f-g-r\|_2^2 &= \frac{1}{d}\sum_{i=1}^{d}\|\mathcal{P}_i f-\mathcal{P}_i g-\mathcal{P}_i r\|_2^2
\end{aligned}$$

where $\mathrm{P}^{(i)}$ is a projection operator on $\mathbb{R}^{m\times n}$ and $\mathrm{P}^{(i)}\mathcal{P}_i = P_{\Lambda^c}$.

Ignoring some constants the optimization becomes

$$\begin{aligned}
&\min_{f,\mathcal{M},\alpha_i}\sum_{i=1}^{d}\Big(||\nabla_{\mathcal{M}}\alpha_i||^2_{L^2(\mathcal{M})}+\phi(\mathcal{P}_i f)\Big) \\
&\text{s.t.}\quad \mathcal{P}f\subset\mathcal{M},\quad \alpha_i(\mathcal{P}f(x))=\mathcal{P}_i f(x)\quad \text{for all } x\in\Omega
\end{aligned}\qquad (11)$$

where for salt and pepper noise

$$\phi^{\mathrm{sp}}(\mathcal{P}_i f)=\bar{\lambda}\|\mathrm{P}_{\Lambda^c}(\mathcal{P}_i f-\mathcal{P}_i g)\|_2^2 \qquad (12)$$

and for random-valued impulsive noise

$$\phi^{\mathrm{rv}}(\mathcal{P}_i f)=\min_r \bar{\lambda}\|\mathcal{P}_i f-\mathcal{P}_i g-\mathcal{P}_i r\|_2^2+\bar{\mu}\|\mathcal{P}_i r\|_0 \qquad (13)$$

with $\bar{\lambda}$ is a parameter depends on the noise level.

3.2 Iterative algorithm

We solve (11) in a coordinate-wise fashion, i.e. update $\mathcal{M}$ and α_i, f iteratively. The algorithm writes

1. With current f^k and its patch manifold $\mathcal{M}^k$, compute $\alpha_i^{k+1}, i=1,...,d$ by

$$(\alpha_1^{k+1},...,\alpha_d^{k+1})=\mathrm{argmin}\sum_{i=1}^{d}\Big(||\nabla_{\mathcal{M}^k}\alpha_i||^2_{L^2(\mathcal{M}^k)}+\phi(\alpha_i(\mathcal{P}f^k)\Big) \qquad (14)$$

2. Update f^{k+1} by solving least square problems

$$\mathcal{P}_i f(x)=\alpha_i^{k+1}(\mathcal{P}f^k(x)),\quad 1\le i\le d,\forall x\in\Omega.$$

3. Update the implicit patch manifold using f^{k+1}

$$\mathcal{M}^{k+1}=\mathcal{P}f^{k+1}. \qquad (15)$$

4. repeat these three steps until convergence.

For the first step (14), α_i are separable, thus it is embarrassingly parallelable. For each i the problem has the form

$$\min_{\alpha_i} ||\nabla_{\mathcal{M}^k} \alpha_i||^2_{L^2(\mathcal{M}^k)} + \phi(\alpha_i) \tag{16}$$

Using standard variational approach, above optimization problem can be reduced to a Laplace-Beltrami equation on $\mathcal{M}^k$. We use PIM in section 2.2 to solve it. The weight function is chosen to be

$$W_{ij} = R_t(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{\sigma(x_i)^2}), \quad \forall x_i, x_j \in \mathcal{P}f^k, \tag{17}$$

where $\sigma(x_i)$ is a parameter depending on x_i.

The second step is standard least-squares problem which is easy to solve. In the third step, we generate the patches from the new image which also update the manifold in the sense that the patches sample the manifold.

Incorporating PIM solver in the algorithm, we get the algorithm used in the real computations.

1. With a guess of f^k , generate its patch $\mathcal{P}f^k$ and then the weight matrix on $\mathcal{P}f$ according to (17). Solving (14) by PIM to obtain $\alpha_i^{k+1}, i = 1, ..., d$.
2. Update f^{k+1} by solving least square problems
$$\mathcal{P}_i f(x) = \alpha_i^{k+1}(\mathcal{P}f^k(x)), \quad 1 \le i \le d, \forall x \in \Omega. \tag{18}$$
3. repeat these two steps until convergence.

Next, We will give a detailed implementation of (14) for salt-and-pepper noise and random-valued impulsive noise respectively. To simplify the notation, we use u to represent each α_i and $\mathcal{M}$ for $\mathcal{M}^k$.

3.3 Salt-and-pepper noise

For salt-and-pepper noise, the problem becomes

$$\min_u ||\nabla_{\mathcal{M}} u||^2_{L^2(\mathcal{M})} + \bar{\lambda}\|\mathrm{P}(u-l)\|_2^2 \tag{19}$$

where we use l to represent $\mathcal{P}_i g$ and P to represent $\mathrm{P}^{(i)}$. We implement split Bregman iteration to solve it.

- $u^{m+1} = \mathrm{argmin}_u \; ||\nabla_{\mathcal{M}} u||^2_{L^2(\mathcal{M})} + \gamma\|u - v^m + d^m\|_2^2$
- $v^{m+1} = \mathrm{argmin}_v \; \bar{\lambda}\|\mathrm{P}(v-l)\|_2^2 + \gamma\|u^{m+1} - v + d^m\|_2^2$
- $d^{m+1} = d^m + u^{m+1} - v^{m+1}$

The first step reduces to a Laplace-Beltrami equation:

$$\begin{cases} -\Delta_{\mathcal{M}} u(x) + \gamma \sum\limits_{y\in P} \delta(x-y)(u(y) - w(y)) = 0, x \in \mathcal{M} \\ \qquad\qquad\qquad\qquad\quad \dfrac{\partial u}{\partial n}(x) = 0, x \in \partial\mathcal{M} \end{cases} \tag{20}$$

where $w = v^k - d^k$. In PIM, the $\sigma(x)$ is set to be the distance between x and its 20th nearst neighbor. And (9) tells us that for each i and $x_i \in \mathcal{P}f$

$$\frac{|\mathcal{M}|}{N}\sum_{j=1}^N W_{ij}(u_i - u_j) + \gamma\sigma(x_i)^2 \sum_{j=1}^N W_{ij}(u_j - w_j) = 0 \tag{21}$$

It is equivalent to a matrix form

$$(L+\bar{\gamma}\Sigma W)u=\bar{\gamma}\Sigma W w, \tag{22}$$

where $W \in R^{|P|\times|P|}$ is the weight matrix. $D=\text{diag}(d_i)$ in which $d_i=\sum_{j=1}^{|P|} w_{ij}$ and $L=D-W$. $\Sigma=\text{diag}(\sigma(x_i)^2)$. The capital $\bar{\gamma}=\frac{\gamma t}{|\mathcal{M}|}$ is a hyperparameter in this method. The second and third step has closed-form solution.

3.4 Random-valued impulsive noise

For random-valued impulsive noise, we use h to represent the term $\mathcal{P}_i r$ and l to represent $\mathcal{P}_i g$. Then the problem for α_i in (14) can be written as

$$\min_{u,h} ||\nabla_{\mathcal{M}} u||^2_{L^2(\mathcal{M})}+\bar{\lambda}\|u-h-l\|_2^2+\bar{\mu}\|h\|_0 \tag{23}$$

We introduce an auxiliary variable $v=u-h-l$ and the problem turns to be

$$\min_{u,v,h} ||\nabla_{\mathcal{M}} u||^2_{L^2(\mathcal{M})}+\bar{\lambda}\|v\|_2^2+\bar{\mu}\|h\|_0 \quad \text{s.t.} \quad u=h+l+v \tag{24}$$

Applying split bregman iteration to obtain:

- $u^{m+1}=\text{argmin}_u \, ||\nabla_{\mathcal{M}} u||^2_{L^2(\mathcal{M})}+\gamma\|u-h^m-l-v^m+d^m\|_2^2$
- $v^{k+1}=\text{argmin}_v \, \bar{\lambda}\|v\|_2^2+\gamma\|u^{m+1}-h^m-l-v+d^m\|_2^2$
- $h^{m+1}=\text{argmin}_h \, \bar{\mu}\|h\|_0+\gamma\|u^{m+1}-h-l-v^{m+1}+d^m\|_2^2$
- $d^{m+1}=d^m+u^{m+1}-h^{m+1}-l-v^{m+1}$

The first one is solved by (21) if we set $w=h^m+l+v^m-d^m$. The second and third steps have closed-form solutions.

4 Experiment

In this section, we provide numerical validations for LDMM algorithms. We conducted extensive experiments and provides a thorough comparative analysis with the state-of-the-art method. We first give details on algorithm implementation and parameter setting, then test the LDMM method on 6 gray figures and TID2008 dataset, with impulsive noise and mixed noise respectively. Peak-to-noise-ratio (PSNR) is calculated and reported as a measure for the quality of the recovery. The code for reproducing our result is included in supplementary materials.

4.1 Parameter Setting

In the experiments, the patch size is set to be 10×10 and we take patch around each pixels. For impulsive noise only, $\bar{\lambda}=\frac{18000}{\sqrt{p_0}}$, here p_0 is the level of noise. For mixed noise, $\bar{\lambda}=\frac{3000}{\sqrt{p_0}}$ and $\bar{\mu}=\frac{10}{\sqrt{\sigma}}$. We have an outer loop for updating the weight matrix, and inner loop for split Bregman iteration. We iterate the inner loop 4 times, and the outer loop 2 times for small impulsive noise case, 5 times for high density impulsive noise.

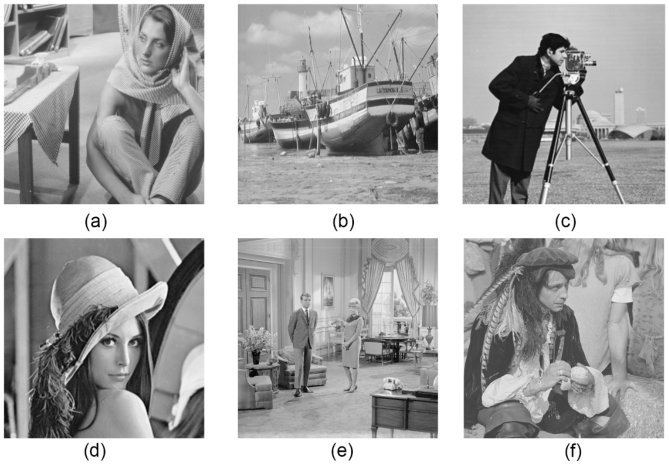

Figure 1: The 6 gray images for test. (a) barbara. (b) boat. (c) cameraman. (d) lena. (e) couple. (f) man. Cameraman and lena are of size 256×256. Other four figures are 512×512.

4.2 Synthetic Gray Images

We use 6 popular gray images in Figure 1, corrupt them with either impulsive noise or Gaussian-impulsive mixed noise and run denoising algorithms on the corrupted images.

4.2.1 Impulsive Noise

For impulsive-only noise, We compare LDMM algorithm with AMF [12], WESNR [14], l0TV-PADMM [36] and KALS [33]. All algorithms are run under their default settings. The maximum window size for adaptive median filter is set to 19. We generate the noisy pictures by corrupting it with different proportion of p_0 impulsive noise. We show the PSNR of the denoised images for both RVIN in Table 1 and SPIN in Table 2. Visual comparison is shown in Figure 2.

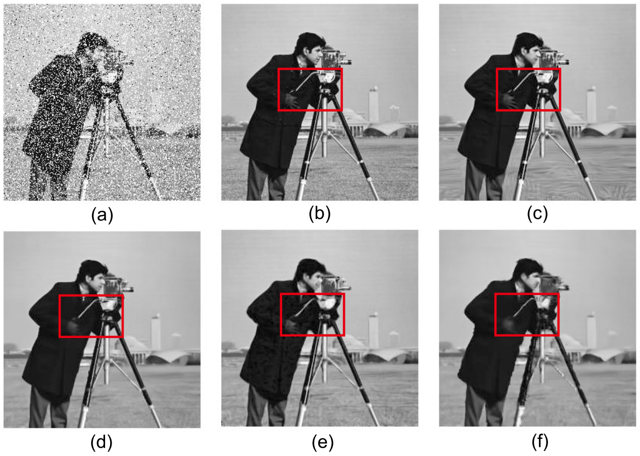

Figure 2: The denoised result for cameraman with $p_0 = 0.3$ salt-and-pepper noise. (a) the noisy image. (b) the original image. (c) LDMM. (d) WESNR. (e) l_0TV. (f) AMF.

Table 1: Performance (in PSNR) of denoising for 6 popular images with random-valued impulsive noise.

	p_0	LDMM	WESNR	l_0TV	KALS	AMF		p_0	LDMM	WESNR	l_0TV	KALS	AMF
barbara	0.1	32.41	29.90	23.80	**32.62**	24.89	boat	0.1	30.62	31.58	30.31	**32.07**	29.86
	0.3	**32.59**	25.49	22.59	24.88	23.40		0.3	**30.99**	26.67	26.96	28.49	26.98
	0.5	**31.57**	21.31	21.85	15.71	21.30		0.5	**29.47**	22.27	24.97	18.32	23.57
	0.7	**29.55**	17.43	21.24	13.28	19.09		0.7	**27.00**	18.95	23.34	14.63	20.77
	0.9	**26.91**	14.40	20.70	11.94	17.31		0.9	**24.62**	16.22	22.37	13.51	18.69
cameraman	0.1	**28.96**	27.84	25.29	28.38	26.38	lena	0.1	30.70	30.53	27.58	**31.07**	26.50
	0.3	**26.55**	23.24	23.31	23.48	23.80		0.3	**28.90**	24.07	25.39	27.23	19.54
	0.5	**24.52**	19.84	21.89	17.93	20.75		0.5	**27.37**	18.07	23.48	19.17	16.44
	0.7	16.01	17.78	**20.94**	15.35	18.18		0.7	**26.37**	14.11	21.88	12.86	14.59
	0.9	15.70	15.75	**20.01**	13.57	16.10		0.9	**23.12**	11.48	21.00	11.62	13.48
couple	0.1	30.58	31.72	30.71	**32.24**	29.63	man	0.1	31.49	31.50	28.51	**33.15**	31.08
	0.3	**30.85**	26.85	27.33	27.85	26.66		0.3	**30.88**	26.84	26.46	29.87	27.80
	0.5	**29.43**	22.63	25.28	17.97	23.47		0.5	**29.27**	22.80	25.08	22.44	23.81
	0.7	**27.27**	19.29	23.86	14.83	20.88		0.7	**27.46**	19.52	23.78	15.46	20.74
	0.9	**24.62**	16.74	22.63	13.96	18.85		0.9	**25.33**	17.36	22.80	13.63	18.63

Table 2: Performance (in PSNR) of denoising for 6 popular images with salt-and-pepper noise.

	p_0	LDMM	WESNR	l_0TV	KALS	AMF		p_0	LDMM	WESNR	l_0TV	KALS	AMF
barbara	0.1	34.33	31.52	24.40	**35.07**	28.82	boat	0.1	32.90	32.78	32.25	**34.25**	34.02
	0.3	**33.42**	31.39	23.22	30.48	26.88		0.3	**32.83**	31.25	29.31	31.15	31.01
	0.5	**32.76**	30.31	22.57	18.36	25.26		0.5	**32.18**	31.26	27.06	25.30	28.77
	0.7	**31.75**	28.93	22.20	14.98	24.02		0.7	**31.18**	30.20	25.35	16.50	27.16
	0.9	**30.49**	27.42	21.63	12.44	23.10		0.9	**30.19**	29.06	24.09	14.36	25.78
cameraman	0.1	30.13	29.79	26.32	29.46	**30.18**	lena	0.1	31.63	**32.51**	29.48	32.35	33.53
	0.3	**29.46**	28.71	24.65	26.74	27.72		0.3	31.14	**31.45**	27.32	28.57	29.51
	0.5	**29.00**	26.49	23.08	21.19	25.55		0.5	**30.53**	30.10	25.32	21.40	27.34
	0.7	**27.99**	25.29	22.57	16.61	24.00		0.7	**29.31**	28.79	23.94	15.27	25.27
	0.9	**26.89**	23.68	21.91	14.97	22.85		0.9	**28.70**	27.44	22.90	12.77	23.94
couple	0.1	32.66	32.66	32.52	**34.35**	33.81	man	0.1	32.81	32.66	29.43	34.55	**35.30**
	0.3	**32.65**	32.01	29.69	30.92	30.77		0.3	**32.70**	32.10	28.31	31.40	32.21
	0.5	**31.92**	31.20	27.33	24.56	28.56		0.5	**32.18**	31.36	27.00	25.17	29.99
	0.7	**31.14**	30.07	25.93	16.48	26.96		0.7	**31.25**	30.71	26.00	17.29	28.23
	0.9	**30.15**	29.20	24.49	14.63	25.65		0.9	**30.40**	29.84	25.08	14.98	26.93

4.2.2 Mixed Noise

LDMM also applies to mixed noise. In this case, we compare LDMM with AMF+BM3D [7], WESNR and KALS. PSNR of denoised images for Additive White Gaussian Noise (AWGN) + RVIN and AWGN + SPIN are given in Table 3 and Table 4 respectively. Visual comparison is shown in Figure 3.

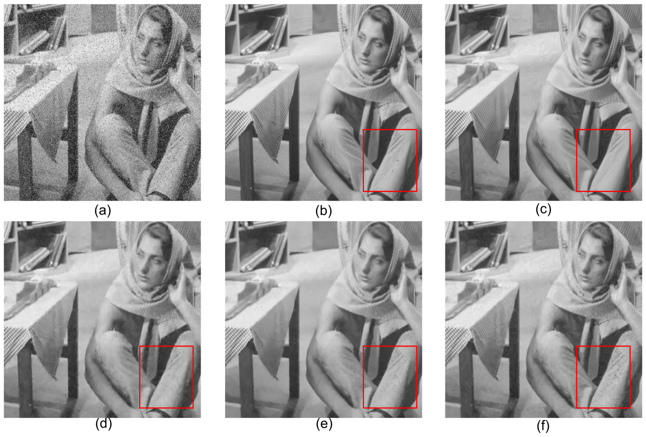

Figure 3: The denoised result for barbara with $p_0 = 0.1$ random-valued impulsive noise plus $\sigma = 20$ Gaussian noise. (a) noisy image. (b) original image. (c) LDMM. (d) KALS. (e) AMF+BM3D. (f) WESNR.

4.3 Synthetic Color Image Dataset: TID2008

For color images, the only difference is that patch becomes a three-dimensional cube with size $n \times n \times 3$. Here, we use the RGB representation. Since the source code of WESNR, LSM-NLR and KALS do not support color image denoising, we only compare our algorithm with Median Filter+BM3D on TID2008 dataset. The results are shown in Table 5.

4.4 Discussion

From the experiment results, we can see that LDMM is pretty robust to a wide range of noise, especially to high-density impulsive noise. Based on our experience in image inpainting, when most of the pixels are corrupted, classical nonlocal discretization method may have inconsistent issues [30, 31]. In LDMM, however, point integral method gives a consistent way to enforce the constraints in the intact pixels. This may give a formal explanation to the excellent performance of LDMM with high-density impulsive noise. The performance of LDMM method seems better in random-valued impulsive noise than that in salt-and-pepper noise. One possible explanation is that in removing salt-and-pepper noise, our current algorithm relies on the estimation of the position of noisy pixels, which may not be accurately located by adaptive median filter. On the other hand, the performance of LDMM is not so good in the case of high Gaussian noise and low-density impulsive noise. This is one problem we want to address in our future work.

5 Conclusion

In this paper, we propose a novel and robust low dimensional manifold model based denoising algorithm for impulsive and Gaussian-impulsive mixed noise. Extensive numerical experiments show that LDMM

Table 3: Performance (in PSNR) of denoising for 6 popular images with Gaussian noise plus random-valued impulsive noise.

	σ	p_0	LDMM	WESNR	AMF+BM3D	KALS		σ	p_0	LDMM	WESNR	AMF+BM3D	KALS
barbara	10	0.1	**32.14**	28.98	26.73	30.94	boat	10	0.1	30.66	30.42	29.14	**30.77**
	10	0.3	**30.89**	24.74	23.77	26.59		10	0.3	**29.31**	25.89	27.28	27.55
	10	0.5	**29.67**	20.84	21.88	16.17		10	0.5	**27.64**	21.81	24.25	17.87
	10	0.7	**27.92**	17.25	19.60	12.41		10	0.7	**25.37**	18.67	21.39	13.52
	10	0.9	**25.70**	14.30	17.70	11.03		10	0.9	**23.86**	16.05	19.19	12.44
	20	0.1	**29.79**	27.11	26.91	27.68		20	0.1	**28.77**	28.36	27.67	28.69
	20	0.3	**28.80**	25.00	23.32	26.51		20	0.3	**27.58**	25.92	26.22	25.76
	20	0.5	**27.59**	21.79	21.83	16.39		20	0.5	**26.18**	22.44	23.98	17.75
	20	0.7	**26.22**	17.15	19.87	11.58		20	0.7	**24.25**	19.02	21.57	12.03
	20	0.9	**24.40**	13.79	18.12	10.42		20	0.9	**23.09**	15.95	19.58	11.39
	50	0.1	**24.44**	21.62	23.64	24.24		50	0.1	23.83	22.71	**25.18**	24.88
	50	0.3	**23.11**	17.21	21.70	21.73		50	0.3	22.63	18.51	**23.30**	22.89
	50	0.5	**21.93**	13.57	19.84	19.23		50	0.5	**21.56**	14.93	21.20	20.38
	50	0.7	**20.45**	10.94	18.27	9.34		50	0.7	**20.59**	12.42	19.63	9.96
	50	0.9	**18.69**	9.48	17.10	9.49		50	0.9	**19.42**	10.29	18.40	9.64
cameraman	10	0.1	**28.03**	27.77	26.30	27.83	lena	10	0.1	29.60	29.51	27.61	**29.67**
	10	0.3	**25.15**	22.98	24.17	24.63		10	0.3	**27.11**	23.43	20.38	26.60
	10	0.5	**22.25**	19.41	21.25	16.70		10	0.5	**25.84**	17.89	16.79	20.48
	10	0.7	**19.86**	17.56	18.58	14.26		10	0.7	**23.24**	13.92	14.83	12.82
	10	0.9	**17.01**	15.64	16.40	12.82		10	0.9	**19.82**	11.41	13.67	11.23
	20	0.1	**27.24**	25.79	25.43	26.88		20	0.1	**28.40**	27.65	27.58	27.94
	20	0.3	**24.53**	22.47	23.81	23.88		20	0.3	**26.24**	23.22	22.40	25.17
	20	0.5	**21.96**	19.35	21.32	16.80		20	0.5	**24.83**	17.30	18.30	20.37
	20	0.7	**19.94**	17.40	18.85	13.16		20	0.7	**22.43**	13.60	15.80	11.80
	20	0.9	**17.32**	15.20	16.74	12.06		20	0.9	**20.21**	11.15	14.43	11.11
	50	0.1	23.25	18.60	**23.78**	23.88		50	0.1	23.60	20.11	**24.28**	24.18
	50	0.3	**21.59**	16.66	21.44	21.42		50	0.3	**22.38**	14.93	22.03	21.73
	50	0.5	**19.85**	14.27	19.10	18.90		50	0.5	**20.74**	11.08	19.67	19.27
	50	0.7	**18.22**	11.65	17.41	10.18		50	0.7	**18.76**	9.46	17.92	10.71
	50	0.9	**16.59**	9.96	15.96	10.25		50	0.9	**16.72**	8.37	16.76	9.96
couple	10	0.1	**30.65**	28.89	30.43	30.54	man	10	0.1	30.56	30.47	29.78	**30.98**
	10	0.3	**29.44**	26.03	26.98	26.91		10	0.3	**29.60**	26.10	27.91	28.41
	10	0.5	**28.22**	22.16	24.16	18.00		10	0.5	**28.53**	22.24	24.55	22.50
	10	0.7	**26.70**	19.06	21.52	13.49		10	0.7	**27.62**	19.29	21.39	13.71
	10	0.9	**24.54**	16.59	19.37	12.77		10	0.9	**25.69**	17.23	19.12	12.73
	20	0.1	**28.86**	28.14	27.48	28.47		20	0.1	**28.83**	28.50	28.15	28.76
	20	0.3	**27.72**	26.03	25.96	25.49		20	0.3	**27.76**	26.68	26.64	26.32
	20	0.5	**26.38**	23.10	23.81	17.02		20	0.5	**26.45**	23.78	24.16	22.55
	20	0.7	**24.98**	19.60	21.66	12.55		20	0.7	**25.72**	20.06	21.57	12.82
	20	0.9	**23.13**	16.50	19.75	11.54		20	0.9	**24.38**	17.29	19.56	11.59
	50	0.1	23.85	22.82	**24.96**	24.56		50	0.1	23.57	23.44	**25.53**	25.13
	50	0.3	22.62	19.15	**23.10**	22.70		50	0.3	22.68	20.00	**23.33**	23.01
	50	0.5	**21.67**	15.58	21.27	20.31		50	0.5	**21.91**	16.05	21.09	20.36
	50	0.7	**20.66**	12.83	19.79	9.98		50	0.7	**21.11**	13.34	19.43	10.04
	50	0.9	**19.47**	10.85	18.70	9.31		50	0.9	**19.62**	11.51	18.14	8.78

Table 4: Performance (in PSNR) of denoising for 6 popular images with Gaussian noise plus salt-and-pepper noise.

	σ	p_0	LDMM	WESNR	AMF+BM3D	KALS		σ	p_0	LDMM	WESNR	AMF+BM3D	KALS
barbara	10	0.1	29.92	30.41	28.70	**32.10**	boat	10	0.1	30.70	31.51	31.60	**31.69**
	10	0.3	30.01	**30.18**	27.05	29.29		10	0.3	29.86	**30.96**	30.18	29.63
	10	0.5	**29.74**	29.25	25.48	21.80		10	0.5	29.37	**30.19**	28.55	25.49
	10	0.7	**29.20**	27.91	24.21	13.17		10	0.7	28.85	**29.23**	27.16	14.42
	10	0.9	**28.46**	26.58	23.29	10.84		10	0.9	**28.14**	28.12	25.87	11.96
	20	0.1	28.31	27.84	27.68	**29.32**		20	0.1	29.03	28.83	29.26	**29.29**
	20	0.3	**28.35**	27.56	26.51	26.72		20	0.3	28.33	**28.46**	28.38	26.98
	20	0.5	**28.02**	26.38	25.13	21.78		20	0.5	27.93	**27.99**	27.32	23.53
	20	0.7	**27.43**	25.22	23.87	11.05		20	0.7	**27.28**	26.88	26.08	11.49
	20	0.9	**26.55**	24.36	22.89	9.07		20	0.9	**26.64**	26.16	25.01	9.35
	50	0.1	23.58	19.57	**24.02**	24.26		50	0.1	24.44	21.70	**25.41**	24.73
	50	0.3	**23.69**	15.47	23.19	21.40		50	0.3	24.25	17.53	**24.70**	22.31
	50	0.5	**22.91**	13.75	22.16	12.92		50	0.5	**23.58**	15.39	23.52	13.29
	50	0.7	**22.12**	12.81	21.21	10.83		50	0.7	**22.90**	14.29	22.41	10.41
	50	0.9	**21.17**	12.25	20.21	5.89		50	0.9	**22.13**	13.35	21.36	5.34
cameraman	10	0.1	28.35	29.13	**29.32**	28.46	lena	10	0.1	29.65	31.17	**31.37**	30.40
	10	0.3	27.48	**27.89**	27.40	26.64		10	0.3	28.61	**30.14**	29.01	27.89
	10	0.5	**27.01**	25.69	25.44	19.82		10	0.5	28.26	**29.04**	27.30	23.63
	10	0.7	**26.38**	24.73	24.02	15.20		10	0.7	27.46	**27.88**	25.35	13.49
	10	0.9	**25.51**	23.06	22.79	13.24		10	0.9	**27.05**	26.14	24.13	12.32
	20	0.1	27.18	26.07	**27.79**	27.37		20	0.1	28.23	28.41	**28.72**	28.39
	20	0.3	26.32	24.77	**26.40**	25.53		20	0.3	27.22	**27.71**	27.47	26.13
	20	0.5	**25.84**	23.26	24.96	21.99		20	0.5	26.67	**26.94**	26.27	23.27
	20	0.7	**25.12**	22.32	23.66	12.40		20	0.7	**26.07**	25.95	24.82	12.38
	20	0.9	**24.34**	20.98	22.51	10.41		20	0.9	**25.37**	23.87	23.68	10.30
	50	0.1	24.14	18.02	**24.18**	23.79		50	0.1	24.62	19.61	**24.69**	24.19
	50	0.3	23.11	15.49	**23.35**	21.54		50	0.3	23.88	16.41	**23.94**	21.84
	50	0.5	**22.58**	14.14	22.41	13.89		50	0.5	**23.09**	14.98	22.91	14.69
	50	0.7	**21.82**	13.31	21.43	10.54		50	0.7	**22.39**	13.76	21.92	10.27
	50	0.9	**21.01**	12.66	20.38	5.58		50	0.9	**21.82**	13.10	21.11	5.32
couple	10	0.1	30.71	31.33	**31.63**	31.57	man	10	0.1	30.93	31.47	**31.84**	31.61
	10	0.3	29.88	**30.76**	30.22	29.41		10	0.3	30.25	**30.94**	30.69	29.82
	10	0.5	29.39	**30.04**	28.59	25.91		10	0.5	29.85	**30.36**	29.39	26.19
	10	0.7	28.87	**29.05**	27.15	14.52		10	0.7	29.42	**29.80**	28.08	14.12
	10	0.9	**28.34**	28.24	25.93	12.09		10	0.9	28.93	**29.02**	26.96	12.54
	20	0.1	28.99	28.61	**29.08**	29.07		20	0.1	**29.26**	28.91	29.23	29.09
	20	0.3	**28.35**	28.23	28.27	27.00		20	0.3	**28.65**	28.60	28.54	27.17
	20	0.5	**27.87**	27.75	27.19	23.48		20	0.5	**28.30**	28.17	27.63	23.72
	20	0.7	**27.34**	26.97	26.10	11.62		20	0.7	**27.84**	27.78	26.66	12.05
	20	0.9	**26.77**	26.33	25.10	9.57		20	0.9	**27.31**	27.09	25.73	10.84
	50	0.1	24.19	21.63	**25.17**	24.39		50	0.1	24.40	22.08	**25.81**	24.96
	50	0.3	24.01	17.84	**24.33**	21.91		50	0.3	24.37	18.50	**24.94**	22.50
	50	0.5	**23.43**	15.57	23.34	13.23		50	0.5	**23.77**	16.22	23.84	13.63
	50	0.7	**22.70**	14.28	22.17	10.99		50	0.7	**23.13**	15.02	22.73	11.61
	50	0.9	**21.90**	13.38	21.09	5.93		50	0.9	**22.25**	14.31	21.64	6.45

Table 5: Average performance (in PSNR) of color image denoising on TID2008 dataset with Gaussian noise plus random-valued impulsive noise.

σ	p_0	LDMM	M+BM3D	σ	p_0	LDMM	M+BM3D
0	0.1	**30.22**	29.41	20	0.1	**27.77**	27.73
0	0.3	**29.89**	27.99	20	0.3	26.63	**27.09**
0	0.5	**29.03**	25.71	20	0.5	25.66	**25.83**
0	0.7	**27.95**	23.53	20	0.7	**24.62**	24.23
0	0.9	**26.57**	21.80	20	0.9	**23.33**	22.71
10	0.1	**28.91**	28.79	50	0.1	25.33	**25.55**
10	0.3	**27.69**	28.06	50	0.3	24.04	**24.75**
10	0.5	**26.69**	26.39	50	0.5	22.82	**23.67**
10	0.7	**25.62**	24.35	50	0.7	21.74	**22.57**
10	0.9	**24.13**	22.56	50	0.9	20.63	**21.58**

works pretty well especially with high-density impulsive noise. In the future research, one direction is to improve the performance of LDMM for high Gaussian noise and low-density impulsive noise. We already get some inspirations from the nonlocal Bayesian methods [16]. It seems that constructing anisotropic metric in the patch manifold is a possible direction to improve the performance of LDMM for heavy Gaussion noise.

References

[1] DRK Brownrigg. The weighted median filter. *Communications of the ACM*, 27(8):807–818, 1984. 1

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

[3] Emmanuel J. Candes, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), 2009. 2

[4] Tao Chen, Kai-Kuang Ma, and Li-Hui Chen. Tri-state median filter for image denoising. *IEEE Transactions on Image processing*, 8(12):1834–1838, 1999. 1

[5] Tao Chen and Hong Ren Wu. Adaptive impulse detection using center-weighted median filters. *IEEE Signal Processing Letters*, 8(1):1–3, 2001. 1

[6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16:2080–2095, 2007. 1, 2

[7] Igor Djurović. BM3D filter in salt-and-pepper noise removal. *EURASIP Journal on Image and Video Processing*, 2016(1):13, 2016. 10

[8] W. Dong, G. Shi, and X. Li. Nonlocal image restoration with bilateral variance estimation: a low-rank approach. *IEEE Transactions on Image Processing*, 22(2):700, 2013. 2

[9] Yiqiu Dong and Shufang Xu. A new directional weighted median filter for removal of random-valued impulse noise. *IEEE Signal Processing Letters*, 14(3):193–196, 2007. 1

[10] T. Goldstein and S. Osher. The split bregman method for l1-regularized problems. *SIAM Journal on Imaging Sciences*, 2:323–343, 2009. 2

[11] Tao Huang, Weisheng Dong, Xuemei Xie, Guangming Shi, and Xiang Bai. Mixed noise removal via laplacian scale mixture modeling and nonlocal low-rank approximation. *IEEE Transactions on Image Processing*, 26(7):3171–3186, 2017. 1

[12] Humor Hwang and Richard A Haddad. Adaptive median filters: new algorithms and results. *IEEE Transactions on Image Processing*, 4(4):499–502, 1995. 1, 8

[13] Hui Ji, Chaoqiang Liu, Zuowei Shen, and Yuhong Xu. Robust video denoising using low rank matrix completion. In *Computer Vision and Pattern Recognition*, pages 1791–1798, 2010. 2

[14] Jielin Jiang, Lei Zhang, and Jian Yang. Mixed noise removal by weighted encoding with sparse nonlocal regularization. *IEEE Transactions on Image Processing*, 23(6):2651–2662, 2014. 1, 2, 8

[15] S-J Ko and Yong Hoon Lee. Center weighted median filters and their applications to image enhancement. *IEEE transactions on circuits and systems*, 38(9):984–993, 1991. 1

[16] M. Lebrun, A. Buades, and J. M. Morel. A nonlocal bayesian image denoising algorithm. *SIAM Journal on Imaging Sciences*, 6(3):1665–1688, 2013. 2, 13

[17] Zhen Li and Zuoqiang Shi. A convergent point integral method for isotropic elliptic equations on point cloud. *SIAM: Multiscale Modeling Simulation*, 14:874–905, 2016. 3

[18] Zhen Li, Zuoqiang Shi, and Jian Sun. Point integral method for solving poisson-type equations on manifolds from point clouds with convergence guarantees. *Communications in Computational Physics*, 22:228–258, 2017. 2, 3

[19] Chih-Hsing Lin, Jia-Shiuan Tsai, and Ching-Te Chiu. Switching bilateral filter with a texture/noise detector for universal noise removal. *IEEE Transactions on Image Processing*, 19(9):2307–2320, 2010. 1

[20] Licheng Liu, Long Chen, CL Philip Chen, Yuan Yan Tang, et al. Weighted joint sparse representation for removing mixed noise in image. *IEEE Transactions on Cybernetics*, 47(3):600–611, 2017. 1

[21] Ezequiel López-Rubio. Restoration of images corrupted by gaussian and uniform impulsive noise. *Pattern Recognition*, 43(5):1835–1846, 2010. 1

[22] Ari Nieminen, Pekka Heinonen, and Yrjo Neuvo. A new class of detail-preserving filters for image processing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):74–90, 1987. 1

[23] Stanley Osher, Zuoqiang Shi, and Wei Zhu. Low dimensional manifold model for image processing. *SIAM Journal on Imaging Sciences*, 10(4):1669–1690, 2017. 1

[24] Shaomin Peng and Lori Lucke. Multi-level adaptive fuzzy filter for mixed noise removal. In *Circuits and Systems, 1995. ISCAS'95., 1995 IEEE International Symposium on*, volume 2, pages 1524–1527. IEEE, 1995. 1

[25] Nemanja I Petrovic and Vladimir Crnojevic. Universal impulse noise filter based on genetic programming. *IEEE Transactions on Image Processing*, 17(7):1109–1120, 2008. 1

[26] G. Peyré. Image processing with non-local spectral bases. *SIAM Multiscale Modeling Simulation*, 7:703–730, 2008. 3

[27] G. Peyré. Manifold models for signals and images. *Computer Vision and Image Understanding*, 113:248–260, 2009. 3

[28] Gouchol Pok, Jyh-Charn Liu, and Attoor Sanju Nair. Selective removal of impulse noise based on homogeneity level information. *IEEE Transactions on Image Processing*, 12(1):85–92, 2003. 1

[29] Z. Shi, W. Zhu, and S. Osher. Low dimensional manifold model in hyperspectral image reconstruction. *arXiv:1605.05652.*, 2016. 1

[30] Zuoqiang Shi, Stanley Osher, and Wei Zhu. Weighted nonlocal laplacian on interpolation from sparse data. *Journal of Scientific Computing*, 73(2):1164–1177, Dec 2017. 10

[31] Zuoqiang Shi, Jian Sun, and Minghao Tian. Harmonic extension on point cloud. *SIAM Multiscale Modeling & Simulation*, 16:215–247, 2018. 10

[32] Tong Sun and Yrjö Neuvo. Detail-preserving median based filters in image processing. *Pattern Recognition Letters*, 15(4):341–347, 1994. 1

[33] Yi Wang, Arthur Szlam, and Gilad Lerman. Robust locally linear analysis with applications to image denoising and blind inpainting. *SIAM Journal on Imaging Sciences*, 6(1):526–562, 2013. 1, 2, 8

[34] Jun Xu, Lei Zhang, Wangmeng Zuo, David Zhang, and Xiangchu Feng. Patch group based nonlocal self-similarity prior learning for image denoising. In *IEEE International Conference on Computer Vision*, pages 244–252, 2015. 2

[35] Ming Yan. Restoration of images corrupted by impulse noise and mixed gaussian impulse noise using blind inpainting. *SIAM Journal on Imaging Sciences*, 6(3):1227–1245, 2013. 1

[36] Ganzhao Yuan and Bernard Ghanem. $l_0$tv: A new method for image restoration in the presence of impulse noise. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5369–5377, 2015. 1, 8

[37] Wei Zhu, Bao Wang, Richard Barnard, Cory D. Hauck, Frank Jenko, and Stanley Osher. Scientific data interpolation with low dimensional manifold model. *Journal of Computational Physics*, 352:213–245, 2018. 2