Research Article

Dictionary-Based Image Denoising by Fused-Lasso Atom Selection

Ao Li¹ and Hayaru Shouno²

¹ School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China

² Graduate School of Informatics and Engineering, University of Electro-Communications, Tokyo 182-8585, Japan

Correspondence should be addressed to Hayaru Shouno; shouno@uec.ac.jp

Received 7 May 2014; Revised 12 August 2014; Accepted 13 August 2014; Published 28 August 2014

Academic Editor: Carla Roque

Copyright © 2014 A. Li and H. Shouno. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We proposed an efficient image denoising scheme by fused lasso with dictionary learning. The scheme has two important contributions. The first one is that we learned the patch-based adaptive dictionary by principal component analysis (PCA) with clustering the image into many subsets, which can better preserve the local geometric structure. The second one is that we coded the patches in each subset by fused lasso with the clustering learned dictionary and proposed an iterative Split Bregman to solve it rapidly. We present the capabilities with several experiments. The results show that the proposed scheme is competitive to some excellent denoising algorithms.

1. Introduction

As an essential low-level image processing procedure, image denoising is studied extensively, which is also a classical inverse problem. The general observation with additive noise is modeled as:

$$y = x + n, \quad (1)$$

where y is the noisy observation and x and n present the original image and white Gaussian noise, respectively.

With the degradation model in (1), many denoising algorithms are proposed during the past decades. From the earlier spatial and frequency domain filters [1, 2] to the lately developed wavelet and beyond wavelet based shrinkage methods [3–5]. Due to the fact that the wavelet is not enough to present the local structure in images, an effective dictionary-learning algorithm called KSVD was proposed in [6] and achieved the good effect in image denoising task. In this method, a highly patch-based overcompleted dictionary is learned firstly, and then denoising is implemented by solving the sparse coding of each patch with assuming that the patches are sparsely representable under the learned dictionary. Foi et al. proposed the pointwise shape-adaptive DCT to be applied to the patch and its neighborhoods,

which achieved the very sparse representation for noisy image and got the effective denoising results [7]. To lower the complexity, an orthogonal dictionary learning method is proposed in [8], which intends to train a global dictionary by collecting samples from degradation image randomly. Though it achieves good performance in image restoration, some progress can still be made in learning better dictionaries to present the patches accurately. To better present the image, a two-stage denoising method with PCA is proposed in [9], which trained the dictionary by the local neighboring patches.

More recently, a set of approaches with nonlocal techniques (NL) is used for removing noise. The idea of NL can be tracked to [10], where the similar pixels are searched to weighted average the filtered pixel. But the weight in NL is determined only by the intensity distance between patches, so it cannot guarantee to search the patches with similar local geometric structure under the strong noise. Zhang et al. proposed a novel nonlocal mean method with application to MRI denoising [11]. To make full use of nonlocal similarity, Mairal et al. proposed the nonlocal sparse model for image restoration [12]. Also motivated by the NL, a collaborative denoising scheme called BM3D is proposed by Dabov et al. [13], where they took the patch matching to search the similar

patches and grouped them into a 3D cube matrix. Then, the algorithm carried out the 3D sparse transform, such as the 3D Wavelet or 3D Curvelet, to the cube matrix and then removed the noise with Wiener filtering in the transform domain. Another effective method addressed the denoising problem under the kernel regression (KR) framework proposed by Takeda et al. [14]. Many spatial classical denoising algorithms, such as the bilateral filtering [15] and nonlocal mean [16, 17], can be seen as the special cases in KR with different constraints.

In this paper, we proposed a novel scheme for image denoising based on clustering dictionary learning. Firstly, we clustered the patches with similar geometric structure by taking a weight function as the feature. Secondly, we learned the patch-based dictionary by principle component analysis for each cluster. Lastly, we coded these patches by fused lasso and developed an iterative Split Bregman to solve it rapidly.

The rest of the paper is outlined as follows. In Section 2, we briefly review the kernel regression framework as we want to choose the weight function in KR as the feature for clustering. And then, we talk about how to learn a suitable dictionary to better describe the patches in each cluster in Section 3. An iterative Split Bregman to solve the fused lasso is proposed in Section 4, which is used to code the patches under the dictionary learned in Section 3 rapidly. Section 5 shows several experiments results compared to the current excellent algorithms and Section 6 concludes the paper with a summary.

2. Clustering with Weight Function in KR

Kernel regression is well studied in statistical signal processing. Recently, the KR is used to address many image restoration tasks, such as the denoising, interpolation, and deblurring [18]. The kernel regression expression can be written as follows:

$$\hat{x}_i = \frac{\sum_{j=1}^N K_h(p_i - p_j) y_j}{\sum_{j=1}^N K_h(p_i - p_j)}, \quad K_h(p_i - p_j) = \frac{1}{h} K\left(\frac{p_i - p_j}{h}\right), \quad (2)$$

where N is the number of pixels in neighborhood. x_i presents the i th pixel in image, whose location is denoted by p_i . $K(\cdot)$ is the local kernel to measure the distance between the center pixel and its neighborhoods, and h is the smoothing parameter to control the penalization strength.

By (2), we can obtain that the key technique in KR was how to determine a effective form of kernel function, which was studied in literatures [19, 20]. Among these methods, the steering kernel is distinguished by producing the local regression weights, where local gradient was taken into consideration to analyze the similarity between the pixels in a neighborhood. The weights in steering kernel can be expressed as:

$$\omega_i^j = \frac{\sqrt{\det(C_j)}}{2\pi h^2 \mu_j^2} \exp \left\{ -\frac{(p_i - p_j)^T C_j (p_i - p_j)}{2h^2 \mu_j^2} \right\}, \quad (3)$$

where ω_i^j denotes the structure similarity between the i th center pixel with the j th pixel in neighborhood. C_j is the covariance matrix with the gradient of the j th pixel. Furthermore, the whole kernel consists of all weights in neighborhood.

Introduced in [14], the ω_i^j can present the underlying local structure of patch centered in i th pixel. In addition, Takeda et al. pointed out that the different locations patched with different intensities but similar underlying structure will still produce the similar kernel. Generally, the clustering is implemented by the Euclidean measurement of intensity, such as the denoising algorithm in NLM. But, distinguished from the regression algorithm, what we want here is to learn a dictionary to describe the patches with similar geometric structure. That is, we do not require them to have similar intensity simultaneously. So, we can take ω_i^j as the feature to measure the structure similarity among the patches. The significant distinction between the general Euclidean measurement and KR weighted function is that the latter can obtain the patches with similar structure, but not the patches with similar intensity. To this end, we can take the weights $\omega_i = (\omega_i^1, \omega_i^2, \dots, \omega_i^N)$ formed from steering kernel as the feature and it will show some advantage on learning the clustering dictionary to better describe the local structure in each cluster. Also, the l_1 norm can be used to measure the distance between the features, which show the anisotropy property. Next, we will talk about how to obtain the weights of each patch.

Conveniently, the matrix C_j is divided into three components [14] and can be reformulated as follows:

$$C_j = \gamma_j U_{\theta_j} \Lambda_j U_{\theta_j}^T, \quad (4)$$

where $U_{\theta_j} = \begin{bmatrix} \cos \theta_j & \sin \theta_j \\ -\sin \theta_j & \cos \theta_j \end{bmatrix}$ and $\Lambda_j = \begin{bmatrix} \sigma_j & 0 \\ 0 & \sigma_j^{-1} \end{bmatrix}$.

By (4), what we need is to determine the two variables θ_j and σ_j . To this end, we calculated the local gradient matrix G_j of j th pixel as follows:

$$G_j = \begin{bmatrix} \vdots & \vdots \\ g_h(p_k) & g_v(p_k) \\ \vdots & \vdots \end{bmatrix}, \quad p_k \in W_j \quad (k = 1, 2, \dots, M), \quad (5)$$

where $g_h(\cdot)$ and $g_v(\cdot)$ present the horizontal and vertical gradient operator, respectively; W_j is an analysis window around the j th pixel and M is the number of pixels in the window. With the singular value decomposition (SVD), we can obtain

$$G_j = U_j S_j V_j^T. \quad (6)$$

With $V_j = (v_1, v_2)^T$ and $S_j = \text{diag}\{s_1, s_2\}$, we can calculate the parameters in (4) as:

$$\theta_j = \arctan\left(\frac{v_1}{v_2}\right), \quad \sigma_j = \frac{s_1 + \eta}{s_2 + \eta}, \quad \gamma_j = \sqrt{\frac{s_1 s_2 + \lambda}{M}}, \quad (7)$$

where η is the regulation parameter, which is used to prevent the ratio from being degenerate. λ is used to keep γ_j from being zero.

We summarized the calculation of weights of each patch in Algorithm 1.

Then, we can cluster the ordering overlapping patches into subset Ω_k (k is the cluster indicator) by K -Means with l_1 norm to measure the similarity between the samples and clustering centers.

3. Adaptive PCA Dictionary Learning

Once the clusters are formed, we can learn a dictionary with the principal component analysis suited to each cluster independently. To this end, according to the general dictionary learning algorithm, we need to solve the following minimization:

$$(\widehat{D}_k, \widehat{A}_k) = \underset{D_k, A_k}{\operatorname{argmin}} \left\{ \|\bar{Y}_k - D_k A_k\|_F^2 \right\}, \quad (8)$$

where $\bar{Y}_k = \{\bar{y}_i, i \in \Omega_k\}$ is the centered samples matrix which satisfied $\bar{Y}_k = Y_k - \bar{y}^{(k)}$. Y_k is the samples matrix, whose columns are the patches in the k th cluster, and also we can present it as $Y_k = \{y_i, i \in \Omega_k\}$. $\bar{y}^{(k)}$ is the mean vector of Y_k and $\|\cdot\|_F$ denotes the Frobenius norm.

As the minimization in (8), the numerical method of alternate minimization is used to estimate the two variables. That is to say, we should estimate one variable while the other is fixed. Separately, we rewrite (8) as follows:

$$(\widehat{D}_k, \widehat{A}_k) = \underset{D_k, A_k}{\operatorname{argmin}} \left\{ \sum_{i=1}^{\#\Omega_k} \|\bar{y}_i - D_k a_k^i\|_2^2 \right\}, \quad (9)$$

where a_k^i is the i th column of A_k . When the dictionary is fixed, the expression of a_k^i is given by

$$a_k^i = (D_k^T D_k)^{-1} D_k^T \bar{y}_i. \quad (10)$$

Then, assuming the A_k is fixed and taking formulation (10) into (9), we can obtain

$$\widehat{D}_k = \underset{D_k, A_k}{\operatorname{argmin}} \left\{ \sum_{i=1}^{\#\Omega_k} \|\bar{y}_i - D_k (D_k^T D_k)^{-1} D_k^T \bar{y}_i\|_2^2 \right\}. \quad (11)$$

The patches in the same cluster have similar structure to each other, so we do not require the dictionary to be redundant enough. So, to simplify the problem (11), we add the constraint $D_k^T D_k = I$ to dictionary and formulation (11) can be changed to

$$\min_{D_k} \left\{ \sum_{i=1}^{\#\Omega_k} \|\bar{y}_i - D_k D_k^T \bar{y}_i\|_2^2 \right\}, \quad \text{s.t. } D_k^T D_k = I. \quad (12)$$

The minimization problem in (12) can be approximated by finding the first $m^{(k)}$ principle components of centered matrix $\bar{Y}_k$ with PCA framework. And $m^{(k)}$ satisfied

$$m^{(k)} = \max(m), \quad \sum_{j=m+1}^{\#\Omega_k} s_j^2 \geq r N \sigma^2, \quad (13)$$

where N is the number of pixels in each patch, r is a constant, and σ is the noise standard deviation. s_j is the singular value of $\bar{Y}_k$ and satisfied $s_j \geq s_t \geq 0$, if $j < t$. The reason why we take (13) to choose $m^{(k)}$ is that it can discard the principle components that present the variance arisen by noise.

Our above learning method can train the dictionary with lower complexity. In additional, to make it more effective and compact, we also show a selection scenario in (13), which can tradeoff between the essential signal preservation and noise reduction.

We summarized the dictionary learning with PCA in Algorithm 2.

4. Patch-Based Coding by Fused Lasso

With the preparation work in Sections 2 and 3, in this paragraph, we will study the sparse coding with fused lasso [21]. The reason why we take the fused lasso for sparse coding is that it not only constrains the sparsity of coefficients but also enforced the differences between the neighboring coefficients, which led to show some advantage in recovering the texture in image [22, 23]. So, we can recover the noisy image by minimizing the cost function of fused lasso, which is expressed as follows:

$$f(\beta) = \frac{1}{2} \|\mathbf{y} - D\beta\|_2^2 + \lambda \sum_{i=1}^p |\beta_i| + \eta \sum_{i=2}^n |\beta_i - \beta_{i-1}|, \quad (14)$$

where $\mathbf{y}$ is the sample patch in image denoising task, D can be seen as the dictionary, $\beta = (\beta_1, \beta_2, \dots, \beta_n)$ is the sparse coding of $\mathbf{y}$, and λ and η are regularization parameters.

The reference [21] points out that solving fused lasso is computationally demanding. So we developed a novel scenario to solve the fused-lasso problem in (14) based on Split Bregman, which is originally proposed in [24] and used to solve the l_1 minimization successfully [25, 26]. Conveniently, we rewrite (14) as follows:

$$\min_{\beta} \left\{ \frac{1}{2} \|\mathbf{y} - D\beta\|_2^2 + \lambda \|\beta\|_1 + \eta \|L\beta\|_1 \right\}, \quad (15)$$

where L is a matrix, whose each row only contains two nonzero values -1 and 1 in the positions corresponding to the third term in (14). L is given by

$$L = \begin{bmatrix} -1 & 1 & 0 & \dots \\ 0 & -1 & 1 & \dots \\ \vdots & \vdots & \vdots & \dots \\ \dots & \dots & -1 & 1 \end{bmatrix}. \quad (16)$$

To solve (15), we introduce the idea with two auxiliary variables based on Split Bregman, which was proposed in our previous work in [27], and change (15) as the following form:

$$\begin{aligned} \min_{\beta} \left\{ \frac{1}{2} \|\mathbf{y} - D\beta\|_2^2 + \lambda \|u\|_1 + \eta \|v\|_1 \right\} \\ \text{s.t. } u = \beta, \quad v = L\beta. \end{aligned} \quad (17)$$

Initialization: patch y_i centered in i th pixel, N is the number of pixels in y_i , $\eta = 1$, $\lambda = 0.01$, $\mu_i = 1$, $K = 1$;
 (1) Calculate the local gradient matrix G_K of K th pixel in y_i ;
 (2) Implement the SVD to G_K and computed the parameters by (7);
 (3) Compute the covariance matrix C_K by (4);
 (4) Compute the ω_i^K by (3);
 (5) If $K < N$, $K = K + 1$; otherwise, output the feature vector $\omega_i = (\omega_i^1, \omega_i^2, \dots, \omega_i^N)$;

ALGORITHM 1

Input: Centered samples matrix $\bar{Y}_k$ ($\# \Omega_k$ presents the length of $\bar{Y}_k$);
 Output: Dictionary D_k for the cluster Ω_k ;
 (1) Compute the decomposition of $1/\# \Omega_k \bar{Y}_k \bar{Y}_k^T : 1/\# \Omega_k \bar{Y}_k \bar{Y}_k^T = U_k \Lambda_k U_k^T$;
 (2) Choose the $m^{(k)}$ by (13) with $\Lambda_k \left(s_j = \sqrt{\Lambda_k(j, j)} \right)$;
 (3) Output the $D_k = (u_1^k, u_2^k, \dots, u_{m^{(k)}}^k)$, u_j^k is the j th column of matrix U_k ;

ALGORITHM 2

(a) Lena

(b) House

(c) Couple

(d) Man

FIGURE 1: Test images.

Initialization: $\beta^0 = D^T \mathbf{y}$, $u^0 = v^0 = p^0 = q^0 = 0$, $\eta = \lambda = 0.2$, $\mu_1 = \mu_2 = 0.1$;
 For $k = 0 : \text{MaxIternum}$
 (1) Compute the β^{k+1} by (25);
 (2) Compute the u^{k+1} by (26);
 (3) Compute the v^{k+1} by (27);
 (4) Update the p^{k+1} and q^{k+1} by (20) and (21);
 End
 Output: the recovered patch $\hat{\mathbf{x}} = D\hat{\beta}$.

ALGORITHM 3

Note that the minimization with the augmentation Lagrangian function of (17) can be expressed as follows:

$$\min_{\beta, u, v} \left\{ \frac{1}{2} \|\mathbf{y} - D\beta\|_2^2 + \lambda \|u\|_1 + \eta \|v\|_1 + \langle p, \beta - u \rangle + \langle q, L\beta - v \rangle + \frac{\mu_1}{2} \|\beta - u\|_2^2 + \frac{\mu_2}{2} \|L\beta - v\|_2^2 \right\}. \quad (18)$$

With the alternating direction method [28] (ADM), we can solve (18) with the following iterative algorithm:

$$\begin{aligned} & (\beta^{k+1}, u^{k+1}, v^{k+1}) \\ &= \operatorname{argmin}_{\beta} \left\{ \frac{1}{2} \|\mathbf{y} - D\beta\|_2^2 + \lambda \|u\|_1 + \eta \|v\|_1 + \langle p^k, \beta - u \rangle \right. \\ & \quad \left. + \langle q^k, L\beta - v \rangle + \frac{\mu_1}{2} \|\beta - u\|_2^2 + \frac{\mu_2}{2} \|L\beta - v\|_2^2 \right\}, \end{aligned} \quad (19)$$

$$p^{k+1} = p^k + \mu_1 (\beta^{k+1} - u^{k+1}), \quad (20)$$

$$q^{k+1} = q^k + \mu_2 (L\beta^{k+1} - v^{k+1}). \quad (21)$$

Note that the initial idea to get the solution of (14) is presented in [21], which is called coordinate descent method (CDM). But, in practice, CDM is slow, complex, and difficult to implement to some specialized algorithms. In addition, it is not effective in some large-scale processing problems.

Furthermore, by the iterative Split Bregman, we can divide (19) into the following three independent minimizations, whose convergence is guaranteed by [29]:

$$\begin{aligned} \beta^{k+1} = \operatorname{argmin}_{\beta} \left\{ \frac{1}{2} \|\mathbf{y} - D\beta\|_2^2 + \frac{\mu_1}{2} \|\beta - u^k + \frac{1}{\mu_1} p^k\|_2^2 \right. \\ \left. + \frac{\mu_2}{2} \|L\beta - v^k + \frac{1}{\mu_2} q^k\|_2^2 \right\}, \end{aligned} \quad (22)$$

$$u^{k+1} = \operatorname{argmin}_u \left\{ \frac{\mu_1}{2} \left\| u - \left(\beta^{k+1} + \frac{1}{\mu_1} p^k \right) \right\|_2^2 + \lambda \|u\|_1 \right\}, \quad (23)$$

$$v^{k+1} = \operatorname{argmin}_v \left\{ \frac{\mu_2}{2} \left\| v - \left(L\beta^{k+1} + \frac{1}{\mu_2} q^k \right) \right\|_2^2 + \eta \|v\|_1 \right\}. \quad (24)$$

As to the quadratic minimization in (22), we can obtain the solution by solving the following linear equation:

$$\begin{aligned} (D^T D + \mu_1 I + \mu_2 L^T L) \beta^{k+1} = D^T \mathbf{y} + \mu_1 (u^k - \mu_1^{-1} p^k) \\ + \mu_2 L^T (v^k - \mu_2^{-1} q^k), \end{aligned} \quad (25)$$

where I is the identity matrix with the same size to $L^T L$.

While the l_1 minimization in (23) and (24) can be solved by the shrinkage technique. So we can obtain the u^{k+1} and v^{k+1} by

$$u^{k+1} = S_{\mu_1^{-1} \lambda} (\beta^{k+1} + \mu_1^{-1} p^k), \quad (26)$$

$$v^{k+1} = S_{\mu_2^{-1} \eta} (L\beta^{k+1} + \mu_2^{-1} q^k), \quad (27)$$

where $S_{\lambda}(\mathbf{x}) = (t_{\lambda}(\mathbf{x}(1)), t_{\lambda}(\mathbf{x}(2)), \dots)$ and $t_{\lambda}(\omega) = \operatorname{sgn}(\omega) \max(0, |\omega| - \lambda)$.

We summarized the coding algorithm in Algorithm 3. Now, with all the preparation works above, we can summarize the synthetic denoising algorithm in Algorithm 4.

5. Experiment Results and Analysis

We conducted various experiments on image denoising to demonstrate the performance of our proposed algorithm. We degrade the images by adding artificial zero mean Gaussian noise with different standard deviations. The test images are shown in Figure 1. The sizes of images in experiment are all 256×256 , and the patch size is 8×8 ; that is, N is 64 in Algorithm 1. Empirically, we set the parameters $r = 1.0$ and $h = 2.4$ for all the experiment images in Algorithm 1. We fixed the window size in (5) to be 9×9 , where we found the small size, such as 5×5 , may not capture the local geometric structure of underlying image data sometimes. Surely, the window size can be tuned according to concrete experiment requirement. In particular, we extend the image boundary with the “symmetric” type according to the window size. The clustering number in K -Means is flexible, which is small in image with compact structure such as House in Figure 1, and large with complex structure such as Lena. According to the image underlying structure, in our experiments, the clustering number K is optional between 5 to 10. The maximum iteration T in Algorithm 4

(a) TV

(b) BM3DW

(c) KSVD

(d) KR

(e) TSPCA

(f) Proposed

FIGURE 2: Denoising results of Lena.

is 10. In addition, the regularization parameters are set in the algorithms themselves.

We compared our proposed algorithm to several current excellent denoising approaches, including the FGTV method

in [30], denoising with the dictionary learned by KSVD in [6] (DKSVD), BM3D with Wavelet (BM3DW) in [13], the kernel regression method in [14], and the two-stage denoising method with PCA (TSPCA) in [9]. Due to the limited space,

(a) TV

(b) BM3DW

(c) KSVD

(d) KR

(e) TSPCA

(f) Proposed

FIGURE 3: Denoising results of Couple.

we only show the experiment results of Lena and Couple with noise standard deviation $\sigma = 20$ in Figures 2 and 3, respectively. Furthermore, the PSNR results of all the recovered images are reported in Figure 4.

From the denoising results, we can note that the FGTV method has the worst visualization among the compared methods. Because it only constrains the total variation but does not consider the local structure adaptively, so it lost

FIGURE 4: PSNR results of test images.

many details in original image and also showed the disadvantage in PSNR results. KR algorithm generates many mottled artifacts in denoised image, but it indeed preserves some texture by capturing the local structure with kernel function, such as the curtain in Couple. But its results declined rapidly with the increasing noise standard deviation both in visual quality and in PSNR. TSPCA generated specific smoothness in the recovered image and was weak to present the texture region. In addition, the other factor resulting in bad performance in texture is that the PCA dictionary is learned with neighboring patches, which will show weaker results than the nonlocal scheme. We can see

the PSNR of BM3DW, DKSVD, and the proposed method are approximate to each other in Figure 4, although the BM3DW obtained the highest PSNR compared to other methods. But BM3DW and DKSVD have more distortions in part of texture regions than the proposed method. This is because in BM3DW, the wavelet is not a good representation for all types of images, such as the ones with some complex structures. Also, although the KSVD can obtain a dictionary learned by image itself to better present the structures in image, it produces a universal dictionary, which may be not effective to certain local structures. Compared to BM3DW and KSVD, the proposed method showed the advantage to present the

Input: A set of overlapping patches from noisy image $\bar{Y} = \{\bar{y}_i\}_{i=1}^P$;
Output: The recovered image $\hat{X}$;
(1) For $t = 1:T$ do
(2) Compute the weights vector ω_i for $\bar{y}_i$ by Algorithm 1;
(3) Cluster the patches into Ω_k ($k = 1, 2, \dots, K$) by K-Means with l_1 norm measurement between ω_i and the cluster center;
(4) For $k = 1:K$ do
(5) Learn the dictionary D_k for cluster Ω_k by Algorithm 2;
(6) Recover all the patches in Ω_k by Algorithm 3;
(7) End
(8) Compute the $\hat{X}$ by averaging the recovered patches according to their indexes. Then, update $\bar{Y}$ by the recovered image $\hat{X}$.
(9) End

ALGORITHM 4

underlying local geometric structures in image, such as the hairs in Lena and the face of woman in Couple.

6. Conclusion

In this paper, we propose a novel scheme for image denoising. To preserve the textures in image, we clustered the patches from the noisy image with the meaningful weights vector which can capture the underlying local structure. And then, we learned the dictionary to better present the patches for each cluster with PCA. Lastly, we coded the noisy patches with the learned dictionary by fused lasso and obtained recovery image. We compared our proposed scheme to some current excellent algorithms, and it can be seen that our method obtains the good performance both in visual quality and in PSNR among the compared methods. In addition, the dictionary learning and coding are performed independently in each cluster, so it can be easily parallelized by the processors with multiple cores when the image has been clustered. That means the proposed method can be used in the large-scale image denoising task and save more computational time effectively.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] X. Jiangtao, W. Lei, and S. Zaifeng, "A switching weighted vector median filter based on edge detection," *Signal Processing*, vol. 98, pp. 359–369, 2014.
- [2] D. L. Lau and J. G. Gonzalez, "Closest-to-mean filter: an edge preserving smoother for Gaussian environments," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, vol. 4, pp. 2593–2596, April 1997.
- [3] J. Starck, E. J. Candes, and D. L. Donoho, "The curvelet transform for image denoising," *IEEE Transactions on Image Processing*, vol. 11, no. 6, pp. 670–684, 2002.
- [4] G. Y. Chen and B. Kégl, "Image denoising with complex ridgelets," *Pattern Recognition*, vol. 40, no. 2, pp. 578–585, 2007.
- [5] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.
- [6] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on Signal Processing*, vol. 54, no. 11, pp. 4311–4322, 2006.
- [7] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1395–1411, 2007.
- [8] C. Bao, J. Cai, and H. Ji, "Fast sparsity -based orthogonal dictionary learning for image restoration," in *IEEE International Conference on Computer Vision*, pp. 3384–3319, 2013.
- [9] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal component analysis with local pixel grouping," *Pattern Recognition*, vol. 43, no. 4, pp. 1531–1549, 2010.
- [10] L. Yaroslavsky, *Digital Picture Processing: An Introduction*, Springer, Berlin, Germany, 1985.
- [11] X. Zhang, G. Hou, J. Ma et al., "Denoising MR images using non-local means filter with combined patch and pixel similarity," *Plos ONE*, vol. 9, no. 6, pp. 1–12, 2014.
- [12] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proceedings of the 12th International Conference on Computer Vision (ICCV '09)*, pp. 2272–2279, October 2009.
- [13] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [14] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Transactions on Image Processing*, vol. 16, no. 2, pp. 349–366, 2007.
- [15] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the IEEE 6th International Conference on Computer Vision*, pp. 839–846, Washington, DC, USA, January 1998.

- [16] A. Buades, B. Coll, and J. M. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 490–530, 2005.
- [17] Y. Zhan, M. Ding, L. Wu, and X. Zhang, "Nonlocal means method using weight refining for despeckling of ultrasound images," *Signal Processing*, vol. 103, pp. 201–213, 2014.
- [18] H. Takeda, S. Farsiu, and P. Milanfar, "Deblurring using regularized locally adaptive kernel regression," *IEEE Transactions on Image Processing*, vol. 17, no. 4, pp. 550–563, 2008.
- [19] P. Yee and S. Haykin, "Pattern classification as an ill-posed, inverse problem: a regularization approach," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '93)*, vol. 1, pp. 597–600, April 1993.
- [20] M. P. Wand and M. C. Jones, *Kernel Smoothing, ser. Monographs on Statistics and Applied Probability*, Chapman and Hall, New York, NY, USA, 1995.
- [21] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society B: Statistical Methodology*, vol. 67, no. 1, pp. 91–108, 2005.
- [22] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani, "Pathwise coordinate optimization," *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.
- [23] H. Gao and H. Zhao, "Multilevel bioluminescence tomography based on radiative transfer equation part 1: l1 regularization," *Optics Express*, vol. 18, no. 3, pp. 1854–1871, 2010.
- [24] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, "An iterative regularization method for total variation-based image restoration," *Multiscale Modeling & Simulation*, vol. 4, no. 2, pp. 460–489, 2005.
- [25] S. Osher, Y. Mao, B. Dong, and W. Yin, "Fast linearized Bregman iteration for compressive sensing and sparse denoising," *Communications in Mathematical Sciences*, vol. 8, no. 1, pp. 93–111, 2010.
- [26] T. Goldstein and S. Osher, "The split Bregman method for L1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.
- [27] A. Li, Y. Li, X. Yang, and Y. Liu, "Image restoration with dual-prior constraint models based on Split Bregman," *Optical Review*, vol. 20, no. 6, pp. 491–495, 2013.
- [28] E. Esser, "Applications of lagrangian-based alternating direction methods and connections to split Bregman," CAM Report, UCLA, 2009.
- [29] J. Eckstein and D. P. Bertsekas, "On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators," *Mathematical Programming*, vol. 55, no. 1, pp. 293–318, 1992.
- [30] A. Beck and M. Teboulle, "Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems," *IEEE Transactions on Image Processing*, vol. 18, no. 11, pp. 2419–2434, 2009.