A note on patch-based low-rank minimization for fast image denoising

Haijuan Hu, Jacques Froment, Quansheng Liu

A Note on Patch-Based Low-Rank Minimization for Fast Image Denoising

Haijuan Hu[a], Jacques Froment[b], Quansheng Liu[b]

[a] *Northeastern University at Qinhuangdao, School of Mathematics and Statistics, Hebei, 066004, China. (huhaijuan61@126.com).*
[b] *Univ Bretagne-Sud, CNRS UMR 6205 LMBA, Campus de Tohannic, F-56000 Vannes, France (Jacques.Froment@univ-ubs.fr, Quansheng.Liu@univ-ubs.fr).*

Abstract

Patch-based low-rank minimization for image processing attracts much attention in recent years. The minimization of the matrix rank coupled with the Frobenius norm data fidelity can be solved by the hard thresholding filter with principle component analysis (PCA) or singular value decomposition (SVD). Based on this idea, we propose a patch-based low-rank minimization method for image denoising. The main denoising process is stated in three equivalent way: PCA, SVD and low-rank minimization. Compared to recent patch-based sparse representation methods, experiments demonstrate that the proposed method is rather rapid, and it is effective for a variety of natural grayscale images and color images, especially for texture parts in images. Further improvements of this method are also given. In addition, due to the simplicity of this method, we could provide an explanation of the choice of the threshold parameter, estimation of PSNR values, and give other insights into this method.

Keywords: image denoising, patch-based method, low-rank minimization, principal component analysis, singular value decomposition, hard thresholding

1. Introduction

Image denoising is a classical image processing problem, but it still remains very active nowadays with the massive and easy production of digital images. We mention below some important works among the vast literature which deals with image denoising.

One category of denoising methods concerns transform-based methods, for example [1, 2]. The main idea is to calculate wavelet coefficients of images, shrink the coefficients and finally reconstruct images by inverse transform. These methods apply fixed transform dictionaries to whole images. However, fixed dictionaries do not generally represent whole images very well due to the complexity of natural images. Many image details are lost while being denoised.

Another category is related to patch-based methods first proposed in [3], which explores the non-local self-similarity of natural images. Inspired by this "patch-based" idea, the authors of K-SVD [4] and BM3D [5] proposed to use dictionaries to represent small local patches instead of whole images so that sparsity of coefficients can be increased, where the dictionaries are fixed or adaptive, and compact or overcomplete. These methods greatly improve the traditional methods [1, 2], leading to very good performance. Since these works, many similar methods have been proposed to improve the denoising process, such as LPG-PCA [6], ASVD [7], PLOW [8], SAIST [9], NCSR [10], GLIDE [11], WNNM [12], ELMA[13]. However, many proposed methods are computationally complex. For example, K-SVD uses overcomplete dictionaries for sparse representation, which is time-consuming. BM3D and LPG-PCA iterate the denoising process twice; SAIST and WNNM iterate about 10 times. The computational cost is directly proportional to the number of iterations.

At the same time, the low-rank matrix approximation has been widely studied [14, 15] and applied to image processing, such as image recovery [16, 17], image super-resolution [18], image ranking [18, 19]. Many low-rank models have no explicit solution. However, the paper [14] proves that the nuclear norm minimization with the Frobenius norm data fidelity can be solved by a soft thresholding filter. (See also the paper [12] where an alternative proof is given.) Furthermore, with the help of Eckart-Young theorem [20], the paper [21] demonstrates that the solution of the exact low-rank matrix minimization problem (l_0 norm) can be obtained by a hard thresholding filter. For the convenience of the reader, we give another proof of this assertion in Appendix.

Inspired by the above theories, we propose a denoising method, which is simple for implementation and theoretical explanation. This method searches similar patches to construct similarity matrices. The denoising of similarity matrices is stated in three equivalent way: principal component analysis (PCA), singular value decomposition(SVD), and low-rank minimization. We

consider the exact matrix rank minimization, this method is denoted by patch-based low-rank minimization (PLR) method. The matrix nuclear norm minimization is also considered, which leads to the Wiener filter, denoted by Wie.

The proposed method PLR is very rapid, since we use compact dictionaries which are more computationally efficient than over-completed dictionaries. Experiments show that the proposed method is as good as the state-of-the-art methods for grayscale images and color images, such as K-SVD [4, 22], BM3D [5, 23], LPG-PCA [6], ASVD [7], PLOW [8], SAIST [9], NL-Bayes [24], DDID [25], and NLDD [26]. Thus this method can be a very good choice for being used in practice or preprocessing images for more complex image processing problems. Further improvements of PLR are also provided.

Thanks to the theoretical simplicity of PLR, we could give more insights into PLR. Firstly, this note gives a justification of the choice of threshold parameter for the proposed method. The other parameters are not very sensitive to denoising performance, therefore the choice of parameters is easy. Secondly, we provide bias-variance analysis and compare PLR with Wie. In addition, we demonstrate that optimization of each similarity matrix does not necessarily lead to final improvement. Finally, we also provide an approach to estimate PSNR values for different grayscale images to judge the denoising performance in practice. Experiments show the estimations are close to real PSNR values.

The rest of the note is organized as follows. In Section 2, we introduce our method PLR for denoising grayscale images, and give an explanation for the choice of the threshold parameter. In Section 3, deep understanding of PLR is provided. In Section 4, we give the extension to the recovery of color images, experimental results for grayscale images and color images, estimation of PSNR vlaues, and further improvement of PLR. Finally, this note is concluded in Section 5.

2. Patch-Based Low-Rank Minimization (PLR)

We consider the following noise model:

$$\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{\eta},$$

where $\boldsymbol{u}$ is the original image, $\boldsymbol{v}$ is the noisy one, and $\boldsymbol{\eta}$ is the Gaussian noise with mean 0 and standard deviation σ. The images $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{\eta}$ are of size

$M \times N$ for grayscale images, and $M \times N \times 3$ for color images. Without loss of generality, we suppose that $M = N$. The rest of this section is concentrated on grayscale images.

2.1. Proposed Algorithm

Divide the noisy image $\boldsymbol{v}$ into overlapped patches of size $d \times d$. Denote the set of all these patches as $\boldsymbol{\mathcal{S}} = \{\boldsymbol{x}_i : i = 1, 2, \cdots, (N - d + 1)^2\}$.

For each patch $\boldsymbol{x} \in \boldsymbol{\mathcal{S}}$, called reference patch, consider all the overlapped patches contained in its $n \times n$ neighborhood[1] (the total number of such patches is $(n - d + 1)^2$). Then choose the m ($m \geq d^2$) most similar patches (including the reference patch itself) to the reference patch among the $(n - d+1)^2$ patches. The similarity is determined by the Frobenius norm distance.

Next, for each reference patch, its similar patches are reshaped as column vectors, and put one next to another to form a matrix of size $d^2 \times m$, called similarity matrix. The similarity matrix is denoted as $\boldsymbol{S} = (\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_m)$, where columns of $\boldsymbol{S}$, i.e. $\boldsymbol{s}_i, i = 1, 2, \cdots, m$, are vectored similar patches. Then all the patches in the matrix $\boldsymbol{S}$ are denoised together using the hard thresholding method with the principal component (PC) basis, or equivalently, with the singular value decomposition (SVD) basis derived from the matrix $\boldsymbol{S}$; the detailed process will be given afterward. For convenience, we assume that the mean of the patches in $\boldsymbol{S}$, denoted by $\boldsymbol{s}_c := \frac{1}{m}\sum_{l=1}^{m} \boldsymbol{s}_l$, is 0. In practice, we subtract $\boldsymbol{s}_c$ from $\boldsymbol{s}_i$ to form the matrix $\boldsymbol{S}$, and add $\boldsymbol{s}_c$ to the final estimation $\bar{\boldsymbol{s}}_l$ of each patch.

Since the patches are overlapped, every pixel is finally estimated as the average of repeated estimates.

The process of denoising the matrix $\boldsymbol{S}$ is shown as follows. Firstly, we derive adaptive basis using PCA. The PC basis is the set of the eigenvectors of $\boldsymbol{S}\boldsymbol{S}^T$. Write the eigenvalue decomposition[2]

$$\boldsymbol{S}\boldsymbol{S}^T = \boldsymbol{P}\boldsymbol{\Lambda}\boldsymbol{P}^{-1} \tag{1}$$

with

$$\boldsymbol{P} = (\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_{d^2}), \boldsymbol{\Lambda} = \mathrm{diag}(m\lambda_1^2, m\lambda_2^2, \cdots, m\lambda_{d^2}^2),$$

[1]The reference patch is located at the center of the neighborhood, if the parities of d and n are the same; otherwise, the reference patch is located as near as possible to the center of the neighborhood.

[2]We assume that the matrix $\boldsymbol{S}\boldsymbol{S}^T$ has full rank, and it has no identical eigenvalues, which are generally true in practice.

where $\boldsymbol{g}_i$ denotes the i-th column of $\boldsymbol{P}$ and $\mathrm{diag}(c_1, c_2, \cdots)$ denotes the diagonal matrix with $(c_1, c_2, \cdots)$ on the diagonal. The PC basis is the set of the columns of $\boldsymbol{P}$, that is, $\{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_{d^2}\}$.

The original patches $\boldsymbol{s}_i$ in the similarity matrix $\boldsymbol{S}$ are denoised as follows:

$$\bar{\boldsymbol{s}}_l = \sum_{k=1}^{d^2} a_k \langle \boldsymbol{s}_l, \boldsymbol{g}_k \rangle \boldsymbol{g}_k, \quad l = 1, 2, \cdots, m \tag{2}$$

where

$$a_k = \begin{cases} 1 & \text{if } \lambda_k^2 > t^2, \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

t being the threshold. Or equivalently, the matrix composed of estimated patches (2) can be written as

$$\bar{\boldsymbol{S}} := (\bar{\boldsymbol{s}}_1, \bar{\boldsymbol{s}}_2, \cdots, \bar{\boldsymbol{s}}_m) = \boldsymbol{P} h(\boldsymbol{\Lambda}) \boldsymbol{P}^{-1} \boldsymbol{S}, \tag{4}$$

with

$$h(\boldsymbol{\Lambda}) = \mathrm{diag}(a_1, a_2, \cdots, a_{d^2}). \tag{5}$$

Note that

$$\frac{1}{m} \sum_{l=1}^{m} (\langle \boldsymbol{s}_l, \boldsymbol{g}_k \rangle)^2 = \lambda_k^2 \tag{6}$$

after a simple calculation. Thus λ_k can be interpreted as the standard deviation of the basis coefficients.

We could also consider the singular value decomposition (SVD) of $\boldsymbol{S}$:

$$\boldsymbol{S} = \boldsymbol{P} \boldsymbol{\Sigma} \boldsymbol{Q}^T, \tag{7}$$

where $\boldsymbol{P}$ is chosen as the same orthogonal matrix in (1), $\boldsymbol{\Sigma}$ is a diagonal matrix, and $\boldsymbol{Q}$ (of size $m \times d^2$) has orthogonal columns such that $\boldsymbol{Q}^T \boldsymbol{Q} = \boldsymbol{I}$ with $\boldsymbol{I}$ the identity matrix. Then the denoised matrix (4) is equal to

$$\hat{\boldsymbol{S}} := \boldsymbol{P} H_{t\sqrt{m}}(\boldsymbol{\Sigma}) \boldsymbol{Q}^T, \tag{8}$$

where $H_{t\sqrt{m}}(\boldsymbol{\Sigma})$ is a diagonal matrix, with the diagonal of $H_{t\sqrt{m}}(\boldsymbol{\Sigma})$ obtained by the hard thresholding operator

$$H_{t\sqrt{m}}(\boldsymbol{\Sigma})_{kk} = \begin{cases} \boldsymbol{\Sigma}_{kk} & \text{if } \boldsymbol{\Sigma}_{kk} > t\sqrt{m}, \\ 0 & \text{otherwise,} \end{cases} \quad k = 1, 2, \cdots, d^2. \tag{9}$$

In fact, the equality of (4) and (8) can be demonstrated as follows. By the equations (1) and (7), we have $\mathbf{\Lambda} = \mathbf{\Sigma}^2$, and $\boldsymbol{P}^{-1}\boldsymbol{S} = \mathbf{\Sigma}\,\boldsymbol{Q}^T$. Furthermore, by the equations (5) and (9), we get $h(\mathbf{\Lambda})\mathbf{\Sigma} = H_{t\sqrt{m}}(\mathbf{\Sigma})$. Thus it follows that $\bar{\boldsymbol{S}} = \hat{\boldsymbol{S}}$.

Finally, we show the denoised matrix in the low-rank minimization form. Theorem 2.1 stated below is an unconstrained version of the Eckart-Young theorem [20], and comes from Theorem 2(ii) in [21]. For the reader's convenience, we provide another short demonstration of the theorem in Appendix, referring to [12]. According to Theorem 2.1, it easily follows that

$$\hat{\boldsymbol{S}} = \arg\min_X \|\boldsymbol{S} - X\|_F^2 + mt^2\mathrm{Rank}(X), \tag{10}$$

where the minimum is taken over all the matrices X having the same size as $\boldsymbol{S}$, and $\|\cdot\|_F$ is the Frobenius norm. Hence the denoised matrix $\hat{\boldsymbol{S}}$ is the solution of the exact low-rank minimization problem.

Theorem 2.1. *The following low-rank minimization problem*

$$\hat{X} = \arg\min_X \|Y - X\|_F^2 + \mu\, \mathit{Rank}(X) \tag{11}$$

has the solution[3]

$$\hat{X} = U H_{\sqrt{\mu}}(\Sigma) V^T, \tag{12}$$

where U and V are derived from the SVD of Y, that is, $Y = U\Sigma V^T$, and $H_{\sqrt{\mu}}$ is the hard thresholding operator.

$$H_{\sqrt{\mu}}(\Sigma)_{kk} = \begin{cases} \Sigma_{kk} & \text{if } \Sigma_{kk} > \sqrt{\mu}, \\ 0 & \text{otherwise.} \end{cases}$$

2.2. Choice of the Threshold Parameter

Choice of parameters in algorithms is important. Theoretical explanation of the choice is generally difficult. So researchers usually choose parameters empirically. However, we could give an justification of the choice of the threshold parameter t in (3), which is crucial for the proposed algorithm. We study it by minimizing the mean squared error of denoised values of vectored patches $\boldsymbol{s}_l, (l = 1, 2, \cdots, m)$ in a similarity matrix $\boldsymbol{S}$. Denote $\boldsymbol{s}_l = \boldsymbol{u}_l + \boldsymbol{\eta}_l$,

[3]Strictly speaking, if none of the singular values of Y equals with $\sqrt{\mu}$, the solution is unique, which is generally true in practice.

where $\boldsymbol{u}_l$ and $\boldsymbol{\eta}_l$ are the vectored patches of the true image and the noise corresponding to $\boldsymbol{s}_l$ respectively.

By (2) or (4), it can be easily obtained that

$$\|\bar{\boldsymbol{s}}_l - \boldsymbol{u}_l\|^2 = \sum_{k=1}^{d^2}(a_k - 1)^2(\langle \boldsymbol{g}_k, \boldsymbol{u}_l\rangle)^2 + \sum_{k=1}^{d^2} a_k^2(\langle \boldsymbol{g}_k, \boldsymbol{\eta}_l\rangle)^2. \tag{13}$$

Assume that the PC basis $\{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_{d^2}\}$ only depends on the true value vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_m\}$ and is hence independent of $\{\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \cdots, \boldsymbol{\eta}_m\}$. Then

$$\mathbb{E}(\langle \boldsymbol{g}_k, \boldsymbol{\eta}_l\rangle)^2 = \sigma^2. \tag{14}$$

Let

$$\theta_k^2 = \frac{1}{m}\sum_{l=1}^{m}(\langle \boldsymbol{g}_k, \boldsymbol{u}_l\rangle)^2, \tag{15}$$

then by (6), we obtain

$$\mathbb{E}(\lambda_k^2) = \theta_k^2 + \sigma^2. \tag{16}$$

Thus, from (13), (14), and (15), it follows that

$$\frac{1}{m}\sum_{l=1}^{m}\|\bar{\boldsymbol{s}}_l - \boldsymbol{u}_l\|^2 \approx \sum_{k=1}^{d^2}(a_k - 1)^2\theta_k^2 + \sigma^2\sum_{k=1}^{d^2} a_k^2. \tag{17}$$

After a simple calculation, the optimal value for a_k is

$$\hat{a}_k = \begin{cases} 1 & \text{if } \theta_k^2 > \sigma^2, \\ 0 & \text{otherwise.} \end{cases}$$

Since $\lambda_k^2 \approx \theta_k^2 + \sigma^2$ by (16), the optimal value of the threshold in (3) is $t^2 \approx 2\sigma^2$. In practice, we find that $t = 1.5\sigma$ is a good choice. (We choose a litter larger threshold, since in practice, the original image contains noise, and we remove more noise than original noise.)

3. Deep Understanding of PLR

Thanks to the theoretical simplicity of our method PLR, we could gain more insights into PLR. In this section, we obtain bias-variance decomposition and compare PLR and Wie. Moreover, we show the results in case of optimization for each similarity matrix.

3.1. Bias-Variance Analysis

In the following, we obtain the Wiener filter, provide bias-variance analysis, and compare PLR and the Wiener filter.

If we replace the rank by the nuclear norm in (10), and consider the following minimization problem,

$$\hat{\boldsymbol{S}} = \arg\min_{X} \|\boldsymbol{S} - X\|_F^2 + mt^2\|X\|_*, \tag{18}$$

then the solution [14] is

$$\hat{\boldsymbol{S}} = \boldsymbol{P} S_{t\sqrt{m}}(\boldsymbol{\Sigma})\boldsymbol{Q}^T, \tag{19}$$

with

$$S_{t\sqrt{m}}(\boldsymbol{\Sigma})_{kk} = (\boldsymbol{\Sigma}_{kk} - mt^2/2)_+ \quad k = 1, 2, \cdots, d^2. \tag{20}$$

Following similar discussion to the last subsection, the optimal choice of the threshold is also $t^2 \approx 2\sigma^2$, which is the Wiener filter, the shrinkage method adopted by [27], denoted here by Wie. For Wie, denoised patches contained in $\hat{\boldsymbol{S}}$ can also be expressed as

$$\bar{\boldsymbol{s}}_l = \sum_{k=1}^{d^2} a_k \langle \boldsymbol{s}_l, \boldsymbol{g}_k \rangle \boldsymbol{g}_k, \quad \text{with } a_k = \frac{(\lambda_k^2 - \sigma^2)_+}{\lambda_k^2}, \quad l = 1, 2, \cdots, m. \tag{21}$$

Due to (2) and (21), we now use the same expression for PLR and Wie, with a_k different. Assume again that the PC basis $\{\boldsymbol{g}_1, \boldsymbol{g}_2, \cdots, \boldsymbol{g}_{d^2}\}$ only depends on the true value vectors $\{\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_m\}$ and is hence independent of $\{\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \cdots, \boldsymbol{\eta}_m\}$. With (2) or (21),

$$E\bar{\boldsymbol{s}}_l = \sum_{k=1}^{d^2} a_k \langle \boldsymbol{u}_l, \boldsymbol{g}_k \rangle \boldsymbol{g}_k, \quad l = 1, 2, \cdots, m. \tag{22}$$

Since

$$\boldsymbol{u}_l = \sum_{k=1}^{d^2} \langle \boldsymbol{u}_l, \boldsymbol{g}_k \rangle \boldsymbol{g}_k, \quad l = 1, 2, \cdots, m, \tag{23}$$

we get the sum of bias

$$\|E\bar{\boldsymbol{s}}_l - \boldsymbol{u}_l\|^2 = \sum_{k=1}^{d^2} (a_k - 1)^2 (\langle \boldsymbol{u}_l, \boldsymbol{g}_k \rangle)^2, \quad l = 1, 2, \cdots, m. \tag{24}$$

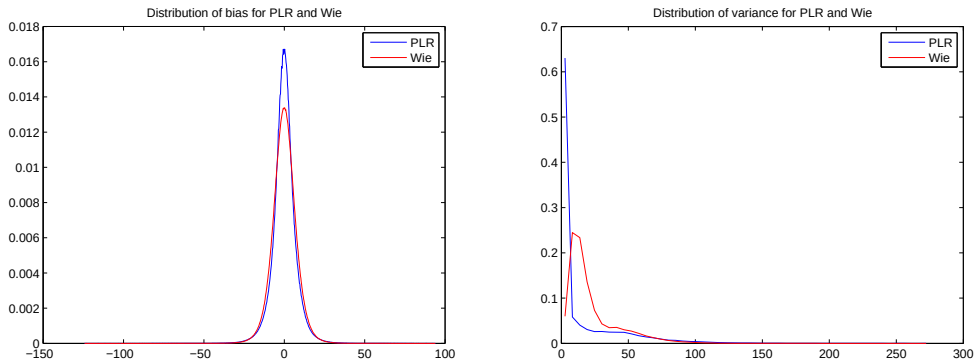

Figure 1: Densities of distribution of bias and sum of variance for PLR and Wie.

Since

$$\bar{\boldsymbol{s}}_l = \boldsymbol{P}h(\boldsymbol{\Lambda})\boldsymbol{P}^{-1}\boldsymbol{s}_l, \tag{25}$$

we obtain the covariance matrix, and further the sum of variance

$$D\bar{\boldsymbol{s}}_l = \sigma^2\boldsymbol{P}h(\boldsymbol{\Lambda})^2\boldsymbol{P}^{-1}, \quad \text{Trace}(D\bar{\boldsymbol{s}}_l) = \sigma^2\text{Trace}(h(\boldsymbol{\Lambda})^2) = \sigma^2\sum_{k=1}^{d^2} a_k^2. \tag{26}$$

On the other side,

$$E\|\bar{\boldsymbol{s}}_l - \boldsymbol{u}_l\|^2 = \sum_{k=1}^{d^2}(a_k - 1)^2(\langle \boldsymbol{g}_k, \boldsymbol{u}_l\rangle)^2 + \sigma^2\sum_{k=1}^{d^2} a_k^2, \tag{27}$$

which is thus the bias-variance decomposition.

We compare the densities of distribution of bias and the sum of variance for PLR and Wie. For bias, we use the true gray scale value, that is, we calculate the square of error for each element in all the similarity matrices, the number of which is d^2mn_s, with n_s the number of similarity matrix. The variance is estimated by (26). The densities of distributions are tested with House image, produced by histograms and displayed in Fig.1, which show that PLR generally has smaller bias and variance.

3.2. Optimization for Each Similarity Matrix

In this section, we show that optimization for each similarity matrix could not improve our PLR method.

Table 1: Comparison of PSNR values with oracle threshod of grayscale images.

Image	Lena	Barbara	Boats	Cameraman	House	Peppers
$\sigma = 10$						
PLR	35.96	35.14	33.84	33.79	36.61	34.35
PLR-Orc	36.05	35.31	33.82	33.48	36.64	34.50
PLR-Wie-Orc	35.82	34.94	33.69	33.77	36.26	34.24
$\sigma = 20$						
PLR	33.01	31.9	30.69	30.18	33.44	30.94
PLR-Orc	33.09	32.02	30.69	30.01	33.48	30.98
PLR-Wie-Orc	32.62	31.43	30.31	29.95	33.09	30.70

According to (13), the optimal choice for a_k is

$$a_k = \begin{cases} 1 & \text{if } \sum_{l=1}^{m}(\langle \boldsymbol{g}_k, \boldsymbol{u}_l\rangle)^2 > \sum_{l=1}^{m}(\langle \boldsymbol{g}_k, \boldsymbol{\eta}_l\rangle)^2, \\ 0 & \text{otherwise.} \end{cases}$$

which is not available in practice. We test this oracle choice in experiments, with the results shown in Table 1. It can be seen that this oracle choice, called PLR-Orc, improves little or does not improve the original results. More generally, if we do not restrict a_k to be 0 or 1 in (2), then the mean squared error for each patch is

$$\begin{aligned} \|\bar{\boldsymbol{s}}_l - \boldsymbol{u}_l\|^2 &= \sum_{k=1}^{d^2}(a_k - 1)^2(\langle \boldsymbol{g}_k, \boldsymbol{u}_l\rangle)^2 + \sum_{k=1}^{d^2} a_k^2(\langle \boldsymbol{g}_k, \boldsymbol{\eta}_l\rangle)^2 \\ &\quad + 2\sum_{k=1}^{d^2}(a_k - 1)a_k\langle \boldsymbol{g}_k, \boldsymbol{u}_l\rangle\langle \boldsymbol{g}_k, \boldsymbol{\eta}_l\rangle. \end{aligned}$$

Since the last term in the above equation approximates 0, we can omit this term. So the optimal choice is

$$a_k = \frac{\sum_{l=1}^{m}(\langle \boldsymbol{g}_k, \boldsymbol{u}_l\rangle)^2}{\sum_{l=1}^{m}(\langle \boldsymbol{g}_k, \boldsymbol{u}_l\rangle)^2 + (\langle \boldsymbol{g}_k, \boldsymbol{\eta}_l\rangle)^2},$$

which is the oracle choice for the Wiener filter. We also test this choice with the true original image, called PLR-Wie-Orc. The results are also shown in Table 1, which generally do not improve original algorithm.

In fact, to give an explanation to this phenomenon, the minimizer of MSE for each similarity matrix does not necessarily coincide with the minimizer of MSE for average of multiple estimate.

4. Experimental Results

In this section, we demonstrate the performance of our PLR method: our method is very fast and has similar performance to state-of-the-art methods for grayscale images and color images. Moreover, we provide estimations of PSNR values of the denoised grayscale images by the proposed method, which are close to the real values. In addition, further improvements of PLR are also given.

For simulations, we test the grayscale images used in SAIST [9] and NCSR [10][4], and test the color images utilized in BM3D [5][5]. The level of noise is supposed to be known, otherwise there are methods to estimate it; see e.g. [28]. To produce noisy images, we use same noise matrix for each tested image as done in the code of BM3D.[6]

To extend the proposed method PLR to recover color images, we adopt the strategy in[5, 24]. The RGB channels of a color image are transformed by the orthogonal matrix

$$\begin{pmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 1/\sqrt{3} \\ 1/\sqrt{2} & 0 & -1/\sqrt{2} \\ 1/\sqrt{6} & -2/\sqrt{6} & 1/\sqrt{6} \end{pmatrix}.$$

The first transformed component is the average of three channels; it is used to determine the locations of similarity matrices. Then with the same locations of similarity matrices, the three transformed components are denoised respectively, and finally are transformed to RGB channels.

4.1. Comparisons of Denoising Performance

For the proposed algorithm, we use same parameters for grayscale images and transformed components of color images. The threshold parameter is set to $t = 1.5\sigma$ as stated before. Other parameters are not very sensitive to denoising performance: the patch size is set to $d = 7$, the size of neighborhoods for selecting similar patches is set to $n = 43$, and the number of

[4]http://see.xidian.edu.cn/faculty/wsdong/Data/NCSR.rar

[5]http://www.cs.tut.fi/∼foi/GCF-BM3D/index.html#ref_software

[6]For C++ codes(K-SVD[22], NL-Bayes[24], NLDD[26]), as original codes treat noisy images as output variables, we change original codes slightly, and transfer the same noisy images in Matlab into PNG type images as input variables.

similar patches in a similarity matrix is chosen as $m = 5d^2$. Image boundaries are handled by assuming symmetric boundary conditions. For the sake of computational efficiency, the moving step from one reference patch to its neighbors both horizontally and vertically is chosen as the size of patches 7. For other comparison algorithms, we utilize the original codes released by theirs authors.

In Tables 2 and 3, we compare the PSNR (Peak Signal-to-Noise Ratio) values of our PLR method with other methods for grayscale images and color images respectively. The PSNR value is defined by PSNR $(\bar{\boldsymbol{v}}) = 20\log_{10}(255N/\|\bar{\boldsymbol{v}} - \boldsymbol{u}\|_F)$dB, where $\boldsymbol{u}$ is the original image and $\bar{\boldsymbol{v}}$ the restored one. It can be seen that, for grayscale images, PLR is better than K-SVD [4, 22], LPG-PCA [6], ASVD [7] and PLOW [8]. The average PSNR value of PLR is closed to that of BM3D [5] and DDID [25]. Comparing to SAIST [9] and WNNM [12], which have excellent performance, PLR has little difference. The comparisons show the good performance of our method. Furthermore, our method is also good for the visual comparisons. For example, as can be seen in Figures.2, 3 and 4, our method preserves the texture parts in Lena and Barbara the best among all the methods.

4.2. Comparison of Running Time

To have a clear comparison of complexities of different methods, we compare the average CPU time to remove noise with $\sigma = 20$ for the tested grayscale images of size 256×256: Peppers, House and Cameraman. All the codes are written in M-files and run in the platform of MATLAB R2011a on a 3.40GHz Intel Core i7 CPU processor. We do not include BM3D for comparison since the original code of BM3D contains MEX-files. The running time is displayed in second in Table 4. The comparisons clearly show that the proposed method is much faster than the others.

4.3. Estimation of PSNR Values

To have an estimation of the denoising performance of our method without referring to true images, we provide a method to get estimated PSNR values for grayscale images.

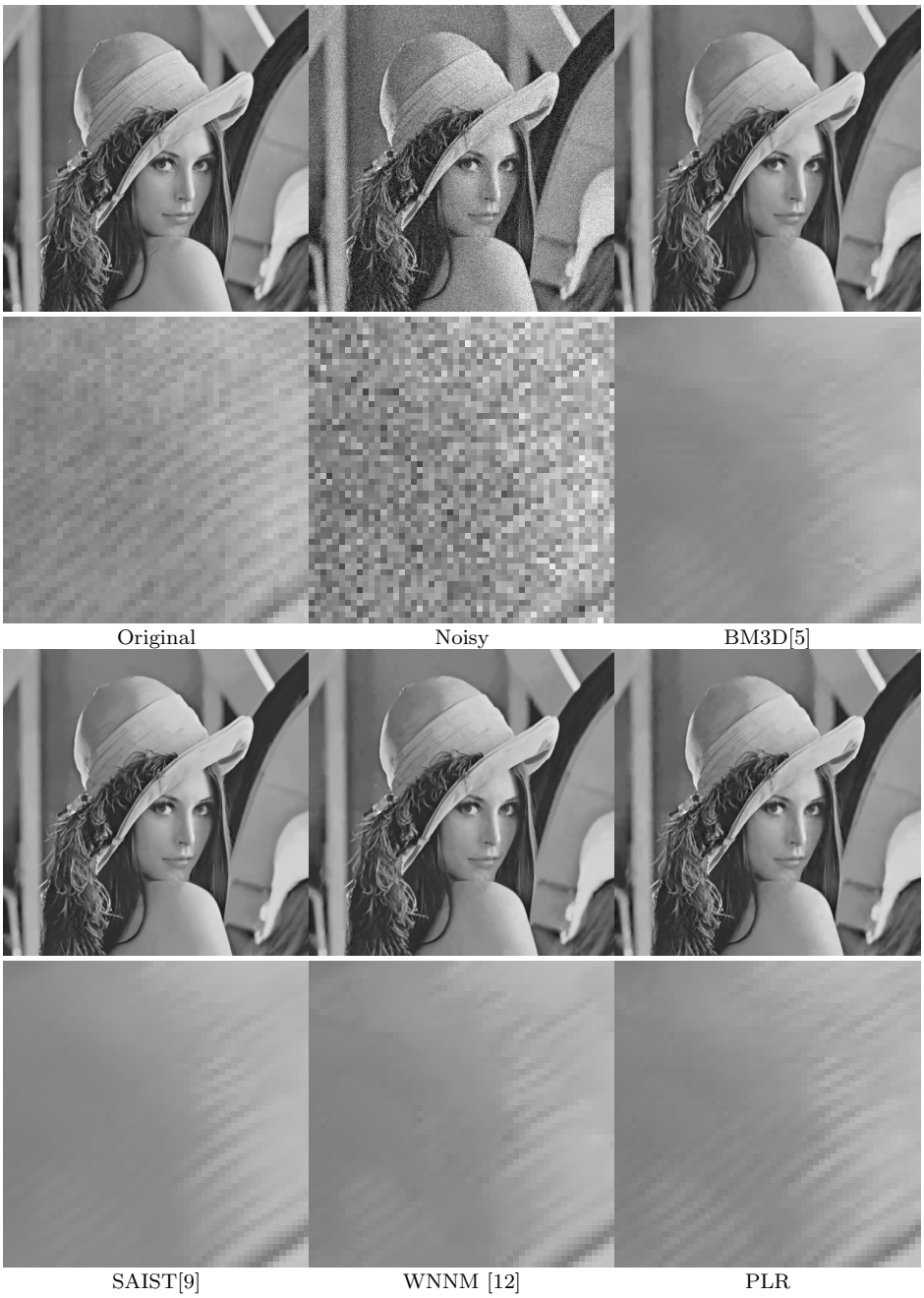

Figure 2: Compare denoised grayscale images Lena for $\sigma = 20$. To make the differences clearer, the bottom row and the second row display parts of the hat in Lena image extracted from the top row and the third row respectively.

Figure 3: Compare denoised grayscale images Barbara for $\sigma = 20$. To make the differences clearer, the bottom row and the second row display parts of Barbara images extracted from the top row and the third row respectively.

Table 2: PSNR values of grayscale images. The last column is the mean PSNR values of all the tested images. From left to right, the images are Lena, Monarch, Barbara, Boats, Cameraman, Couple, Fingerprint, Hill, House, Man, Peppers, and Straw.

Image	Lena	Mon	Bar	Boa	Cam	Cou	Fin	Hill	Hou	Man	Pep	Str	mean
$\sigma = 10$													
K-SVD[4]	35.51	33.67	34.47	33.65	33.75	33.55	32.41	33.38	35.97	33.62	34.24	30.96	33.76
K-SVD[22]	34.72	33.69	33.69	33.47	33.15	33.35	32.47	33.23	35.0	33.36	33.91	30.97	33.42
LPGPCA[6]	35.75	34.0	35.07	33.65	33.7	33.6	32.64	33.4	36.17	33.68	34.05	31.33	33.92
ASVD[7]	35.62	33.02	34.92	33.31	31.63	33.0	31.93	32.85	36.46	32.84	33.55	29.99	33.26
PLOW[8]	35.32	32.84	33.79	32.99	33.16	33.13	31.04	32.63	36.23	32.95	33.57	28.36	33.0
DDID [25]	35.81	34.17	34.67	33.74	34.05	33.87	31.84	33.56	36.5	34.02	34.56	31.24	34.0
BM3D[5]	35.93	34.12	34.98	33.92	34.19	34.04	32.46	33.62	36.71	33.98	34.68	30.92	34.13
SAIST[9]	35.9	34.76	35.24	33.91	34.3	33.96	32.69	33.65	36.66	34.12	34.82	31.6	34.3
WNNM[12]	36.05	35.04	35.52	34.07	34.42	34.13	32.81	33.75	36.93	34.2	34.95	31.72	34.46
PLR	35.96	34.02	35.14	33.84	33.79	33.8	32.59	33.63	36.61	33.93	34.35	31.1	34.06
PLR-W	35.99	34.33	35.27	33.92	34.05	33.91	32.64	33.69	36.66	34.03	34.53	31.29	34.19
PLR-Iter	35.99	34.5	35.38	33.92	34.1	33.92	32.69	33.64	36.72	34.04	34.57	31.45	34.24
$\sigma = 20$													
K-SVD[4]	32.39	29.92	30.88	30.39	30.0	30.01	28.45	30.19	33.15	30.14	30.79	26.91	30.27
K-SVD[22]	30.31	29.34	29.3	29.36	28.9	29.11	28.6	29.44	30.4	29.33	29.55	26.86	29.21
LPGPCA[6]	32.63	30.01	31.43	30.29	29.78	30.05	28.59	30.23	33.11	30.12	30.51	27.08	30.32
ASVD[7]	32.95	29.69	31.94	30.48	29.35	30.23	28.47	30.44	33.46	30.09	30.54	26.88	30.38
PLOW[8]	32.73	29.52	30.98	30.39	29.61	30.22	27.76	30.32	33.55	30.13	30.52	25.8	30.13
DDID [25]	33.09	30.49	31.8	30.77	30.52	30.62	28.44	30.62	33.6	30.63	31.33	27.48	30.78
BM3D[5]	33.05	30.35	31.78	30.88	30.49	30.76	28.81	30.72	33.77	30.59	31.29	27.08	30.8
SAIST[9]	33.08	30.76	32.16	30.84	30.45	30.66	29.01	30.69	33.75	30.67	31.32	27.58	30.91
WNNM[12]	33.12	31.11	32.2	30.98	30.74	30.8	29.01	30.78	34.03	30.74	31.55	27.63	31.06
PLR	33.01	30.25	31.9	30.69	30.18	30.46	28.96	30.64	33.44	30.57	30.94	27.35	30.7
PLR-W	32.92	30.46	31.94	30.67	30.36	30.48	28.97	30.6	33.51	30.56	31.04	27.45	30.75
PLR-Iter	33.08	30.61	32.14	30.72	30.36	30.55	28.91	30.58	33.59	30.57	31.1	27.52	30.81

Table 3: PSNR values for removing noise for color images. The last column is the mean PSNR values of all the test images.

Image	Lena	Peppers	House	F16	Baboon	mean
$\sigma = 10$						
PLOW[8]	34.72	33.86	35.58	35.47	30.25	33.98
BM3D[5]	35.21	33.78	36.23	36.68	30.64	34.51
NL-Bayes[24]	35.26	33.81	35.82	36.71	31.00	34.52
NLDD[26]	35.05	33.59	35.56	36.45	30.52	34.23
PLR	35.16	33.83	36.00	36.72	30.65	34.47
PLR-W	35.22	33.87	36.10	36.80	30.76	34.56
$\sigma = 20$						
PLOW[8]	32.07	31.41	32.77	32.25	26.19	30.94
BM3D[5]	33.01	31.82	33.83	33.76	26.97	31.88
NL-Bayes[24]	32.90	31.53	33.28	33.84	27.45	31.80
NLDD[26]	32.92	31.54	33.32	33.93	27.34	31.81
PLR	32.93	31.62	33.38	33.92	27.07	31.78
PLR-W	32.96	31.66	33.42	33.96	27.16	31.84

Table 4: Running time in second with grayscale images of size 256×256.

K-SVD	LPG-PCA	ASVD	PLOW	SAIST	WNNM	**PLR**
210	138	337	43	25	134	2

Figure 4: Compare denoised color images Lena for $\sigma = 20$. To make the differences clearer, the second row and the bottom row display parts of the hat in Lena image extracted from the top row and the third row respectively.

By means of (17), we obtain the mean squared error of vectored patches $\bar{s}_l$ in similarity matrices, where θ_k^2 is estimated by $\max\{\lambda_k^2 - \sigma^2, 0\}$. That is,

$$\frac{1}{d^2 m} \sum_{l=1}^{m} \|\bar{\boldsymbol{s}}_l - \boldsymbol{u}_l\|^2 \tag{28}$$

$$\approx \quad \frac{1}{d^2} \Big(\sum_{k=1}^{d^2} (a_k - 1)^2 \max\{\lambda_k^2 - \sigma^2, 0\} + \sigma^2 \sum_{k=1}^{d^2} a_k^2 \Big). \tag{29}$$

Denote the set of all the similarity matrices by $\{\boldsymbol{S}_i : i = 1, 2, \cdots, n_s\}$ with n_s the number of similarity matrices. Let e_i be the estimation (29) for $\boldsymbol{S}_i$. Then we can have an initial mean squared error estimation for the whole image by averaging e_i, $e = \frac{1}{n_s} \sum_i e_i$.

The final denoised gray value for each pixel is the average of denoised values obtained by denoising similarity matrices. Therefore, the estimation of mean squared error e is greater than the true one. The difference, denoted by δ, depends on the inner relation of the image. We consider correlation coefficients to measure the relation. Denote by c_i the mean correlation coefficient of all the pairwise patches in the similarity matrix $\boldsymbol{S}_i$. Let c be the average of c_i, $c = \frac{1}{n_s} \sum_i c_i$. For simplicity, we suppose that c and δ are linearly related.

To obtain the relation of c and δ, we train the images from Berkeley Segmentation Database[7] which do not include our tested images. By experiments, the best linear relation in L_2 sense is $\delta = 47.5599c + 5.3905$ in the case of $\sigma = 10$, and $\delta = 275.8265c + 19.1569$ in the case of $\sigma = 20$. Hence, the final mean squared error of denoised image can be estimated by $e - \delta$, and the estimated PSNR value is then $10 \log_{10}(255^2/(e - \delta))$.

We get estimated PSNR values for 6 tested images in the cases $\sigma = 10, 20$ and compare them with the true PSNR values in Fig.5, which shows that the estimated values are close to the real ones.

4.4. Further Improvements

To further improve the performance, we can use weighted average for the final restoration. We use the inverse of the estimated mean squared error as

[7] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping /segbench/; We utilize the 200 images in the "train" directory, transfer them to grayscale images, and take the largest squared region from left-upper corner for each image for simplicity.

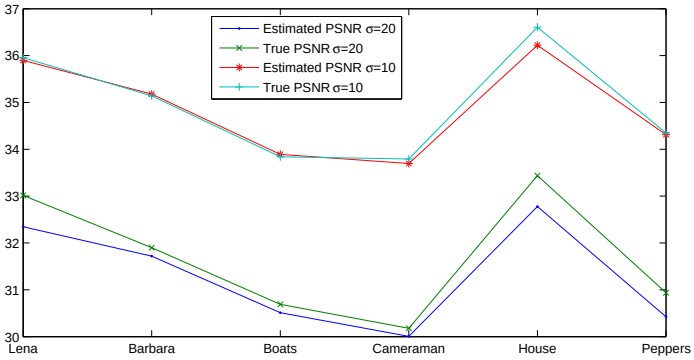

Figure 5: Estimation of PSNR values and true PSNR values.

the weight for each similarity matrix, that is,

$$\frac{1}{\sum_{k=1}^{d^2}(a_k-1)^2\max\{\lambda_k^2-\sigma^2,0\}+\sigma^2\sum_{k=1}^{d^2}a_k^2}.$$

About the choice of parameters, this algorithm, denoted by PLR-W, uses the threshold $t = 1.6\sigma$, while other parameters remain the same as PLR. The PSNR values by PLR-W are also listed in Tables 2 and 3, which demonstrate the improvement. Note that for color images, the average performance of PLR-W is the best or nearly the best among the comparing methods.

Moreover, if we use denoised image to determine the similarity matrix, the denoising performance can be further improved especially when the noise level is high. The initial denoised image is constructed with $t = 1.6\sigma$, the size of column of similarity matrices $m = 300$; the final restoration uses $t = 1.8\sigma$, $m = 100$. Other parameters stay the same as PLR. This algorithm is denoted by PLR-Iter. It improves denoising results further for grayvalue images, and the average performance is comparable to BM3D, which can be seen in Table 2.

5. Conclusion

In this note, a patch-based low-rank minimization method for image denoising is proposed, where the choice of the threshold parameter is justified and other parameters are not very sensitive to denoising performance. Compared to recently reported methods, the proposed method is very rapid

and has good denoising performance for grayscale images and color images. Therefore, this method can be a very good choice for being employed for image denoising in practice or preprocessing images to deal with more complex problems. Further improvements are also given. Moreover, we provide an effective approach to estimate PSNR values for grayscale images, which can help to judge denoising performance in practice.

Appendix A. Proof of the Theorem 2.1

Since orthogonal matrices U, V don't change the F-norm and the Rank of matrices, the problem (11) and the solution (12) are equivalent to

$$\hat{\Sigma} = \arg\min_{B} \|\Sigma - B\|_F^2 + \mu \,\mathrm{Rank}(B), \tag{A.1}$$

and

$$\hat{\Sigma} = H_{\sqrt{\mu}}(\Sigma), \quad \hat{X} = U\hat{\Sigma}V^T. \tag{A.2}$$

For the problem (A.1), it is easy to verify that, (A.2) holds for diagonal matrices B, that is,

$$H_{\sqrt{\mu}}(\Sigma) = \arg\min_{B}\{\|\Sigma - B\|_F^2 + \mu \,\mathrm{Rank}(B) : B \text{ is diagonal}\}. \tag{A.3}$$

Next, consider the general case. Let $B = U_1 D V_1$ be the SVD of B. By (A.3), to obtain (12), it suffices to prove that

$$\|\Sigma - U_1 D V_1^T\|_F^2 \geq \|\Sigma - D\|_F^2 \tag{A.4}$$

due to $\mathrm{Rank}(U_1 D V_1^T) = \mathrm{Rank}(D)$.

Note that

$$\begin{aligned} & \|\Sigma - U_1 D V_1^T\|_F^2 \\ = \;& Tr\left((\Sigma - U_1 D V_1^T)(\Sigma - U_1 D V_1^T)^T\right) \\ = \;& Tr(\Sigma\Sigma^T) + Tr(DD^T) - 2Tr(U_1 D V_1^T \Sigma^T), \end{aligned} \tag{A.5}$$

where Tr represents the matrix trace. Denote by $\gamma_k(A)$ and $\sigma_k(A)$ the k-th eigenvalue and singular value of the matrix A respectively, then we get

$$Tr(U_1 D V_1^T \Sigma^T) \leq |Tr(U_1 D V_1^T \Sigma^T)| \leq \sum_k |\gamma_k(U_1 D V_1^T \Sigma^T)|. \tag{A.6}$$

By Theorem 3.3.13 and Theorem 3.3.14 in [29], we have

$$\begin{aligned}\sum_k |\gamma_k(U_1 D V_1^T \Sigma^T)| &\leq \sum_k \sigma_k(U_1 D V_1^T \Sigma^T) \\ &\leq \sum_k \sigma_k(U_1 D V_1^T)\sigma_k(\Sigma^T) \\ &= \sum_k \sigma_k(D)\sigma_k(\Sigma^T) \\ &= Tr(D\Sigma^T). \end{aligned} \tag{A.7}$$

Thus by (A.5), (A.6) and (A.7), the inequality (A.4) holds. The proof is completed.

[1] D. Donoho, J. Johnstone, Ideal spatial adaptation by wavelet shrinkage, Biometrika 81 (3) (1994) 425–455.

[2] R. Coifman, D. Donoho, Translation-invariant denoising, Wavelets and statistics 103 (1995) 125–150.

[3] A. Buades, B. Coll, J. Morel, A review of image denoising algorithms, with a new one, Multiscale Model. Simul. 4 (2) (2005) 490–530.

[4] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, IEEE Trans. Image Process. 15 (12) (2006) 3736–3745.

[5] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Image denoising by sparse 3-D transform-domain collaborative filtering, IEEE Trans. Image Process. 16 (8) (2007) 2080–2095.

[6] L. Zhang, W. Dong, D. Zhang, G. Shi, Two-stage image denoising by principal component analysis with local pixel grouping, Pattern Recognition 43 (4) (2010) 1531–1549.

[7] Y. He, T. Gan, W. Chen, H. Wang, Adaptive denoising by singular value decomposition, IEEE Signal Process. Lett. 18 (4) (2011) 215–218.

[8] P. Chatterjee, P. Milanfar, Patch-based near-optimal image denoising, IEEE Trans. Image Process. 21 (4) (2012) 1635–1649.

[9] W. Dong, G. Shi, X. Li, Nonlocal image restoration with bilateral variance estimation: a low-rank approach, IEEE Trans. Image Process. 22 (2) (2013) 700–711.

[10] W. Dong, L. Zhang, G. Shi, X. Li, Nonlocally centralized sparse representation for image restoration., IEEE Trans. Image Process. 22 (4) (2013) 1620–1630.

[11] H. Talebi, P. Milanfar, Global image denoising, IEEE Trans. Image Process. 23 (2) (2014) 755–768.

[12] S. Gu, L. Zhang, W. Zuo, X. Feng, Weighted nuclear norm minimization with application to image denoising, in: Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 2862–2869.

[13] A. Parekh, I. W. Selesnick, Enhanced low-rank matrix approximation, IEEE Signal Process. Lett. 23 (4) (2016) 493–497.

[14] J.-F. Cai, E. J. Candès, Z. Shen, A singular value thresholding algorithm for matrix completion, SIAM J. Optimiz. 20 (4) (2010) 1956–1982.

[15] H. Schaeffer, S. Osher, A low patch-rank interpretation of texture, SIAM J. Imaging Sci. 6 (1) (2013) 226–262.

[16] J. Liu, P. Musialski, P. Wonka, J. Ye, Tensor completion for estimating missing values in visual data, IEEE Trans. Pattern Anal. Mach. Intell. 35 (1) (2013) 208.

[17] Y. Wang, C. Xu, S. You, C. Xu, D. Tao, Dct regularized extreme visual recovery, IEEE Trans. Image Process.

[18] K. Zeng, J. Yu, R. Wang, C. Li, D. Tao, Coupled deep autoencoder for single image super-resolution., IEEE Trans. Cybern. 47 (1) (2016) 27–37.

[19] J. Yu, X. Yang, G. Fei, D. Tao, Deep multimodal distance metric learning using click constraints for image ranking, IEEE Trans. Cybern. P-P (99) (2016) 1–11.

[20] C. Eckart, G. Young, The approximation of one matrix by another of lower rank, Psychometrika 1 (3) (1936) 211–218.

[21] J.-B. Hiriart-Urruty, H. Y. Le, From eckart and young approximation to moreau envelopes and vice versa, RAIRO-Operations Research 47 (03) (2013) 299–310.

[22] M. Lebrun, A. Leclaire, An implementation and detailed analysis of the k-svd image denoising algorithm, Image Processing on Line 2 (6) (2012) 96–133.

[23] M. Lebrun, An analysis and implementation of the bm3d image denoising method, American Society of Mechanical Engineers 2 (25) (2012) 175–213.

[24] M. Lebrun, A. Buades, J. M. Morel, A nonlocal bayesian image denoising algorithm, Siam Journal on Imaging Sciences 6 (3) (2013) 1665–1688.

[25] C. Knaus, M. Zwicker, Dual-domain image denoising, in: IEEE International Conference on Image Processing, 2013, pp. 440–444.

[26] N. Pierazzo, M. Lebrun, M. E. Rais, J. M. Morel, Non-local dual image denoising, in: IEEE International Conference on Image Processing, 2014, pp. 813–817.

[27] D. Muresan, T. Parks, Adaptive principal components and image denoising, in: ICIP 2003. Proceedings. 2003 International Conference on Image Processing, Vol. 1, IEEE, 2003, pp. I–101–I–104.

[28] I. M. Johnstone, B. W. Silverman, Wavelet threshold estimators for data with correlated noise, Journal of the royal statistical society: series B (statistical methodology) 59 (2) (1997) 319–351.

[29] R. A. Horn, C. R. Johnson, Topics in matrix analysis, Cambridge UP, New York.