Journal of Scientific Computing (2019) 78:607–624
https://doi.org/10.1007/s10915-018-0785-8

Block Matching Local SVD Operator Based Sparsity and TV Regularization for Image Denoising

Jun Liu[1] · Stanley Osher[2]

Received: 21 June 2017 / Revised: 20 December 2017 / Accepted: 9 July 2018 / Published online: 21 July 2018

Abstract

We propose a denoising method by integrating group sparsity and TV regularization based on self-similarity of the image blocks. By using the block matching technique, we introduce some local SVD operators to get a good sparsity representation for the groups of the image blocks. The sparsity regularization and TV are unified in a variational problem and each of the subproblems can be efficiently optimized by splitting schemes. The proposed algorithm mainly contains the following four steps: block matching, basis vectors updating, sparsity regularization and TV smoothing. The self-similarity information of the image is assembled by the block matching step. By concatenating all columns of the similar image block together, we get redundancy matrices whose column vectors are highly correlated and should have sparse coefficients after a proper transformation. In contrast with many transformation based denoising methods such as BM3D with fixed basis vectors, we update local basis vectors derived from the SVD to enforce the sparsity representation. This step is equivalent to a dictionary learning procedure. With the sparsity regularization step, one can remove the noise efficiently and keep the texture well. The TV regularization step can help us to reduced the artifacts caused by the image block stacking. Besides, we mathematically show the convergence of the algorithms when the proposed model is convex (with $p = 1$) and the bases are fixed. This implies the iteration adopted in BM3D is converged, which was not mathematically shown in the BM3D method. Numerical experiments show that the proposed method is very competitive and outperforms state-of-the-art denoising methods such as BM3D.

Keywords Block matching · TV regularization · Sparsity regularization · Nuclear norm · SVD · Image restoration

✉ Jun Liu
jliu@bnu.edu.cn

Stanley Osher
sjo@math.ucla.edu

[1] Laboratory of Mathematics and Complex Systems, School of Mathematical Sciences, Beijing Normal University, Beijing 100875, People's Republic of China

[2] Department of Mathematics, UCLA, 520 Portola Plaza, Los Angeles, CA 90095, USA

1 Introduction

Image denoising is a fundamental low level computer vision task and has a long history. In this paper, we focus on the classical additive Gaussian white noise removing problem. Mathematically, the observed noisy image can be modeled as $g = f + n$, where f is the latent clean image and n is the Gaussian noise with 0 mean. To restore f from g, thousands of methods have been proposed in the past several decades.

To keep the non-continuity of f, TV was proposed in [1]. The bounded variation space admits piece-wise constant functions and thus TV regularization is very efficient in denoising cartoon images. However, small structures such as textures can not be identified by TV and these repetitive structures can be removed together with the noise. To make a distinction between textures and noise, self-similarity information was introduced to the denoising methods. The nonlocal means method [2] used the image blocks's self-similarity to average the pixels and texture preserving was greatly improved . The nonlocal means method triggered the self-similarity study in recent years and a variety of methods based on different mathematical tools were designed, such as nonlocal TV [3], block nonlocal TV [4] , BM3D [5] and so on.

Sparsity regularization is a hot research topic in the recent years. The strong assumption of the sparsity method is that the image signal can be represented sparsely under some proper basis function. Usually, the basis vectors or functions can be chosen as some well-known orthogonal basis such as FFT, DCT, wavelets, SVD and so on. One of the representative methods for the sparsity regularization is BM3D [5]. In this method, similar image blocks were stacked together into a 3D array to enforce the image information and sparsity. By choosing proper basis vectors, this 3D matrix would have a very sparse representation. Then each image block can be estimated by thresholding the sparse coefficients. BM3D can produce state-of-the-art denoising results due to its nonlocal self-similarity. However, there are two main flaws in this method. The first one is that the basis vectors are fixed and some image blocks might not have enough sparse coefficients. We will design a numerical experiment to show this below. The other is that artificial ringing effects will occur in the restoration due to the image block stacking method. In fact, these phenomena can be reduced by using global TV.

Low-rank methods for image denoising have received much attention in recent years. It is not difficult to observe that a matrix formed by the columns of some nonlocal similar patches in a natural image is low-rank. By integrating the self-similarity property, this method can produce good restorations [6]. In fact, low-rank is a variant of sparsity regularization and is associated with an l_0 minimization problem for a matrix's singular values. In low rank methods, the basis vectors are chosen as the SVD of a matrix. However, the general low-rank problem is non convex and difficult to optimize. A good choice to approximate the low-rank is to use the nuclear norm, which is defined by a sum of the singular values of a matrix and it can be easily solved by a singular values thresholding method [7]. It is well-known that the l_1 norm is the tightest convex relaxation of l_0. Thus in matrix completion, the nuclear norm minimization can be regarded as a convex relaxation of low-rank problem since the nuclear norm is defined by an l_1 norm of a vector which is composed by a matrix's singular values. Many methods such as reweighting [8], truncated nuclear norm [9], weighted nuclear norm [10] and Schatten Norm [11] and other methods [12] have been proposed to enforce the sparsity. However, these methods just enhance the sparsity and do not consider the basis. In fact, a proper basis functions system is very important in a sparse representation [13].

In this paper, we propose a local SVD operators based sparsity and TV regularization method. This method is developed by formulating the local nuclear norm denoising method [6] with a variational problem, which is easily extended to many other problems. In fact, the local image block processing is closely related to the Kronecker product of matrices. By using vectorization, we can get local SVD based operators. These local SVD operators can help us to find good basis functions and get sparse and redundant representations for image blocks. To reduce the artificial ringing caused by the image blocks processing, a global TV regularization is integrated into the cost functional. The proposed minimization problem can be efficiently solved by splitting schemes. Experimental results show that it can provide some impressive denoising results. It has better performance than BM3D in both PSNR and visually, and BM3D is often regarded as a state-of-the-art denoising algorithm.

The main contributions of this paper are as follows:

- We build a variational formulation for the block matching based low rank method. This new formulation can be easily extended to many other image processing problems such as deblurring.
- We introduce the TV to the block matching based method and elegantly integrate it to the cost functional.
- We propose a splitting denoising optimization algorithm and achieve state-of-the-art performance. The convergence of the proposed algorithms can be obtained under some mild conditions.

Let us point out that the proposed method is self-contained and do not need training, which is different from the data driven machine learning type algorithms such as convolutional neural network based techniques [14–16].

The rest of the paper is organized as follows: In Sect. 2, we will briefly introduce related work such as BM3D and some SVD based methods. We present our proposed model in Sect. 3. In Sect. 4, the optimization scheme to solve the proposed model and some of the related details will be described. The convergence results are contained in Sect. 5. Section 6 includes some experimental results and comparisons with related methods. Finally, we will conclude the paper in Sect. 7.

2 Related Work

2.1 Notations

We first give some notation. Throughout this paper, we write matrices as bold letters such as $\boldsymbol{U}, \boldsymbol{V}, \boldsymbol{P}$. The lowercase letters stand for column vectors. Let $f, g \in \mathbb{R}^N$ be images. We sometimes use the lowercase letters f or $vec(\boldsymbol{F})$ to represent a column vector by stacking the columns of a matrix $\boldsymbol{F}$, and the inverse operator of vec is defined as $array(f)$, i.e $array(f) = \boldsymbol{F}$. The superscripts i, j in matrices such as $\boldsymbol{P}^{ij}$ always stand for different matrices. Similarly to [13], $\boldsymbol{P}^{ij}$ is an extract matrix whose elements of each row are zeros except for one with value 1. The symbol $\otimes$ stands for the Kronecker product.

2.2 BM3D and Low Rank

The BM3D method [5] is a famous denoising method. It has become a baseline algorithm to test the performance of denoising algorithms. BM3D includes the following steps: the first step is block matching: for each image block located at j, the similar image blocks with size $\sqrt{n} \times \sqrt{n}$ are collected in groups with member number I_j. Image blocks in each group are stacked together to form a $\sqrt{n} \times \sqrt{n} \times I_j$ 3-D data array. Second, the sparsity regularization step, the 3-D arrays are decorrelated by using an invertible 3D transform such as DCT and then are filtered by thresholding. Finally, the restoration is obtained by aggregating all the estimated image patches. These steps are formulated as a nice variational problem by Danielyan et al. in [17]. In this paper, we will borrow representations from it. Based on the block matching method, in Ji et al. [6] proposed a nuclear norm based method for a video denoising algorithm. The method in [6] decorrelates the redundant information by thresholding the singular values of the SVD. Since there is no a good SVD for an array whose dimension is more than 2 [18], in the SVD based method, a common choice is to create $n \times I_j$ 2-D matrices by concatenating all columns of the patch. i.e.

$$\boldsymbol{M}^j = \left[\boldsymbol{P}^{1j}g, \boldsymbol{P}^{2j}g, \ldots, \boldsymbol{P}^{I_j j}g\right],$$

where $\boldsymbol{P}^{ij} \in \mathbb{R}^{n\times N}, i = 1, 2\cdots, I_j$ are some extract matrices whose elements are binary and $\boldsymbol{P}^{ij}g \in \mathbb{R}^n$ stands for a vectorized i-th most similar image block to the image patch at j. Locally, the restoration problem can be written as

$$\min_{\boldsymbol{X}^j} \left\{\frac{1}{2}||\boldsymbol{M}^j - \boldsymbol{X}^j||_F^2 + \mu_j||\boldsymbol{X}^j||_*\right\}, \tag{1}$$

where $||\cdot||_*$ is the nuclear norm defined as the sum of the singular values of $\boldsymbol{X}^j$ [6] and μ_j is a regularization parameter. In order to extend this method, we will reformulate the problem (1) with basis functions and sparse representations. Let the SVD of $\boldsymbol{M}^j$ be

$$\boldsymbol{M}^j = \boldsymbol{U}^j\Sigma_{\boldsymbol{M}^j}(\boldsymbol{V}^j)^{'},$$

where $\Sigma_{\boldsymbol{M}^j} \in \mathbb{R}^{n\times I_j}$ is a diagonal matrix and $\boldsymbol{U}^j \in \mathbb{R}^{n\times n}$, $\boldsymbol{V}^j \in \mathbb{R}^{I_j\times I_j}$ are orthogonal unitary matrices. If we chose $\boldsymbol{U}^j$ and $\boldsymbol{V}^j$ as bases, then we get for any $\boldsymbol{X}^j \in \mathbb{R}^{n\times I_j}$,

$$\boldsymbol{X}^j = \boldsymbol{U}^j\Sigma_{\boldsymbol{X}^j}(\boldsymbol{V}^j)^{'}.$$

Note that the coefficient matrix $\Sigma_{\boldsymbol{X}^j}$ may not be a diagonal matrix. If we require this representation to be sparse under the condition $||\boldsymbol{M}^j - \boldsymbol{X}^j||_F^2 = c\sigma^2$, then we can get the Lagrangian version of this problem

$$\min_{\boldsymbol{\Sigma}_{\boldsymbol{X}^j}} \left\{\frac{1}{2}||\boldsymbol{\Sigma}_{\boldsymbol{M}^j} - \boldsymbol{\Sigma}_{\boldsymbol{X}^j}||_F^2 + \mu_j||\boldsymbol{\Sigma}_{\boldsymbol{X}^j}||_1\right\}. \tag{2}$$

Here $||\cdot||_1$ represents the “entry-wise” l_1 norm, i.e. $||\boldsymbol{A}||_1 = \sum_{ij}|a_{ij}|$ for any matrix $\boldsymbol{A} = (a_{ij})$.

It is not difficult to show that the problem (1) is equivalent to the sparsity regularization problem (2).

Proposition 1 *Suppose $\boldsymbol{U}^j, \boldsymbol{V}^j$ are the left and right singular matrices of $\boldsymbol{M}^j$, i.e. $(\boldsymbol{U}^j)^{'}\boldsymbol{M}^j\boldsymbol{V}^j = \boldsymbol{\Sigma}_{\boldsymbol{M}^j}$, then we have that $\boldsymbol{\Sigma}^*_{\boldsymbol{X}^j}$ is a minimizer of (2) if and only if $(\boldsymbol{X}^j)^* = \boldsymbol{U}^j\boldsymbol{\Sigma}^*_{\boldsymbol{X}^j}(\boldsymbol{V}^j)^{'}$ is a minimizer of (1).*

Proof To simplify notations, we omit the superscript and subscript j for all the matrices and variables in (1), (2) and Proposition 1. Let us suppose $\boldsymbol{\Sigma}_{\boldsymbol{M}} = diag(\{\sigma_i\}_{1\leqslant i\leqslant\min(n,I)})$. Then the minimizer $\boldsymbol{X}^*$ of (1) is related to a singular value shrinkage operator according to [7], i.e. $\boldsymbol{X}^* = \boldsymbol{U}diag(\{\max(0, \sigma_i - \mu)\})\boldsymbol{V}^{'}$. We just need to prove $diag(\{\max(0, \sigma_i - \mu)\})$ is a

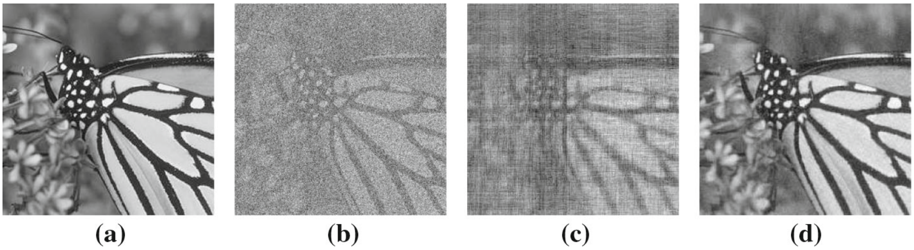

Fig. 1 Denoising results with different basis functions. **a** Clean, **b** Noisy, 8.10 dB, **c** 16.26 dB, **d** 27.01 dB

minimizer of (2). Denote $\mathbf{\Sigma}_{\boldsymbol{M}} = ((\sigma_M)_{ij})_{n\times I}$, $\mathbf{\Sigma}_{\boldsymbol{X}} = ((\sigma_X)_{ij})_{n\times I}$, then $(\sigma_M)_{ij} = \delta_{ij}\sigma_i$, and problem (2) becomes a $l_1 - l_2$ minimization problem

$$\min_{(\sigma_X)_{ij}} \left\{ \sum_{i=1}^{n} \sum_{j=1}^{I} \left[\frac{1}{2}((\sigma_X)_{ij} - (\sigma_M)_{ij})^2 + \mu |(\sigma_X)_{ij}| \right] \right\}.$$

It is well-known that the above problem has a close-formed minimizer $(\sigma_X^*)_{ij} = \frac{(\sigma_M)_{ij}}{|(\sigma_M)_{ij}|} \max(0, |(\sigma_M)_{ij}| - \mu)$. Please note $(\sigma_M)_{ij} = \delta_{ij}\sigma_i \geqslant 0$ and $\mu > 0$, then we have $(\sigma_X^*)_{ij} = max(0, (\sigma_M)_{ij} - \mu) = \delta_{ij} max(0, \sigma_i - \mu)$. Thus $\mathbf{\Sigma}_{\boldsymbol{X}}^* = ((\sigma_X^*)_{ij})_{n\times I} = diag(\{\max(0, \sigma_i - \mu)\})$ is a minimizer of problem (2). Please recall that these two problems (1) and (2) are strictly convex and both of them have a unique minimizer, and thus $\boldsymbol{X}^*$, $\mathbf{\Sigma}_{\boldsymbol{X}}^*$ are their unique minimizers, respectively. □

From the above analysis, we can see that the nuclear norm minimization problem is closely related to an l_1 minimization in a transformed domain with the transformation basis functions $\boldsymbol{U}^j$ and $\boldsymbol{V}^j$. However, we will show that the basis functions $\boldsymbol{U}^j$, $\boldsymbol{V}^j$ are not good enough to ensure that $\boldsymbol{X}^j$ has a sparse representation. For simple comparison, we let $\boldsymbol{M}^j$ be a noisy image displayed in Fig. 1b, the restoration produced by the solution of the problem (2) with $\mu_j = 2230$ are showed in Fig. 1c. We test several μ_j and choose the result with the highest PSNR 16.26 dB for comparison. Next, we change the basis functions $\boldsymbol{U}^j$, $\boldsymbol{V}^j$ as $\hat{\boldsymbol{U}}^j$, $\hat{\boldsymbol{V}}^j$ and $\hat{\boldsymbol{U}}^j$, $\hat{\boldsymbol{V}}^j$ are set as the singular matrices of the clean image displayed in Fig. 1a. i.e. $\boldsymbol{F} = \hat{\boldsymbol{U}}^j \Sigma_{\boldsymbol{F}} (\hat{\boldsymbol{V}}^j)'$. Then we calculate $\mathbf{\Sigma}_{\boldsymbol{M}^j} = (\hat{\boldsymbol{U}}^j)' \boldsymbol{M}^j \hat{\boldsymbol{V}}^j$ in problem (2), we solve this problem again with $\mu_j = 260$, and get the result demonstrated in Fig. 1d. We see that there is much improvement in both PSNR and visual effect in Fig. 1d. The reason is very simple: we use better basis vectors for the transformation. In the denoising problem, the $\hat{\boldsymbol{U}}^j$, $\hat{\boldsymbol{V}}^j$ are not available since the latent clean image $\boldsymbol{F}$ is unknown. However, this inspires us to update the basis function by using an iteration method. We will formulated a local SVD operator based method in the next section.

3 Proposed Model

Our method is mainly based on (2). First, We reformulate (2) as a linear operator representation. For an image $g \in \mathbb{R}^N$, please recall $\boldsymbol{M}^j = [\boldsymbol{P}^{1j} g, \boldsymbol{P}^{2j} g, \ldots, \boldsymbol{P}^{I_j j} g]$, and

$\boldsymbol{\Sigma}_{\boldsymbol{M}^j} = (\boldsymbol{U}^j)'\boldsymbol{M}^j\boldsymbol{V}^j$. Let $(\boldsymbol{V}^j)' = [v_1^j, v_2^j, \ldots, v_{I_j}^j]$, then

$$\begin{aligned}\boldsymbol{\Sigma}_{M^j} &= (\boldsymbol{U}^j)'\left[\boldsymbol{P}^{1j}g, \boldsymbol{P}^{2j}g, \ldots, \boldsymbol{P}^{I_jj}g\right]\left[v_1^j, v_2^j, \ldots, v_{I_j}^j\right]' \\ &= \sum_{i=1}^{I_j}(\boldsymbol{U}^j)'\boldsymbol{P}^{ij}g(v_i^j)'.\end{aligned}$$

and obtain

$$vec(\boldsymbol{\Sigma}_{\boldsymbol{M}^j}) = \sum_{i=1}^{I_j}\left(v_i^j \otimes \left((\boldsymbol{U}^j)'\boldsymbol{P}^{ij}\right)\right)g.$$

We denote the local SVD operator as

$$\boldsymbol{T}^j = \sum_{i=1}^{I_j} v_i^j \otimes ((\boldsymbol{U}^j)'\boldsymbol{P}^{ij}), \tag{3}$$

thus we have

$$vec(\boldsymbol{\Sigma}_{\boldsymbol{M}^j}) = \boldsymbol{T}^j g.$$

Based on the above analysis, we propose the following general model for denoising:

$$\min_f \left\{\frac{1}{2}\sum_{j=1}^{J}||\boldsymbol{T}^j(f-g)||_2^2 + \sum_{j=1}^{J}\mu_j||\boldsymbol{T}^j f||_p + \mu TV(f)\right\}, \tag{4}$$

where $\boldsymbol{T}^j$ is a block matching local SVD operator defined in (3). $j \in \{1, 2, \ldots, J\}$ are locations of pixels. $|| \cdot ||_p$ is the p norm in which p can be chosen as 0 (low rank) or 1 (nuclear norm). TV is the discrete isotropic TV operator which has the following discrete expression

$$TV(f) = ||\nabla f||_{2,1},$$

where $\nabla f = ((\boldsymbol{I} \otimes \boldsymbol{D}^1)f, (\boldsymbol{D}^2 \otimes \boldsymbol{I})f)$ and $\boldsymbol{D}^1$, $\boldsymbol{D}^2$ are two 1D difference matrices with respect to x-direction and y-direction. While $|| \cdot ||_{2,1}$ has the precise representation $||\boldsymbol{A}||_{2,1} = \sum_{i=1}^{N}\sqrt{a_{i1}^2 + a_{i2}^2}$ when $\boldsymbol{A} = (a_{ij})_{N\times 2} \in \mathbb{R}^{N\times 2}$. $\mu_i \geqslant 0, \mu \geqslant 0$ are regularization parameters.

The first term in (4) is a fidelity term in transformation domain, which requires the transform coefficients of clean image f and noisy image g are similar. The second term in (4) is a priority term which controls the sparsity of the transform coefficients of f. Finally, the third term is a constraint in spatial domain which controls the smoothness of the reconstructed small image patches.

Theoretically, the $\boldsymbol{U}^j, \boldsymbol{V}^j$ in the local SVD operator $\boldsymbol{T}^j$ can be set as any orthogonal matrices. However, as mentioned earlier, inappropriate basis matrices may lead the image patch groups to not have a sparse representation.

On the block local SVD operator $\boldsymbol{T}^j$, we have the following properties:

Proposition 2 $(\boldsymbol{T}^j)'\boldsymbol{T}^j = \sum_{i=1}^{I_j}(\boldsymbol{P}^{ij})'\boldsymbol{P}^{ij}$, *and* $\sum_{j=1}^{J}(\boldsymbol{T}^j)'\boldsymbol{T}^j = \sum_{j=1}^{J}\sum_{i=1}^{I_j}(\boldsymbol{P}^{ij})'\boldsymbol{P}^{ij}$ *is invertible.*

Proposition 3 *For any* $y \in \mathbb{R}^{nI_j}$, $(\boldsymbol{T}^j)'y = \sum_{i=1}^{I_j}(\boldsymbol{P}^{ij})'\boldsymbol{U}^j array(y)v_i^j$.

Proposition 4 $ker(\sum_{j=1}^{J}(\boldsymbol{T}^j)'\boldsymbol{T}^j) = \{0\}$.

The proofs of these propositions are not difficult. Here we omit the proof details and leave them to the readers.

Equation (4) is not smoothed and to solve it directly would often be slow. However, it can be efficiently optimized by popular splitting methods such as ALM [19], ADMM and Split Bregman [20].

4 Algorithm

To solve (4), we can introduce auxiliary variables $\alpha_j \in \mathbb{R}^{nI_j}$ and $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_J] \in \mathbb{R}^{nI_j \times J}$, then one can get the following constrained minimization problem

$$\begin{aligned} &\min_{\boldsymbol{\alpha}, f} \left\{\tfrac{1}{2}\textstyle\sum_{j=1}^{J} ||\alpha_j - \boldsymbol{T}^j g||^2 + \sum_{j=1}^{J} \mu_j ||\alpha_j||_p + \mu TV(f)\right\}, \\ &s.t.\ \ \alpha_j = \boldsymbol{T}^j f, \quad for \ \ j = 1, 2, \ldots, J. \end{aligned}$$

The standard Augmented Lagrangian method produces the following scheme:

$$\begin{cases} (\boldsymbol{\alpha}^k, f^k) = \underset{f, \boldsymbol{\alpha}}{\arg\min} \left\{ \dfrac{1}{2} \displaystyle\sum_{j=1}^{J} ||\alpha_j - \boldsymbol{T}^j g||^2 + \sum_{j=1}^{J} \mu_j ||\alpha_j||_p \right. \\ \qquad\qquad \left. + \mu TV(f) + \dfrac{\eta}{2} \displaystyle\sum_{j=1}^{J} \left|\left|\alpha_j - \boldsymbol{T}^j f - \lambda_j^{k-1}\right|\right|^2 \right\}, \\ \lambda_j^k = \lambda_j^{k-1} + \delta\left(\boldsymbol{T}^j f^k - \alpha_j^k\right). \end{cases} \tag{5}$$

By applying alternating minimization, it becomes

$$\begin{cases} \boldsymbol{\alpha}^k = \underset{\boldsymbol{\alpha}}{\arg\min} \left\{ \frac{1}{2} \sum_{j=1}^{J} ||\alpha_j - \boldsymbol{T}^j g||^2 + \sum_{j=1}^{J} \mu_j ||\alpha_j||_p \right. \\ \qquad \left. + \frac{\eta}{2} \sum_{j=1}^{J} \left|\left|\alpha_j - \boldsymbol{T}^j f^{k-1} - \lambda_j^{k-1}\right|\right|^2 \right\}, \\ f^k = \underset{f}{\arg\min} \left\{ \mu TV(f) + \frac{\eta}{2} \sum_{j=1}^{J} \left|\left|\alpha_j^k - \boldsymbol{T}^j f - \lambda_j^{k-1}\right|\right|^2 \right\}, \\ \lambda_j^k = \lambda_j^{k-1} + \delta\left(\boldsymbol{T}^j f^k - \alpha_j^k\right), \quad j = 1, 2, \ldots, J. \end{cases} \tag{6}$$

We note that if the local SVD operator $\boldsymbol{T}^j$ is fixed as a patch based SVD of g, then $\boldsymbol{T}^j f^{k-1}$ may not be sparse enough. This will lead to $\boldsymbol{\alpha}$ not being sparse. In order to make the transformation coefficients $\boldsymbol{\alpha}$ be more sparse, we must adjust the local SVD operator $\boldsymbol{T}^j$. Inspired by the solution of $\boldsymbol{\alpha}^k$ [Propositions (5), (6)], we choose a good $\boldsymbol{T}^j$ as following

$$\boldsymbol{T}^{j,k} = \sum_{i=1}^{I_j} v_i^{j,k} \otimes \left((\boldsymbol{U}^{j,k})' \boldsymbol{P}^{ij}\right)$$

Here, the local left-singular matrix $\boldsymbol{U}^{j,k} \in \mathbb{R}^{n \times n}$ and right-singular matrix $(\boldsymbol{V}^{j,k})' \in \mathbb{R}^{I_j \times I_j}$ are the SVD

$$\left[\boldsymbol{P}^{1j}\hat{g}, \boldsymbol{P}^{2j}\hat{g}, \ldots, \boldsymbol{P}^{I_j j}\hat{g}\right] = \boldsymbol{U}^{j,k}\Sigma_{\hat{g}}^{j}(\boldsymbol{V}^{j,k})',$$
$$(\boldsymbol{V}^{j,k})' = \left[v_1^{j,k}, v_2^{j,k}, \ldots, v_{I_j}^{j,k}\right].$$

where $\hat{g} = (\frac{g}{1+\eta} + \frac{\eta f^{k-1}}{1+\eta})$. Thus in our method, the basis functions $\boldsymbol{U}^{j,k}$ and $\boldsymbol{V}^{j,k}$ are updated during the iteration. This updating process can make the transformation coefficients more sparse, and we can get improved restoration results. This is different from the BM3D method. In that algorithm, the basis functions are chosen as DCT or wavelet basis functions and they are all fixed.

Therefore, the iteration becomes

$$\boldsymbol{T}^{j,k} = \sum_{i=1}^{I_j} v_i^{j,k} \otimes \left((\boldsymbol{U}^{j,k})'\boldsymbol{P}^{ij}\right), \tag{7}$$

$$\boldsymbol{\alpha}^k = \arg\min_{\boldsymbol{\alpha}} \left\{ \frac{1}{2}\sum_{j=1}^{J} ||\alpha_j - \boldsymbol{T}^{j,k}g||^2 + \sum_{j=1}^{J} \mu_j ||\alpha_j||_p \right. \tag{8}$$

$$\left. + \frac{\eta}{2}\sum_{j=1}^{J} \left|\left|\alpha_j - \boldsymbol{T}^{j,k}f^{k-1} - \lambda_j^{k-1}\right|\right|^2 \right\}, \tag{9}$$

$$f^k = \arg\min_f \left\{ \mu TV(f) + \frac{\eta}{2}\sum_{j=1}^{J} \left|\left|\alpha_j^k - \boldsymbol{T}^{j,k}f - \lambda_j^{k-1}\right|\right|^2 \right\}, \tag{10}$$

$$\lambda_j^k = \lambda_j^{k-1} + \delta\left(\boldsymbol{T}^{j,k}f^k - \alpha_j^k\right), \quad j = 1, 2, \ldots, J. \tag{11}$$

Both of the two minimization subproblems can be efficiently solved. For subproblem $\boldsymbol{\alpha}$, there is a closed-form solution represented by the soft and hard thresholding operators when $p = 1$ and $p = 0$, respectively.

Proposition 5 *For $p = 1$ in subproblem (8), $\alpha_j^k = \mathcal{S}(\boldsymbol{T}^{j,k}\hat{g} + \frac{\eta\lambda_j^{k-1}}{1+\eta}, \frac{\mu_j}{1+\eta})$, where $\mathcal{S}$ is a shrink operator and $\mathcal{S}(f, \mu) = \frac{f}{|f|}\max\{|f| - \mu, 0\}$.*

Proposition 6 *For $p = 0$ in subproblem (8), then $\alpha_j^k = \mathcal{H}(\boldsymbol{T}^{j,k}\hat{g} + \frac{\eta\lambda_j^{k-1}}{1+\eta}, \sqrt{\frac{2\mu_j}{1+\eta}})$ is a minimizer of subproblem (8), where $\mathcal{H}$ is a hard thresholding operator and $\mathcal{H}(f, \mu) = \begin{cases} 0, & |f| \leqslant \mu, \\ f, & |f| > \mu. \end{cases}$*

Remark: the subproblem (8) is strictly convex when $p = 1$ and thus the minimizer is unique. However, when $p = 0$, this subproblem is non-convex and it may have many minimizers. One can see this from the proof.

As to the subproblem of f, it is a ROF model [1] with a local SVD operator. It can be solved quickly by Split Bregman iteration [20]. We list the iteration scheme in the following:

$$\begin{cases} (f^{k,l}, \boldsymbol{q}^l) = \arg\min\limits_{f} \left\{ \begin{array}{l} \frac{\eta}{2}\sum_{j=1}^J \left\|\alpha_j^k - \boldsymbol{T}^{j,k} f - \lambda_j^{k-1}\right\|^2 \\ + \frac{\eta_1}{2}\left\|q_1 - (\boldsymbol{I}\otimes \boldsymbol{D}^1) f - b_1^{l-1}\right\|^2 \\ + \frac{\eta_1}{2}\left\|q_2 - (\boldsymbol{D}^2\otimes\boldsymbol{I})f - b_2^{l-1}\right\|^2 \\ +\mu||\boldsymbol{q}||_2 \end{array}\right\}, \\ b_1^{l+1} = b_1^l + (\boldsymbol{I}\otimes\boldsymbol{D}^1)f^{k,l} - q_1^l, \\ b_2^{l+1} = b_2^l + (\boldsymbol{D}^2\otimes\boldsymbol{I})f^{k,l} - q_2^l. \end{cases}$$

By applying the alternating algorithm again, we have the TV inner iteration:

$$\begin{cases} \left(\eta\sum_{j=1}^J(\boldsymbol{T}^{j,k})'\boldsymbol{T}^{j,k} + \eta_1\Delta\right) f^{k,l} = h^{k,l}, \\ q_1^l = \mathcal{S}_1\left((\boldsymbol{I}\otimes\boldsymbol{D}^1)f + b_1^{l-1},\ \ (\boldsymbol{D}^2\otimes\boldsymbol{I})f + b_2^{l-1},\ \ \frac{\mu}{\eta_1}\right), \\ q_2^l = \mathcal{S}_1\left((\boldsymbol{D}^2\otimes\boldsymbol{I})f + b_2^{l-1}, (\boldsymbol{I}\otimes\boldsymbol{D}^1)f + b_1^{l-1}, \frac{\mu}{\eta_1}\right), \\ b_1^{l+1} = b_1^l + (\boldsymbol{I}\otimes\boldsymbol{D}^1)f^{k,l} - q_1^l, \\ b_2^{l+1} = b_2^l + (\boldsymbol{D}^2\otimes\boldsymbol{I})f^{k,l} - q_2^l, \end{cases} \tag{12}$$

where $\boldsymbol{\Delta} = \boldsymbol{I}\otimes((\boldsymbol{D}^1)'\boldsymbol{D}^1) + ((\boldsymbol{D}^2)'\boldsymbol{D}^2)\otimes\boldsymbol{I}$ is the discrete Laplacian matrix, and

$$\begin{array}{l} h^{k,l} = \eta\sum_{j=1}^J(\boldsymbol{T}^{j,k})'\left(\alpha_j^k - \lambda_j^{k-1}\right) + \eta_1\left(\left(\boldsymbol{I}\otimes(\boldsymbol{D}^1)'\right)\right. \\ \qquad \left.\left(q_1^{l-1} - b_1^{l-1}\right) + \left((\boldsymbol{D}^2)'\otimes\boldsymbol{I}\right)\left(q_2^{l-1} - b_2^{l-1}\right)\right), \end{array}$$

$\mathcal{S}_1$ is an isotropic soft thresholding operator which has the expression

$$\mathcal{S}_1(x, y, \mu) = \frac{x}{\sqrt{x^2+y^2}}\max\left\{\sqrt{x^2+y^2} - \mu, 0\right\}.$$

The linear equation of $f^{k,l}$ in (12) can be efficiently solved by Gauss–Seidel iteration since $\sum_{j=1}^J(\boldsymbol{T}^{j,k})'\boldsymbol{T}^{j,k} = \sum_{j=1}^J\sum_{i=1}^{I_j}(\boldsymbol{P}^{ij})'\boldsymbol{P}^{ij}$ is a invertible diagonal matrix.

We summary the proposed BMLSVDTV method in Algorithm (1)

Algorithm 1 **BMLSVDTV** denoising algorithm
Set the initial $f^0 = g$ and some regularization parameters η, η_1, μ, μ_j, for $k = 1, 2, \ldots$

Step 1, **Block Matching**: for each image block of $\hat{g} = \frac{g}{1+\eta} + \frac{\eta f^k}{1+\eta}$ at j, find the I_j most similar image block. This is equivalent to obtaining the extract matrix $\boldsymbol{P}^{ij}, j = 1, 2, \ldots, J,\ i = 1, 2, \ldots, I_j$.

Step 2, **Basis updating**: Get the local SVD transform operator $\boldsymbol{T}^{j,k}$ by (7).

Step 3, **Sparsity Regularization**: Compute $\boldsymbol{\alpha}^k$ with soft or hard thresholding operator by Propositions (5) or (6).

Step 4, **TV Regularization**: Solve the TV subproblem (10) with several Bregman iterations (12) to get f^k. If f^k satisfies the stopping criterion $\frac{||f^k - f^{k-1}||}{||f^{k-1}||} < \epsilon$ or reaches the maximum iteration number, then stop and get the restoration result, else, go to the next step.

Step 5, **Lagrangian multiplier updating** Calculate (11) and go to step 1.

5 Convergence Analysis

If we set $p = 1$, then the proposed model (4) is strictly convex since the first term is strictly convex by the Proposition 2. With a fixed basis $\boldsymbol{T}^j$, we can show the convergence of the iteration scheme (5) and (6) as follows.

Theorem 1 *For $p = 1$, suppose f^* is the minimizer of problem* (4)*, $\forall 0 < \delta < 2$, then the sequence f^k produced by the iteration scheme* (5) *is converged and* $\lim_{k\to\infty} f^k = f^*$.

Proof The proof can be followed by some standard discussions of augmented Lagrangian method with operator splitting such as [21–23]. □

The difference between the iteration scheme (5) and (6) is that α^k is updated by a given f^{k-1} in alternating minimization scheme (6) while α^k, f^k in scheme (5) are updated simultaneously. Similarly, we have the following convergence result for this alternating scheme:

Theorem 2 . *For $p = 1$, suppose f^* is the minimizer of problem* (4)*, let $\delta = 1$, then the sequence f^k produced by the iteration scheme* (6) *is converged and* $\lim_{k\to\infty} f^k = f^*$.

Proof The proof is very similar as Theorem 1. □

Similar convergence results also can be found in [24] for splitting Bregman method, Douglas-Rachford Splitting [25] and augmented Lagrangian method [23] etc.

Let us point out if the basis $\boldsymbol{T}^j$ is updated, then this problem becomes a non-convex, and we have not got its convergence yet. However, numerical experiments show that one can obtain some better restorations with updating $\boldsymbol{T}^j$. Thus in practical computing, we use some updating bases.

6 Experiments

6.1 Implementation Details

Let us mention that both the sparsity regularization and the basis updating in the proposed algorithm can be implemented block by block because of the linear structure of the energy.

There are some parameters in the proposed method. Generally speaking, the reconstructed results are affected by these parameters. For all the cases, we let the penalty parameter $\eta = 9$, and the time step be $\delta = 0.01$. The parameter $\sqrt{n}$ which is the size of image blocks are empirically set according to the levels of the noise. For $\sigma = 10, 20, 30, 50, 75, 100$, we set $\sqrt{n} = 6, 6, 7, 8, 8, 9$, respectively. For ease of computation, we set the number of each image block group I_j as a constant. We simply let $I_j = 40, 40, 50, 60, 70, 90$ for noise with $\sigma = 10, 20, 30, 50, 75, 100$. The TV regularization parameters μ are set as $\mu = 0.1$ when $\sigma <= 20$ and $\mu = 0.2$ when $\sigma > 20$. As for the sparsity regularization parameters μ_j, we let $\mu_j = n\sigma_{local}^2(1 + \eta)$ when $p = 0$. Here the local noise variance σ_{local}^2 can be determined by some simple local noise estimation technique [26,27]. As for the parameter of the sparse norm p, we have tested some images under different noise levels and it appears that l_0 norm can produce better results than l_1. Thus in the following experiments, we just list the results produced by l_0.

Table 1 The PSNR values for different methods under several levels Gaussian noise

	$\sigma = 10$				$\sigma = 20$			
	TV	KSVD	BM3D	Proposed	TV	KSVD	BM3D	Proposed
Peppers (256 × 256)	33.34	34.25	34.68	**34.88**	29.67	30.80	31.29	**31.49**
Monarch (256 × 256)	32.98	33.67	34.12	**34.93**	29.04	29.90	30.35	**31.14**
C.man (256 × 256)	32.66	33.74	34.18	**34.34**	28.88	30.00	30.48	**30.68**
Square (256 × 256)	39.55	42.76	44.45	**47.17**	35.83	37.73	38.70	**40.59**
House (256 × 256)	34.35	35.93	**36.71**	36.68	31.25	33.11	33.77	**33.83**
Barbara (512 × 512)	30.70	34.45	34.98	**35.31**	26.54	30.85	31.78	**32.18**
Boat (512 × 512)	32.52	33.66	33.92	**33.96**	29.25	30.38	30.88	**30.95**
Hill (512 × 512)	32.55	33.38	33.62	**33.67**	29.54	30.20	30.72	**30.75**
Average	33.58	35.22	35.84	**36.37**	30.00	31.62	32.24	**32.70**
	$\sigma = 30$				$\sigma = 50$			
	TV	KSVD	BM3D	Proposed	TV	KSVD	BM3D	Proposed
Peppers (256 × 256)	27.65	28.80	29.28	**29.56**	25.21	26.08	26.68	**26.97**
Monarch (256 × 256)	26.89	27.85	28.36	**28.95**	24.32	25.32	25.82	**26.37**
C.man (256 × 256)	26.89	28.02	28.64	**28.70**	24.60	25,70	26.12	**26.43**
Square (256 × 256)	33.64	34.37	35.85	**36.80**	30.59	30.27	32.40	**32.82**
House (256 × 256)	29.50	31.18	32.09	**32.39**	27.21	28.00	29.69	**30.20**
Barbara (512 × 512)	24.78	28.57	29.81	**30.15**	23.31	25.47	27.23	**27.62**
Boat (512 × 512)	27.51	28.46	29.12	**29.17**	25.50	25.97	26.78	**26.95**
Hill (512 × 512)	28.00	28.45	29.16	**29.19**	26.21	26.32	27.19	**27.29**
Average	28.11	29.46	30.28	**30.62**	25.87	26.64	27.74	**28.09**
	$\sigma = 75$				$\sigma = 100$			
	TV	KSVD	BM3D	Proposed	TV	KSVD	BM3D	Proposed
Peppers (256 × 256)	23.40	23.64	24.73	**25.10**	22.16	21.96	23.39	**23.64**
Monarch (256 × 256)	22.36	22.81	23.90	**24.46**	21.00	20.67	22.52	**23.03**
C.man (256 × 256)	22.96	23.44	24.33	**24.56**	21.87	21.57	23.07	**23.27**
Square (256 × 256)	28.69	27.68	30.06	**30.60**	27.15	25.91	28.38	**29.11**
House (256 × 256)	25.47	25.04	27.50	**28.12**	24.21	23.60	25.87	**26.63**
Barbara (512 × 512)	22.40	22.97	25.12	**25.48**	21.75	21.87	23.62	**24.28**
Boat (512 × 512)	24.06	24.04	25.12	**25.25**	23.07	22.86	23.97	**24.15**
Hill (512 × 512)	24.92	24.89	25.68	**25.80**	23.99	24.01	24.58	**24.75**
Average	24.28	24.31	25.80	**26.17**	23.15	22.77	24.43	**24.86**

The highest PSNR values (best denoising results) are given in bold

6.2 Numerical Results

We take 8 images to test the performance of the proposed algorithm. To show its impressive denoising results, the most related TV, KSVD and BM3D are used to make some comparisons. All the PSNR values for these four methods can be found in Table 1. It is well-known that the denoising result may be affected by some parameters in these regularization methods. For KSVD and BM3D, we use the codes provided by the authors and adopt the suggested parameters. For TV, we test several regularization parameters under each noise level for one

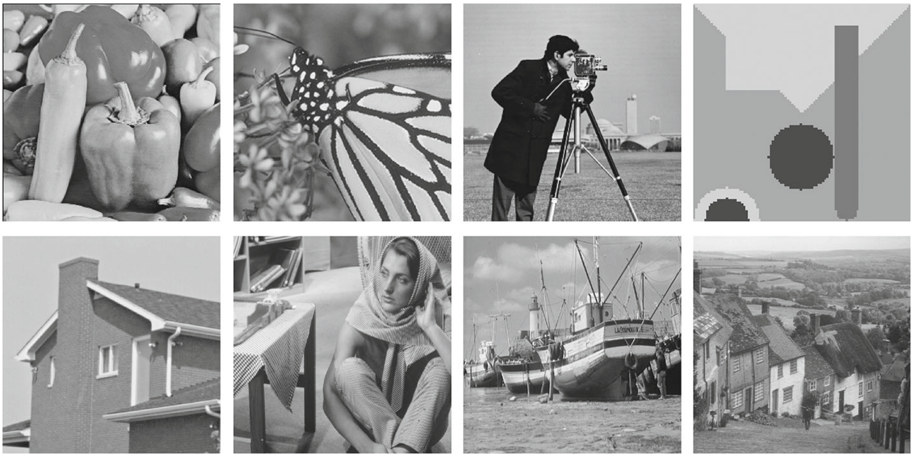

Fig. 2 The 8 test images. From left to right and top to bottom, Peppers, Monarch, Camera man, Square, House, Barbara, Boat, and Hill

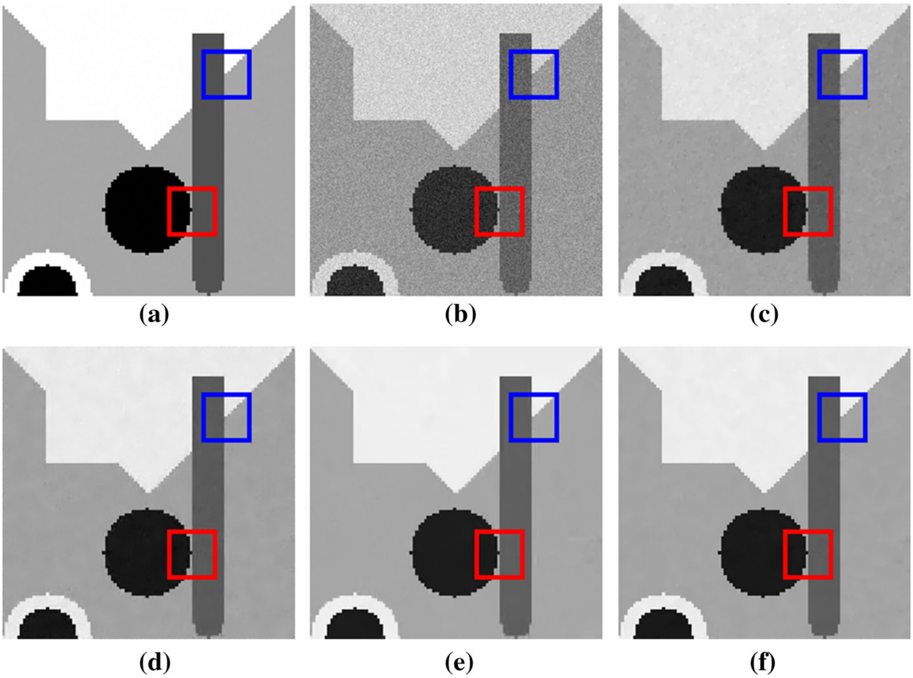

Fig. 3 The comparison of the performance on the geometry structure image 'Square' contaminated by the low level Gaussian noise with standard deviation $\sigma = 10$. For the details in the blue and red square areas, please see Fig. 4. **a** Clean, **b** Noisy, 28.10 dB, **c** TV, 39.55 dB, **d** KSVD, 42.76 dB, **e** BM3D, 44.45 dB, **f** Proposed, 47.17 dB (Color figure online)

image such as 'Monarch' and take the results with the highest PSNR for comparison, then we use the same parameters for all other 7 images. One can see that the proposed method outperforms the other methods in almost all the cases. Compared to the BM3D, which is a state-of-the-art denoising method, the improved PSNR is about 0.35 dB better on average (Fig. 2).

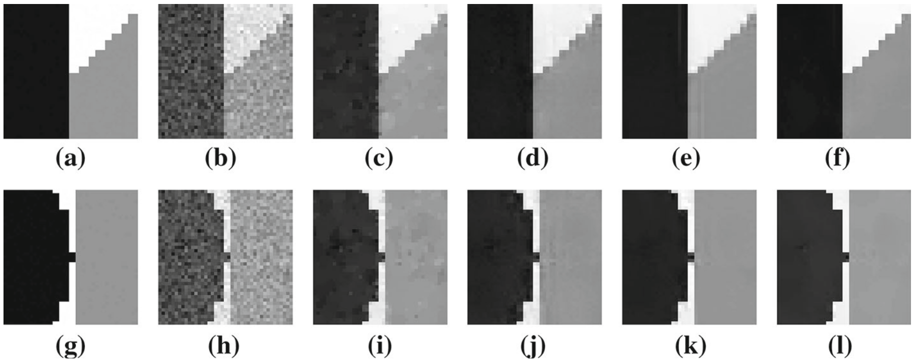

Fig. 4 The local details of the blue and red square areas in Fig. 3. **a** Clean, **b** Noisy, **c** TV, **d** KSVD, **e** BM3D, **f** Proposed, **g** Clean, **h** Noisy, **i** TV, **j** KSVD, **k** BM3D, **l** Proposed

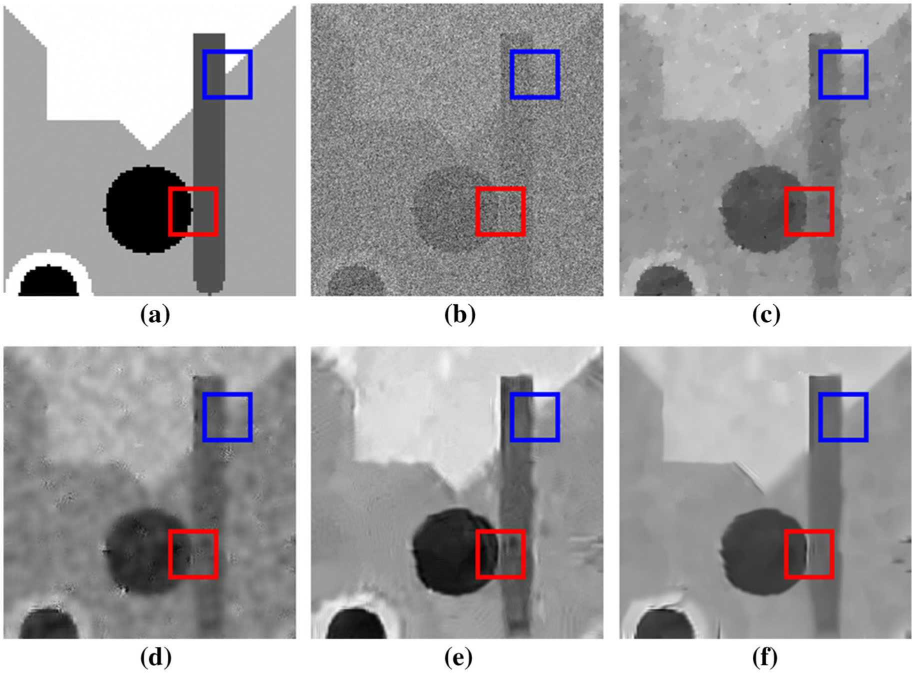

Fig. 5 The comparison of performance on the geometry structure image 'Square' contaminated by the high level Gaussian noise with standard deviation $\sigma = 100$. For the details in the blue and red square areas, please see Fig. 6. **a** Clean, **b** Noisy, 8.10 dB, **c**TV, 27.15 dB, **d** KSVD, 25.91 dB, **e** BM3D, 28.38 dB, **f** Proposed, 29.11 dB (Color figure online)

As to the visual effects, our method can reduce the artificial ringings effect which is caused by stacking the image blocks in the BM3D method, thanks to the existence of TV in the proposed method. Also, similar to the KSVD, the basis functions in our method are adaptive and thus can make the sparse representation better than the fixed ones. Our model can then keep the texture better. We take 2 example images, one is the simple image 'Square', which is almost piece-wise constant and has strong geometric structures, the other is 'Barbara', which

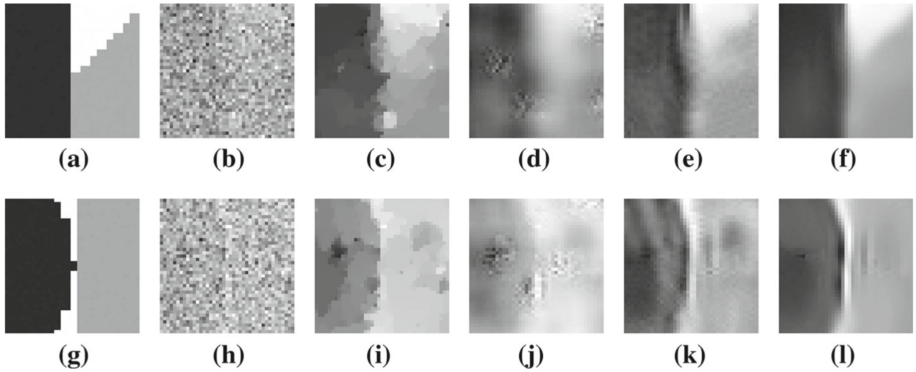

Fig. 6 The local details of the blue and red square areas in Fig. 5. **a** Clean, **b** Noisy, **c** TV, **d** KSVD, **e** BM3D, **f** Proposed, **g** Clean, **h** Noisy, **i** TV, **j** KSVD, **k** BM3D, **l** Proposed

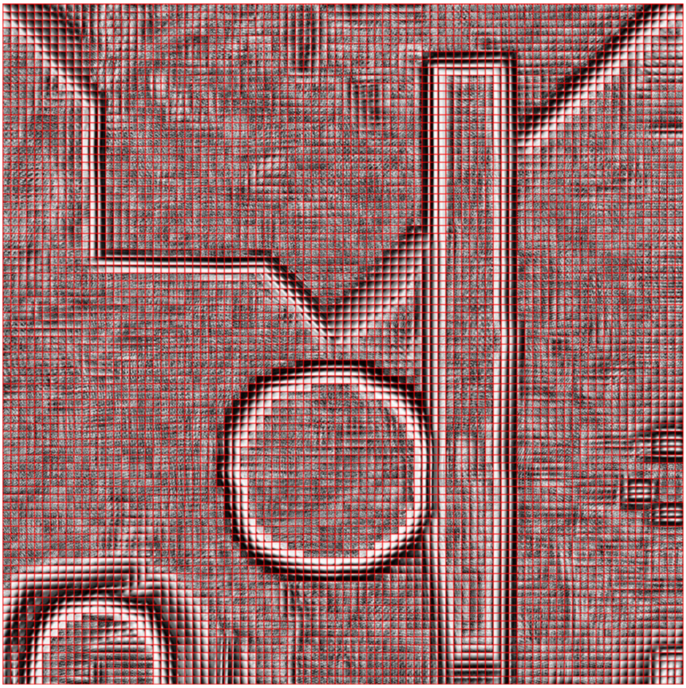

Fig. 7 The first column of the final basis functions $\boldsymbol{U}$ in the proposed method

contains much repeated texture. To show the performance under different levels of noise, for 'Square' image, we show the result under low level noise with standard deviation $\sigma = 10$ and high level with $\sigma = 100$. For 'Barbara' image, we just list the image contaminated by heavy Gaussian noise with standard deviation $\sigma = 100$. The Figs. 3, 5 and 8 demonstrate the restored results produced by the four methods. The TV in Fig. 3c can keep the strong edges well under low level noise, but produces some false edges due to the heavy noise (see Figs. 6c and 9c). Also, the repeated texture details are almost be removed due to its weak texture preserving ability. See Figs. 8c and 9i for examples. The KSVD and BM3D have

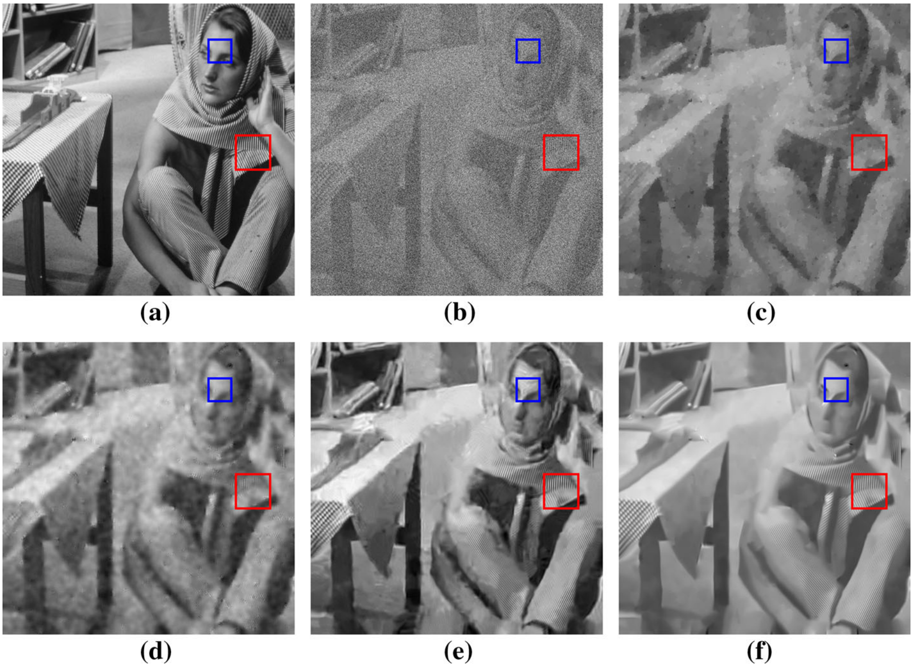

Fig. 8 The performance comparison on the texture structure image ’Barbara’ contaminated by Gaussian noise with standard deviation $\sigma = 100$. For the details in the blue and red square areas, please see Fig. 9. **a** Clean, **b** Noisy, 8.14 dB, **c** TV, 21.75 dB, **d** KSVD, 21.87 dB, **e** BM3D, 23.62 dB, **f** Proposed, 24.28 dB (Color figure online)

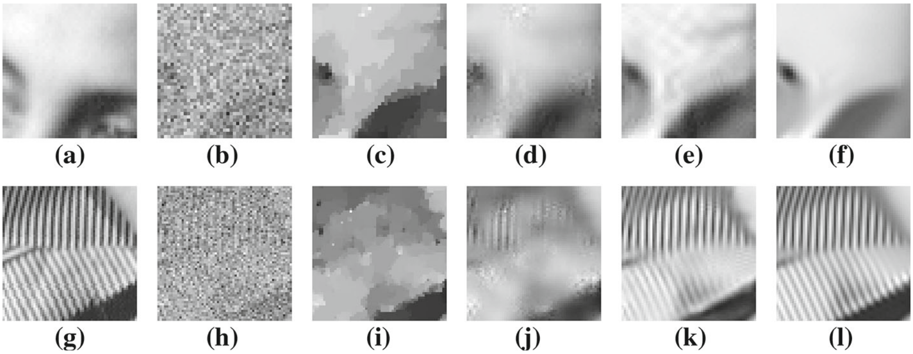

Fig. 9 The local details of the blue and red square areas in Fig. 8. **a** Clean, **b** Noisy, **c** TV, **d** KSVD, **e** BM3D, **f** Proposed, **g** Clean, **h** Noisy, **i** TV, **j** KSVD, **k** BM3D, **l** Proposed

better performance on texture restoration. However, in the smooth areas, the BM3D may produce some ringing effects as displayed in Figs. 6e, k and 9e. These artificial effects can be well controlled in the proposed method by adding the TV regularization, see the enlarged areas image in Figs. 6f, l and 9f. One can find that the restorations in these figures contain the cleanest strong edges and smooth areas. Moreover, our model can improve the texture preserving ability by basis updating, one can take Fig. 9k, l for comparison. It is easy to find that the restored textures in Fig. 9l are cleaner than the ones in Fig. 9k. For better comparison,

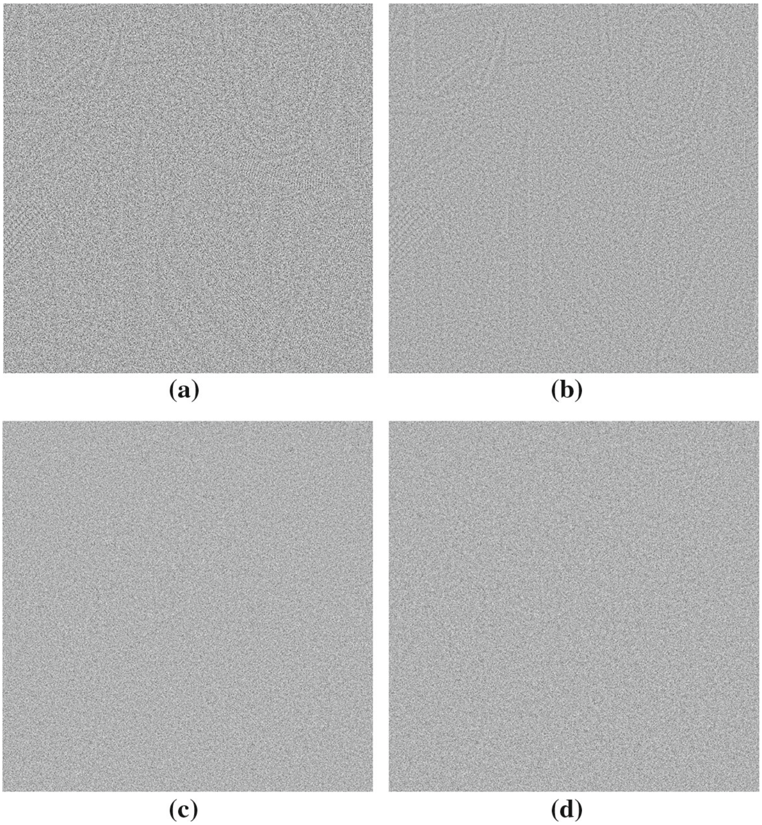

Fig. 10 The removed noise in Fig. 8 by the four methods. **a** TV, **b** KSVD, **c** BM3D, **d** Proposed

in Fig. 10, we show the removed noise for Fig. 8. Both of the removed noise by the BM3D and proposed method have little information, in fact, ours has less than BM3D's (Figs. 4, 5, 6).

As mentioned early, the basis functions $\boldsymbol{U}^{j,k}$, $\boldsymbol{V}^{j,k}$ in the proposed method are adaptive. Thus, similar to KSVD, we can show the basis functions for each image. However, the number of the basis functions $\boldsymbol{U}^{j,k} = (u_1^{j,k}, u_2^{j,k}, \ldots, u_n^{j,k})$ in our method is very large and we can not show all of them. From SVD, we can see that the eigenvector $u_1^{j,k}$ related to the first largest eigenvalue plays the most important role in sparse representation. Thus, we just show this eigenvector $u_1^{j,k}$ for each image block. For 256 × 256 image shown in Fig. 5, by using the sliding window technique which is adopted by KSVD and BM3D with 3 pixels steps, we get $J = 86 \times 86$ images patches with size 9×9, for each 9×9 image patch and its similar block groups, we display $u_1^{j,k} \in \mathbb{R}^{81}$, $j = 1, 2, \ldots, J$ in Fig. 7. In this figure, each red image block is a 9×9 array of $u_1^{j,k}$, and the total number of the image blocks is 86×86. From this basis functions, one can find near the strong edges, the basis are almost binary and it can represent the non-continuous edges well. Meanwhile, the basis functions in the tiny edges or smooth areas have many oscillations and they can represent the textures well. This is totally different from BM3D, which employs a fixed dictionary such as DCT or wavelets (Fig. 8, 9, 10).

7 Conclusion and Discussion

We have proposed a local SVD operators based sparsity and TV regularization method for image denoising. Sparsity and TV are naturally unified in a variational energy and can produce some impressive restoration results. The local SVD basis functions can improve the texture recovering ability and the global TV can reduce some artificial ringing effects in the restoration. However, the computational cost is heavy due to the existence of block matching and local SVD. Generally speaking, with our unoptimized matlab codes on a PC equipped with 3.2 GHz CPU, for images with size 256 × 256, for each outer iteration, the block matching step will take about 16 s, the basis updating and sparsity regularization will take about 20 s and the TV step is fast and will take less than 0.3 s. But the efficiency of our codes can be greatly improved by optimization since the same block matching step in BM3D just costs less than 1 s. It also can be further improved by parallel processing with a GPU. We do not discuss the implementation efficiency in this paper.

The proposed method can be easily extended to image deblurring, inpainting and even segmentation. Due to the limited page space, we could not include these. We will consider these in an upcoming paper.

Acknowledgements Liu was partially supported by The National Key Research and Development Program of China (2017YFA0604903). Liu was also supported by the China Scholarship Council for a one year visiting at UCLA. Osher was partially supported by NSF DMR 1548924 and DOE-SC0013838.

References

1. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**, 259–268 (1992)
2. Buades, A., Coll, B., Morel, J.: A review of image denoising algorithms, with a new one. Multiscale Model. Simul. **4**(2), 490–530 (2005)
3. Gilboa, G., Osher, S.: Nonlocal linear image regularization and supervised segmentation. Multiscale Model. Simul. **6**(2), 595–630 (2007)
4. Liu, J., Zheng, X.: A block nonlocal tv method for image restoration. SIAM J. Imaging Sci. **10**(2), 920–941 (2017)
5. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (2007)
6. Ji, H., Liu, C., Shen, Z., Xu, Y.: Robust video denoising using low rank matrix completion. In: *Proceeding IEEE Computer Society Conference Computer Vision and Pattern Recognition*, pp. 1791–1798 (2010)
7. Cai, J.-F., Candès, E.J., Shen, Z.: A singular value thresholding algorithm for matrix completion. SIAM J. Optim. **20**(4), 1956–1982 (2010)
8. Cands, M.B., Wakin, E.J., Boyd, S.P.: Enhancing sparsity by reweighted l1 minimization. J. Fourier Anal. Appl. **14**, 877–905 (2008)
9. Zhang, D., Hu, Y., Ye, J., Li, X., He, X.: Matrix completion by truncated nuclear norm regularization. In: *Proceeding IEEE Conference Computer Vision and Pattern Recognition*, pp. 2192–2199 (2012)
10. Gu, S., Xie, Q., Meng, D., Zuo, W., Feng, X., Zhang, L.: Weighted nuclear norm minimization and its applications to low level vision. Int. J. Comput. Vis. **121**(2), 183 (2017)
11. Xie, Y., Gu, S., Liu, Y., Zuo, W., Zhang, W., Zhang, L.: Weighted schatten -norm minimization for image denoising and background subtraction. IEEE Trans. Image Process. **25**(10), 4842–4857 (2016)
12. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: ICCV, (2011)
13. Elad, M., Aharon, M.: Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Process. **15**(12), 3736–3745 (2006)
14. Jain, V., Seung, S.: Natural image denoising with convolutional networks. In: Conference on Neural Information Processing Systems, pp. 769–776 (2009)
15. Xie, J., Xu, J., Chen, E.: Image denoising and inpainting with deep neural networks. In: International Conference on Neural Information, vol. 1, pp. 341–349 (2012)

16. Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion: a flexible framework for fast and effective image restoration. IEEE Trans. Pattern Anal. Mach. Intell. **99**, 1256–1272 (2015)
17. Danielyan, A., Katkovnik, V., Egiazarian, K.: Bm3d frames and variational image deblurring. IEEE Trans. Image Process. **21**(4), 1715–1728 (2012)
18. De Lathauwer, L., De Moor, B., Vandewalle, J.: A multilinear singular value decomposition. SIAM J. Matrix Anal. Appl. **21**(4), 1253–1278 (2000)
19. Tai, X., Wu, C.: Augmented lagrangian method, dual methods and split bregman iteration for rof model. UCLA CAM Report, Tech. Rep. 09-05, (2009)
20. Goldstein, T., Osher, S.: The split bregman method for l1 regularized problems. SIAM J. Imaging Sci. **2**, 323–343 (2009)
21. Glowinski, R.: Augmented Lagrangians and Operator-Splitting Methods in Nonlinear Mechanics. SIAM, Philadelphia (1989)
22. Wang, Y., Yang, J., Yin, W., Zhang, Y.: A new alternating minimization algorithm for total variation image reconstruction. SIAM J. Imaging Sci. **1**(3), 248–272 (2008)
23. Wu, C., Tai, X.-C.: Augmented lagrangian method, dual methods, and split bregman iteration for rof, vectorial tv, and high order models. SIAM J. Imaging Sci. **3**(3), 300–339 (2012)
24. Cai, J., Osher, S., Shen, Z.: Split bregman methods and frame based image restoration. SIAM J. Multiscale Model. Simul. **8**(2), 337–369 (2009)
25. Setzer, S.: Split bregman algorithm, douglas-rachford splitting and frame shrinkage. In: International Conference on Scale Space and Variational Methods in Computer Vision, vol. 5567, pp. 464–476 (2009)
26. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: Proceeding IEEE Conference Computer Vision and Pattern Recognition, pp. 2862–2869 (2014)
27. Liu, J., Tai, X.C., Huang, H., Huan, Z.: A weighted dictionary learning model for denoising images corrupted by mixed noise. IEEE Trans. Image Process. **22**(3), 1108–1120 (2013)