

## Principal component dictionary-based patch grouping for image denoising<sup>☆</sup>

Shoukui Yao<sup>\*</sup>, Yi Chang, Xiaojuan Qin, Yaozong Zhang, Tianxu Zhang

National Key Laboratory of Science and Technology on Multispectral Information Processing, School of Automation, Huazhong University of Science and Technology, Wuhan, China



### ARTICLE INFO

#### Keywords:

Nonlocal self-similarity  
Transform-domain  
External knowledge  
Image denoising

### ABSTRACT

Improving denoising algorithms based on nonlocal self-similarity (NSS) to cope with increasing noise levels has become difficult. This is primarily because of difficulty in accurately grouping similar image patches solely on original spatial-domain of noisy images. To solve this problem, we propose to group similar patches on transform-domain learned from clean natural images. In this paper, we introduce a denoising algorithm comprising principal component dictionary (PCD)-based patch grouping and a low-rank approximation process. In the proposed algorithm, PCD learns from clean natural images and uses the knowledge gained to guide similar patches grouping results in noisy images. Patch grouping is directly implemented on PCD-based transform-domain. And, external knowledge and internal NSS prior are used jointly for image denoising. The results of experiments conducted indicate that the proposed denoising algorithm outperforms several state-of-the-art denoising algorithms, especially in heavy noise conditions.

### 1. Introduction

As rapid progress of digital imaging devices, image resolution increases quickly. Higher resolution images are more easily contaminated by noise. Therefore, there are increasing requirements of better denoising algorithms. Given a noisy image  $\mathbf{y} \in \mathbb{R}^{n \times 1}$ , image denoising can be generally formulated by  $\mathbf{y} = \mathbf{x} + \mathbf{v}$ , where  $\mathbf{x} \in \mathbb{R}^{n \times 1}$  is the latent clean image,  $\mathbf{v} \in \mathbb{R}^{n \times 1}$  is the additive white Gaussian noise, and  $n$  is the number of pixels in the image. As a classical problem in low-level vision, image denoising is an active topic [1–17]. Numerous denoising algorithms have been developed to date. Existing techniques can be roughly divided into two categories: conventional local prior-based methods and nonlocal self-similarity (NSS)-based methods. Conventional local prior-based methods include wavelet shrinkage based methods [1–4], total variation based methods [5,6,12], and sparse representation based methods [7,8]. Because these methods only concentrate on local priors, their performances are limited.

Algorithms based on nonlocal self-similarity (NSS) prior, which uses the recurrence of small patches in natural images, have achieved great success [9,18–27]. Buades et al. [20] proposed the nonlocal means (NLM), which was the first method to explicitly exploit NSS for image denoising. NLM method estimated each pixel as the weighted average of all pixels in image. Inspired by the success of NLM method, Dabov et al. [21] proposed the “block matching” and 3D filtering (BM3D) method. They used “block matching” to search for similar patches in the

image and grouped those patches into a 3D cube. The 3D filtering was realized by using three steps: 3-D transformation of a group, shrinkage of transform spectrum, and inverse 3-D transformation. BM3D algorithm becomes an image denoising benchmark. Mairal et al. [18] proposed the learned simultaneous sparse coding (LSSC) method by incorporating NSS and group sparse coding. They grouped similar patches and jointly decomposed the groups on subsets of learned dictionary. To improve the performance of sparse representation-based image restoration, Dong et al. [19] proposed the non-locally centralized sparse representation (NCSR) to reduce the sparse coding noise for image denoising. They exploited NSS to obtain good estimates of the sparse coding coefficients of the original image. Gu et al. [23] presented weighted nuclear norm minimization (WNNM) algorithm and applied it to image denoising by exploiting NSS. They have also achieved state-of-the-art image denoising performance. All of these algorithms adopted “block matching” to group similar patches. However, the performance of “block matching” decreases remarkably as noise levels increase. Fig. 1 shows the results of an experiment conducted using the “block matching” method, in which the red points are the centroids of the most similar patches. As shown in the figure, the error-matching rate of similar patches increases with the noise variance. The issue of how to group similar patches correctly in noisy images is an open problem.

“Block matching” is implemented on the original spatial-domain of noisy images. Because noise covers the evidence of similarity on the original spatial-domain, the performance of this method is limited. To

<sup>☆</sup> This paper has been recommended for acceptance by Zicheng Liu.

<sup>\*</sup> Corresponding author.

E-mail address: [ysk2013@hust.edu.cn](mailto:ysk2013@hust.edu.cn) (S. Yao).

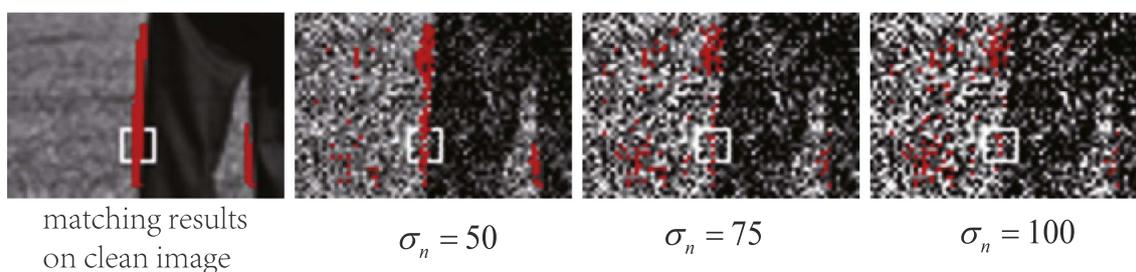


Fig. 1. Patch-matching results of the “block matching” method. White boxes denote reference patches; red points label the location of searched similar patches. The error-matching rate of similar patches increases with noise variance. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

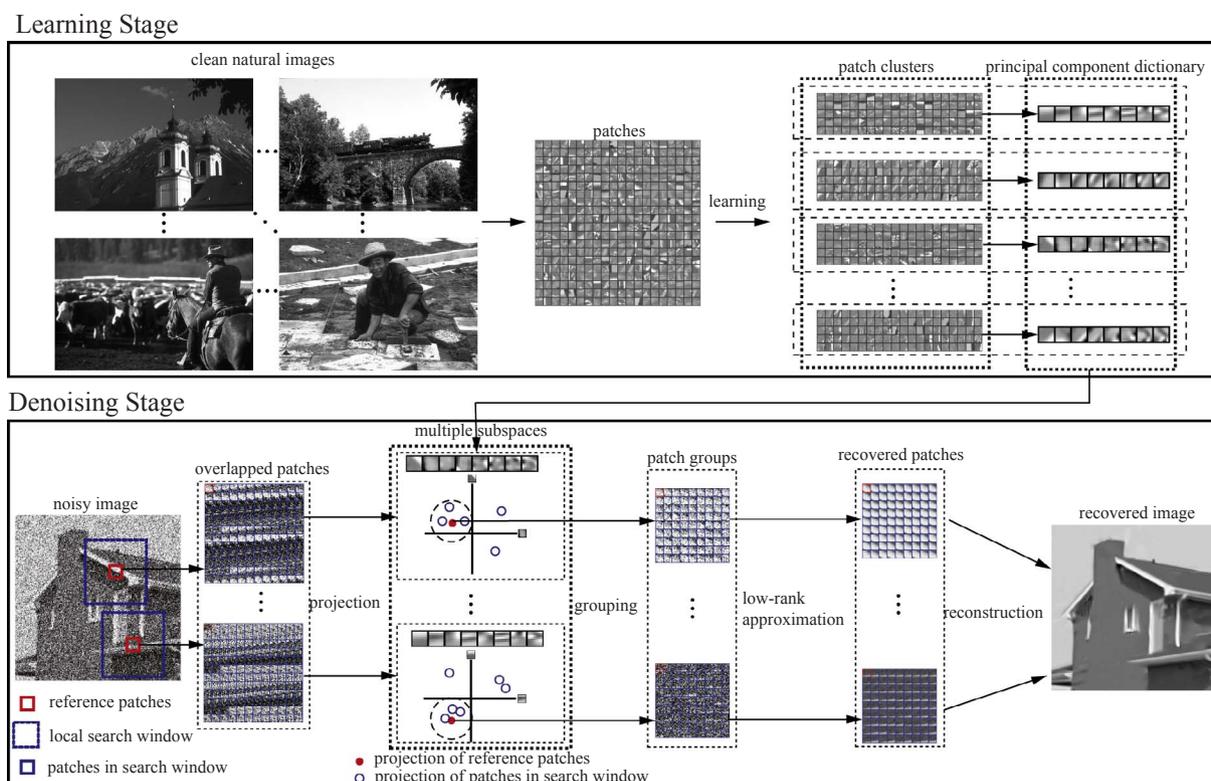


Fig. 2. Flowchart of the proposed denoising algorithm based on PCD. In the learning stage, PCD learns from clean natural images. In the denoising stage, noisy patches are projected onto the multiple subspaces constructed by PCD to estimate their principal components, which are then used as features to group similar patches. A low-rank approximation process is applied to the patch clusters for denoising.

solve this problem, we propose to group similar patches on transform-domain learned from clean natural images. On one hand, the original spatial-domain of noisy images is redundant. We hope to obtain more compact subspaces from transform-domain for more accurate image structure expression, so that patch grouping results can be promoted. On the other hand, structure information in noisy images has been contaminated by noise. We hope to enhance patch grouping results by using external clean image structure information. To achieve this goal, we learn principal component dictionary (PCD) from clean natural images, and use PCD to guide similar patches grouping. We introduce a denoising algorithm that combines PCD-based patches grouping with a low-rank approximation process. Fig. 2 presents a flowchart of the proposed denoising algorithm. As shown in the figure, PCD learns from clean natural images and constructs multiple subspaces. The eigenvectors in the sub-dictionaries of PCD represent the directions of coordinate axes in multiple subspaces. Noisy patches are projected onto multiple subspaces to estimate their principal components. Patches are grouped on transform-domain by using the estimated principal components as features. The PCD-based grouping scheme is robust to noise. A low-rank approximation process is then applied to restore similar

patches groups. Because patch grouping uses external knowledge from clean natural images and the low-rank approximation process exploits internal NSS prior from noisy images, external knowledge and internal NSS prior are simultaneously used in our proposed denoising algorithm. Experimental results show that our proposed algorithm outperforms many state-of-the-art denoising methods, especially in heavy noise conditions.

Chen et al. [28] proposed an external patch prior guided internal clustering algorithm. They learned Gaussian mixture models (GMMs) from clean images and used them to guide the clustering of noisy patches, followed by a low-rank approximation process for image restoration. However, their proposed patch grouping method is not good enough. GMM-based method cannot, in general, complete patches grouping through clustering of the patches all at once. Following their first clustering process, the number of patches in some classes was found to be too large. They, therefore, used the K-means algorithm to partition the larger classes, but the second clustering process imported errors.

Pre-process for better grouping has been used in some other NSS-based algorithms. BM3D algorithm adopted a two-stage denoising

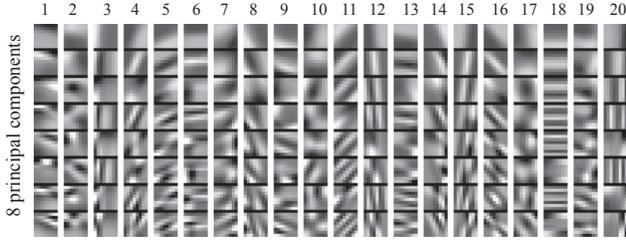


Fig. 3. Twenty sub-dictionaries of PCD. Each sub-dictionary has eight principal component vectors.

strategy with basic estimate and final estimate [21]. Basic estimate was used to improve the grouping by block-matching in the final estimate. Zhang et al. [29] proposed an efficient image denoising scheme by using principal component analysis (PCA) with local pixel grouping (LPG). The LPG-PCA algorithm also had two stages and used an initial estimation of noisy images to improve the LPG accuracy in the second stage. Different from those algorithms, our algorithm directly groups similar patches on transform-domain learned from clean natural images. On one hand, patches are grouped with their estimated principal components, which are robust to noise. On the other hand, external knowledge from clean natural images are used for better patch grouping results.

The following contributions are made in this work. (1) We propose to group similar patches on transform-domain learned from clean natural images, so as to enhance patch grouping results. (2) To promote denoising performance, we propose a denoising method that integrates external knowledge learning from clean natural images and internal NSS prior from noisy images.

The remainder of this paper is organized as follows. Section 2 introduces PCD and analyzes it. Section 3 describes the PCD-based patch grouping method and integrates it into the proposed denoising algorithm. Section 4 discusses and analyzes the experimental results obtained. Finally, Section 5 presents concluding remarks.

## 2. Principal component dictionary (PCD)

### 2.1. Definition of PCD

PCD is defined based on principal component analysis (PCA), which is used for tasks such as dimensionality reduction and feature extraction [30]. The principal component subspaces of clean natural images can be obtained by applying PCA to their patch clusters. PCD is defined as the combination of these principal component subspaces.

Denote  $\Phi$  as a parameter of a principal component subspace, with the column vectors of  $\Phi$  representing the directions of the principal component subspace. Suppose a patch cluster in which the direct current component has been removed is given. We denote the patch cluster by an  $n$ -dimensional dataset  $\{x_i\}$ , where  $1 \leq i \leq N$ . The covariance matrix of the data set  $\{x_i\}$  can be calculated as

$$\Omega = \frac{1}{N} \sum_{i=1}^N x_i x_i^T. \quad (1)$$

Because  $\Omega$  is symmetrical, its singular value decomposition (SVD) can be written as

$$\Omega = U \Lambda U^T, \quad (2)$$

where  $U = [\phi_1, \phi_2, \dots, \phi_n]$  is the orthonormal eigenvector matrix and  $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \dots, \lambda_n\}$  is the diagonal eigenvalue matrix with  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n$ . The principal components of  $\{x_i\}$  are eigenvectors corresponding to the  $m$  ( $m < n$ ) top eigenvalues, which construct a principal component subspace:

$$\Phi = [\phi_1, \phi_2, \dots, \phi_m]. \quad (3)$$

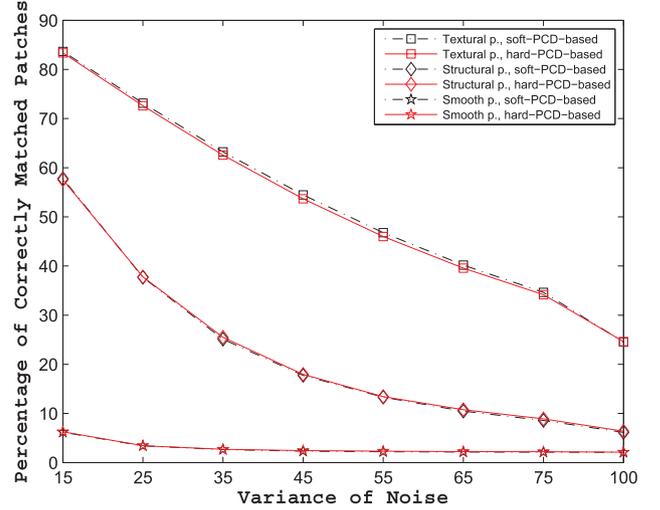


Fig. 4. Curves of percentage of correctly matched patches with two different PCD models, learning by the hard assignment method and the soft assignment method. Black broken lines denote the results of the soft method, and red full lines denote the results of the hard method. Square, rhombus, and pentacle denote patch matching results for textural, structural, and smooth, respectively. It is clear that the hard assignment method behaves nearly as same as the soft assignment method. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

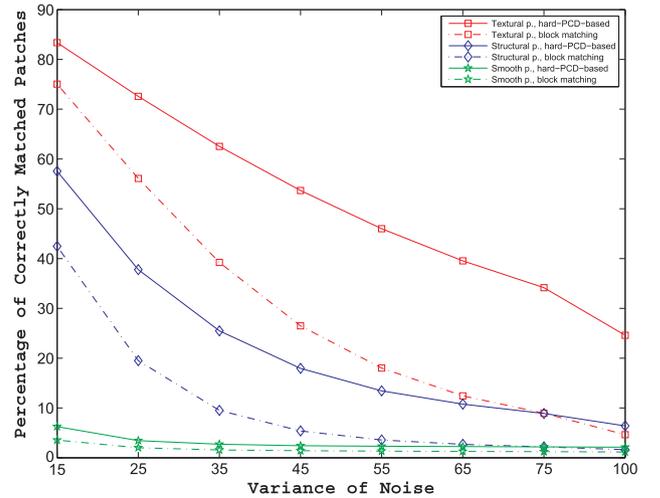


Fig. 5. Curves of percentage of correctly matched patches with different methods. Full lines denote the results of the PCD-based method, and broken lines denote the results of the “block matching” method. Red square, blue rhombus, and green pentacle denote patch matching results for textural, structural, and smooth, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Suppose that we already have several patch clusters from natural images; then, we can obtain  $K$  principal component subspaces. The combination of these principal component subspaces can construct a PCD model, which can be parameterized by  $\{\Phi_k\}, 1 \leq k \leq K$ . Dong et al. [31] learned PCA dictionary from clean patch clusters and used it to code image patches. Our proposed PCD is different from that dictionary as it does not contain the centroids of the training datasets.

### 2.2. Learning PCD from clean natural images

Hinton et al. [32] have introduced a hard version and a soft version algorithm to learn the mixtures of PCA. Here, we applied it to learn the PCD from clean natural images. A cluster can be treated as a group of patches that belong to the same principal component subspace. The projection cost is at a minimum when a patch is projected onto the

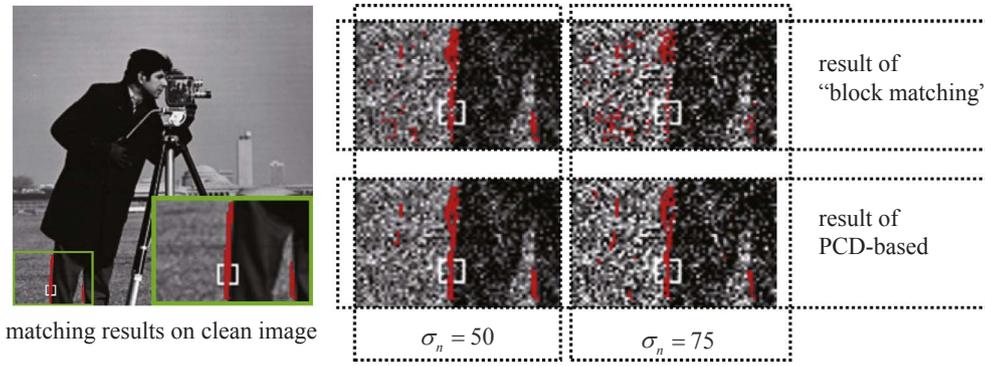


Fig. 6. Results of patch grouping for the “block matching” method and our PCD-based method. White boxes denote reference patches; red points denote the location of searched similar patches. Our method is clearly more stable than the block matching method for both  $\sigma_n = 50$  and  $\sigma_n = 75$ . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

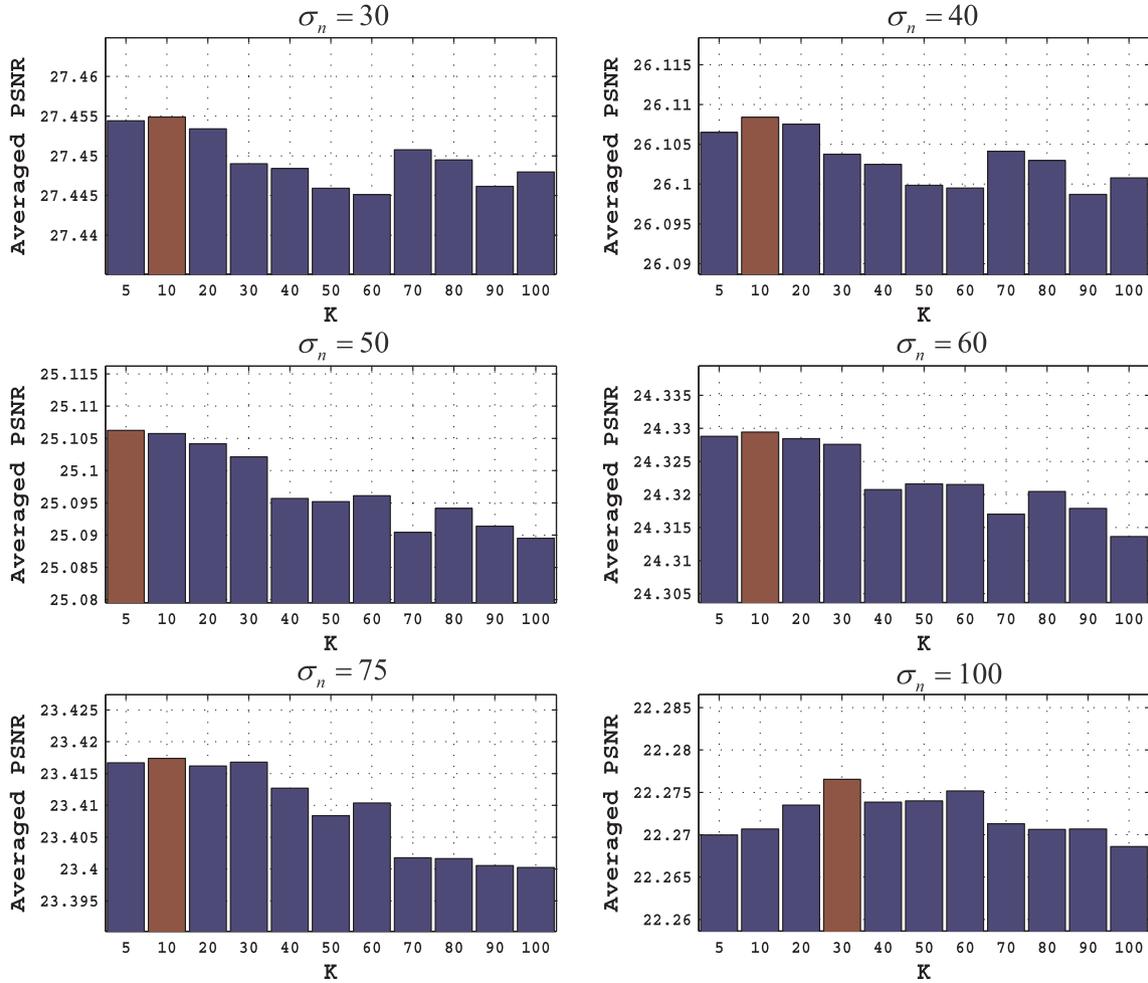


Fig. 7. Influence of changing  $K$  on the denoising results of PSNR under different noise levels, with  $\sigma_n = (30, 40, 50, 60, 75, 100)$ .

principal component subspace corresponding to the cluster to which the patch is assigned. We assumed that all patches are independently sampled.

For the hard version, an objective function can be proposed as the total projection cost of the patches:

$$J = \sum_{i=1}^N \sum_{k=1}^K \gamma_{ik} \|x_i - \Phi_k(\Phi_k^T x_i)\|^2, \quad (4)$$

where  $\gamma_{ik} \in \{0,1\}$  describes the cluster to which the image patch  $x_i$  is assigned, such that if data point  $x_i$  is assigned to cluster  $k$ , then  $\gamma_{ik} = 1$ , and  $\gamma_{ij} = 0$  for  $j \neq k$ .  $\{x_i\}$  denotes the patches from natural images. The parameter of the principal component subspace corresponding to cluster  $k$  is defined as matrix  $\Phi_k$ .

The values for  $\hat{\gamma}_{ik}$  and  $\hat{\Phi}_k$  are calculated to minimize the total projection cost as the following function:

$$(\hat{\gamma}_{ik}, \hat{\Phi}_k) = \underset{\gamma_{ik}, \Phi_k}{\operatorname{argmin}} J. \quad (5)$$

The EM algorithm is an iterative procedure in which each iteration involves two successive steps corresponding to successive optimizations with respect to  $\gamma_{ik}$  and  $\Phi_k$ . Some patches are randomly assigned into  $K$  clusters and get the initial values for  $\gamma_{ik}$  and  $\Phi_k$ . In the first phase  $J$  is minimized with respect to  $\gamma_{ik}$ , while keeping  $\Phi_k$  fixed. In the second phase  $J$  is minimized with respect to  $\Phi_k$ , while keeping  $\gamma_{ik}$  fixed. This two-stage optimization is repeated until convergence occurs. These two stages of updating  $\gamma_{ik}$  and updating  $\Phi_k$  correspond respectively to the E (expectation) and M (maximization) steps of the EM algorithm (hereafter, we use the

**Table 1**  
PSNR (dB) results of different denoising algorithms on 16 commonly used images.

	$\sigma_n = 30$						$\sigma_n = 40$					
	BM3D	EPLL	NCSR	EPIC	WNNM	PCDPG	BM3D	EPLL	NCSR	EPIC	WNNM	PCDPG
Baboon	24.59	24.88	24.74	24.79	24.84	<b>24.90</b>	23.34	23.64	23.54	23.53	23.63	<b>23.70</b>
Barbara	26.95	25.02	26.79	26.94	<b>27.40</b>	27.34	25.07	23.73	25.22	25.32	<b>25.91</b>	25.84
Boat	26.40	26.43	26.27	26.69	26.60	<b>26.65</b>	25.06	25.26	24.86	25.41	25.32	<b>25.39</b>
Cameraman	29.69	29.41	29.59	29.89	29.84	<b>29.87</b>	28.09	27.96	28.09	28.51	28.42	<b>28.46</b>
Couple	26.31	26.29	26.05	26.44	26.39	<b>26.44</b>	24.94	25.00	24.63	25.02	25.05	<b>25.12</b>
Fingerprint	23.79	23.36	23.92	23.69	24.03	<b>24.06</b>	22.04	22.07	22.48	22.46	22.77	<b>22.82</b>
Flight	28.21	28.39	28.26	28.64	28.60	<b>28.62</b>	26.71	27.06	26.66	27.22	27.19	<b>27.22</b>
Flinstones	24.08	24.30	24.29	<b>24.71</b>	24.68	24.67	22.11	22.66	22.25	<b>23.09</b>	23.02	23.01
Hill	26.84	26.87	26.77	26.99	27.03	<b>27.06</b>	25.88	25.81	25.63	25.88	25.95	<b>25.99</b>
House	32.31	31.49	32.42	32.55	32.76	<b>32.81</b>	30.89	30.09	31.13	31.17	31.47	<b>31.52</b>
Lena	29.44	29.17	29.36	29.69	29.73	<b>29.79</b>	27.77	27.79	28.01	28.27	28.26	<b>28.34</b>
Man	27.49	27.62	27.46	27.67	27.67	<b>27.70</b>	26.17	26.32	26.09	<b>26.44</b>	26.39	26.43
Monarch	28.43	28.39	28.47	28.83	28.91	<b>28.97</b>	26.71	26.90	26.85	27.43	27.47	<b>27.54</b>
Peppers	29.16	29.22	29.10	29.56	29.49	<b>29.53</b>	27.85	27.85	27.68	28.18	28.05	<b>28.11</b>
Straw	24.97	24.84	25.13	25.23	25.47	<b>25.51</b>	23.15	23.28	23.64	23.74	23.94	<b>24.01</b>
Walkbridge	25.12	25.39	25.22	25.34	25.32	<b>25.37</b>	24.02	24.21	23.96	24.19	24.18	<b>24.24</b>
Avg.	27.11	26.94	27.11	27.35	27.42	<b>27.45</b>	25.61	25.60	25.67	25.99	26.06	<b>26.11</b>
	$\sigma_n = 50$						$\sigma_n = 60$					
	BM3D	EPLL	NCSR	EPIC	WNNM	PCDPG	BM3D	EPLL	NCSR	EPIC	WNNM	PCDPG
Baboon	22.65	22.72	22.80	22.80	22.88	<b>22.92</b>	22.21	22.19	22.23	22.25	22.31	<b>22.36</b>
Barbara	24.37	22.85	24.06	24.25	<b>24.86</b>	24.85	23.42	22.23	23.06	23.25	<b>23.97</b>	23.96
Boat	24.19	24.38	23.95	24.45	24.36	<b>24.42</b>	23.48	23.65	23.20	23.65	23.64	<b>23.72</b>
Cameraman	27.08	26.91	27.01	27.42	27.31	<b>27.32</b>	26.00	26.06	26.06	26.58	26.43	<b>26.47</b>
Couple	23.88	23.94	23.73	24.08	24.07	<b>24.15</b>	23.22	23.24	23.02	23.33	23.34	<b>23.43</b>
Fingerprint	21.79	21.01	21.56	21.62	21.92	<b>21.96</b>	21.28	20.14	20.82	20.92	21.25	<b>21.31</b>
Flight	25.79	25.90	25.59	26.17	26.09	<b>26.13</b>	24.98	25.16	24.82	25.33	25.28	<b>25.32</b>
Flinstones	20.77	21.51	21.06	<b>21.71</b>	21.49	21.50	19.80	20.46	20.10	<b>20.75</b>	20.52	20.51
Hill	25.00	25.10	24.85	25.07	25.14	<b>25.18</b>	24.43	24.36	24.20	24.39	24.49	<b>24.55</b>
House	29.83	28.94	29.90	30.06	30.45	<b>30.53</b>	29.06	27.73	28.77	29.04	29.53	<b>29.60</b>
Lena	26.93	26.65	26.95	27.19	27.28	<b>27.35</b>	26.17	25.91	26.07	26.29	26.45	<b>26.51</b>
Man	25.24	25.48	25.19	25.51	25.48	<b>25.51</b>	24.54	24.69	24.43	24.79	24.73	<b>24.80</b>
Monarch	25.72	25.71	25.76	26.25	26.31	<b>26.41</b>	24.84	24.91	24.85	25.40	25.45	<b>25.55</b>
Peppers	26.63	26.74	26.52	27.03	26.91	<b>27.01</b>	25.71	25.69	25.54	26.15	26.06	<b>26.18</b>
Straw	22.38	22.02	22.49	22.74	22.94	<b>23.00</b>	21.72	21.08	21.62	21.90	22.05	<b>22.13</b>
Walkbridge	23.15	23.38	23.18	23.42	23.40	<b>23.45</b>	22.54	22.72	22.58	22.82	22.81	<b>22.86</b>
Avg.	24.71	24.58	24.66	24.99	25.06	<b>25.11</b>	23.96	23.76	23.84	24.18	24.27	<b>24.33</b>
	$\sigma_n = 75$						$\sigma_n = 100$					
	BM3D	EPLL	NCSR	EPIC	WNNM	PCDPG	BM3D	EPLL	NCSR	EPIC	WNNM	PCDPG
Baboon	21.69	21.73	21.63	21.67	21.79	<b>21.83</b>	21.08	21.02	20.99	21.08	21.12	<b>21.20</b>
Barbara	22.41	21.69	22.02	22.24	22.96	<b>23.02</b>	21.30	20.71	20.81	21.09	21.55	<b>21.66</b>
Boat	22.61	22.73	22.41	22.82	22.87	<b>22.95</b>	21.65	21.83	21.48	21.83	21.88	<b>21.95</b>
Cameraman	25.02	25.02	24.94	25.51	25.29	<b>25.36</b>	23.75	23.54	23.55	24.13	23.99	<b>24.02</b>
Couple	22.41	22.51	22.25	22.57	22.58	<b>22.65</b>	21.60	21.46	21.34	21.64	21.69	<b>21.74</b>
Fingerprint	20.50	18.95	19.99	20.13	20.51	<b>20.58</b>	19.60	17.35	18.96	19.02	19.54	<b>19.63</b>
Flight	23.88	24.01	23.87	24.28	24.30	<b>24.37</b>	22.80	22.54	22.65	23.09	23.17	<b>23.25</b>
Flinstones	18.71	19.29	18.90	<b>19.48</b>	19.24	19.24	17.47	17.86	17.56	<b>18.13</b>	17.91	17.91
Hill	23.71	23.59	23.46	23.67	23.76	<b>23.84</b>	22.78	22.65	22.48	22.80	22.79	<b>22.95</b>
House	27.57	26.81	27.47	27.88	28.28	<b>28.38</b>	26.04	25.08	25.75	26.23	26.73	<b>26.76</b>
Lena	25.12	24.70	25.02	25.33	25.53	<b>25.56</b>	23.91	23.44	23.64	24.09	24.35	<b>24.37</b>
Man	23.71	23.82	23.57	23.90	23.88	<b>23.95</b>	22.83	22.59	22.50	22.93	22.82	<b>22.91</b>
Monarch	23.79	23.82	23.67	24.27	24.31	<b>24.41</b>	22.43	22.12	22.11	22.93	22.95	<b>23.03</b>
Peppers	24.72	24.61	24.36	25.07	24.93	<b>25.10</b>	23.28	23.03	22.84	<b>23.70</b>	23.46	23.68
Straw	20.68	20.06	20.62	20.97	21.21	<b>21.26</b>	19.50	18.89	19.41	19.81	19.88	<b>20.03</b>
Walkbridge	22.07	22.03	21.86	22.13	22.13	<b>22.18</b>	21.08	21.13	20.97	21.26	21.27	<b>21.33</b>
Avg.	23.04	22.84	22.88	23.25	23.35	<b>23.42</b>	21.94	21.58	21.69	22.02	22.19	<b>22.28</b>

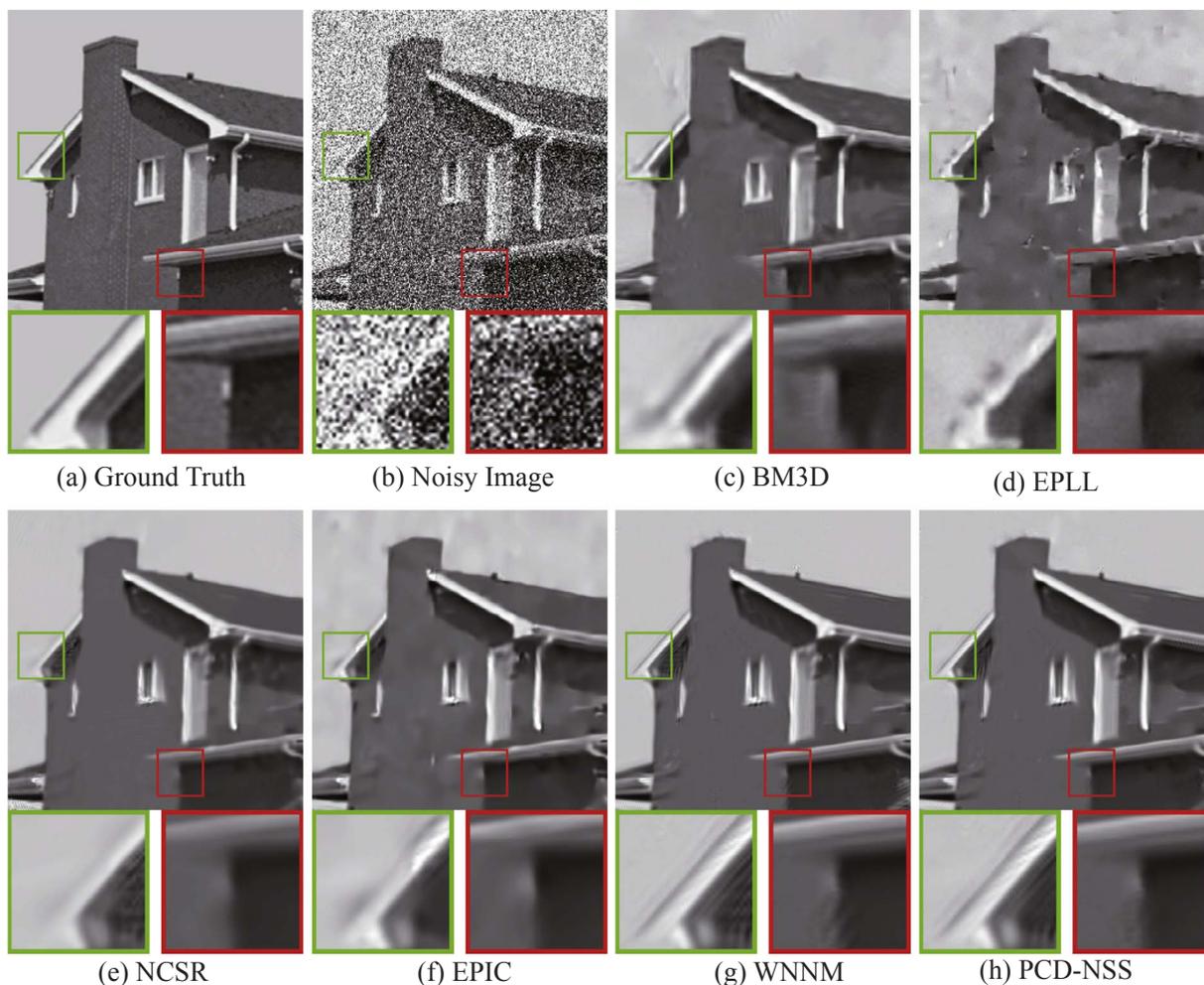
terms E-step and M-step).

In the E-step, the terms involving different  $i$  are independent; thus, we can optimize for each  $i$  separately by choosing  $\gamma_{ik}$  to be one for whichever value of  $k$  gives the minimum value of  $\|\mathbf{x}_i - \Phi_k(\Phi_k^T \mathbf{x}_i)\|^2$ . This can be expressed as

$$\gamma_{ik} = \begin{cases} 1 & \text{if } k = \underset{j}{\operatorname{argmin}} \|\mathbf{x}_i - \Phi_j(\Phi_j^T \mathbf{x}_i)\|, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

In the M-step, we consider the optimization of with  $\gamma_{ik}$  held fixed. We can minimize  $J$  by implementing PCA on the patches assigned to cluster  $k$  to update .

For the soft version, in the E-step, the responsibility of sub-model  $k$



**Fig. 8.** Denoising results on image *House* by different methods (noise level  $\sigma_n = 75$ ). (a) Ground truth. (b) Noisy image. (c) BM3D, PSNR = 27.57 dB. (d) EPLL, PSNR = 26.81 dB. (e) NCSR, PSNR = 27.47 dB. (f) EPIC, PSNR = 27.88 dB. (g) WNNM, PSNR = 28.28 dB. (h) PCDPG, PSNR = 28.38 dB. The figure is better viewed in zoomed portable document format (PDF).

for data point  $x_i$  is calculated as

$$q_k^i = e^{-E_k^i/2\sigma^2} / \left( \sum_j e^{-E_j^i/2\sigma^2} \right) \quad (7)$$

where  $E_k$  is the squared reconstruction error and  $\sigma^2$  acts like a temperature parameter. In M-step, the examples are weighted for the PCA by the responsibilities. Convergence is assessed by examining the change in the overall log-likelihood

$$-\frac{1}{2\sigma^2} \sum_{i=1}^N \sum_{k=1}^K q_k^i E_k^i - \sum_{i=1}^N \sum_{k=1}^K q_k^i \log q_k^i \quad (8)$$

In the next subsection, we conducted an patch matching experiment to compare the results of the hard version and the soft version learning method. We also implemented an patch matching experiment to compare the PCD-based method and the “block matching”.

### 2.3. Analysis of PCD

We learned PCD from patches randomly sampled from the Berkeley Segmentation Database (BSD) [33] training images (millions of patches) through both the hard assignment and the soft assignment method. The direct current components of all patches were removed in order to make the mean of the patches zero. Fig. 3 shows the sub-

dictionaries of the learned PCD, where the number of sub-dictionaries is twenty and each subspace is constructed by eight principal component vectors. As shown, each principal component sub-dictionary in PCD captures one type of contrast variation of natural images.

First, we compared the patch clustering results of the hard version and the soft version learning method. Then, we compared the performance of patch grouping based on PCD with the “block matching” method used in BM3D [21]. The patches used in the experiment were extracted from five test images (*House*, *Cameraman*, *Lena*, *Peppers*, and *Monarch*), which were divided into three classes in terms of variance: smooth patches ( $v_p < 0.002$ ), structural patches ( $0.002 < v_p < 0.02$ ), and textural patches ( $0.02 < v_p$ ). Principal component sub-dictionary was then selected for each patch. The selected principal component sub-dictionary was used to estimate the principal components of patches. The patch grouping method based on PCD used the estimated principal components as features. We randomly selected several reference patches and searched for similar patches with the PCD-based method and the “block matching” method used in BM3D. In Figs. 4 and 5, horizontal coordinate denotes the values of standard deviation of added Gaussian noise with = (15, 25, 35, 45, 55, 65, 75, 100), and the vertical coordinate represents the averaged correct rate under a specific noise level. As can be seen in Fig. 4, the hard assignment method behaves nearly as same as the soft assignment method. As can be seen in Fig. 5, the correct matching rates of the proposed PCD-based method are obviously higher than the matching rates of the “block matching” method.

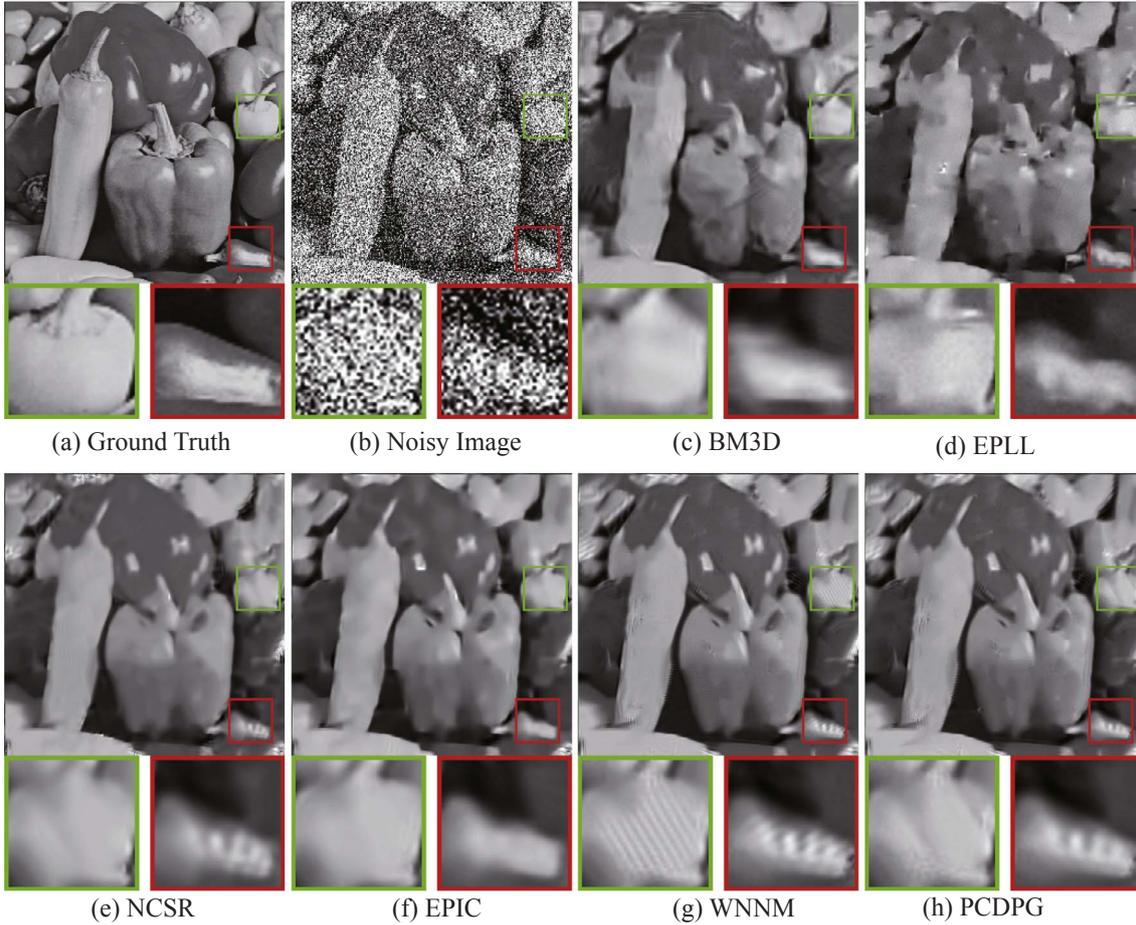


Fig. 9. Denoising results on image *Peppers* by different methods (noise level  $\sigma_n = 100$ ). (a) Ground truth. (b) Noisy image. (c) BM3D, PSNR = 23.28 dB. (d) EPLL, PSNR = 23.03 dB. (e) NCSR, PSNR = 22.84 dB. (f) EPIC, PSNR = 23.70 dB. (g) WNNM, PSNR = 23.46 dB. (h) PCDPG, PSNR = 23.68 dB. The figure is better viewed in zoomed PDF.

This advantage provides benefits for the denoising algorithm based on NSS.

### 3. PCD based patch grouping for image denoising

In this section, we give the details of PCD-based patch grouping for image denoising. The learned PCD is used to guide patch grouping, and a low-rank approximation process is applied to the patch clusters.

#### 3.1. PCD guided patch grouping

Given a noisy image, the nonlocal similar patches are searched in a local window for each reference patch. The principal component subspace is first selected for each reference patch. Then, noisy patches and their neighboring patches are projected onto the corresponding principal component subspace for similar patch searching. The patch grouping method based on PCD includes a principal component subspace selection step and a similar patch search step.

**Principal component subspace selection.** A principal component subspace from PCD is selected for every reference patch. The projection cost is minimized when the patch is projected onto its most appropriate principal component subspace. Suppose that the learned PCD has  $K$  principal component subspaces, parameterized by matrices  $\Phi_k$ . We introduce model label  $\{g_i\}, 1 \leq i \leq N$  to denote the principal component subspace selected for patch  $y_i$ , and  $g_i \in \{1, 2, \dots, K\}$ . The value for  $g_i$  is calculated by minimizing the following function:

$$g_i = \operatorname{argmin}_k \sum_{i=1}^N \|y_i - \Phi_k^T \Phi_k y_i\|_2^2. \quad (9)$$

**Similar patch search.** In this step, we project  $y_i$  and the neighboring patches in local window  $\{y_j\}, 1 \leq j \leq M$ , onto the selected principal component subspace to search similar patches. The projection of the patch  $y_i$  is defined by

$$l_i = \Phi_{g_i}^T y_i. \quad (10)$$

The estimated principal components can be calculated as  $\Phi_{g_i} l_i$ . We use the estimated principal components as the features for similar patch searching. The distance between patch  $y_i$  and  $y_j$  can be expressed as follows:

$$d_{ij} = \|\Phi_{g_i} l_i - \Phi_{g_j} l_j\|_2^2 = \|l_i - l_j\|_2^2 \quad (11)$$

As shown in the equation, the distance between two patches can be transformed into the distance between two points on the PCD-based transform-domain. Therefore, we directly search for similar patches on the PCD-based transform-domain.

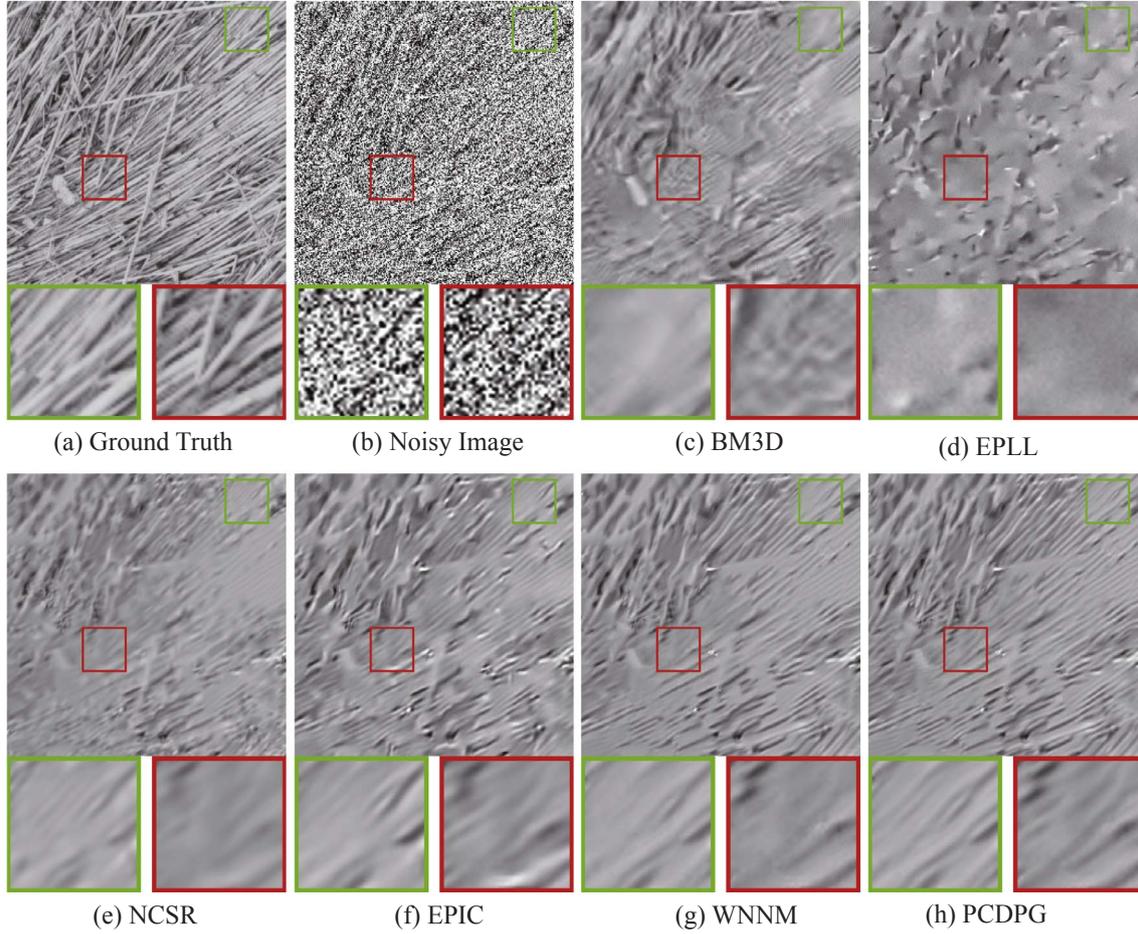
Fig. 6 compares the results of patch grouping by the “block matching” method to that of our PCD-based method. The white boxes denote the reference patches, whereas the red points denote the location of the searched similar patches. It is clear that our method is more stable than the “block matching” method for both  $\sigma_n = 50$  and  $\sigma_n = 75$ .

#### 3.2. Low-rank approximation process

Low-rank approximation is achieved by stacking the patch  $y_i$  and its similar patches into a matrix, denoted by  $Y_i \in \mathbb{R}^{n \times (S+1)}$ ,

$$Y_i = [y_i, y_{i_1}, y_{i_2}, \dots, y_{i_S}], \quad (12)$$

where  $S$  is the number of similar patches selected for the patch  $y_i$  in  $i$ -th



**Fig. 10.** Denoising results on image *Straw* by different methods (noise level  $\sigma_n = 100$ ). (a) Ground truth. (b) Noisy image. (c) BM3D, PSNR = 19.50 dB. (d) EPLL, PSNR = 18.89 dB. (e) NCSR, PSNR = 19.41 dB. (f) EPIC, PSNR = 19.81 dB. (g) WNNM, PSNR = 19.88 dB. (h) PCDPG, PSNR = 20.03 dB. The figure is better viewed in zoomed PDF.

**Table 2**  
Average denoising results (PSNR (dB)) by different denoising methods on 40 randomly selected images from the BSD test dataset.

$\sigma_n$	BM3D	NCSR	EPLL	EPIC	WNNM	PCDPG
10	33.49	33.64	33.48	33.75	33.73	<b>33.79</b>
30	28.01	28.08	28.05	28.23	28.24	<b>28.31</b>
50	25.92	25.95	25.92	26.16	26.16	<b>26.24</b>
75	24.54	24.43	24.38	24.68	24.71	<b>24.80</b>
100	23.10	23.43	23.38	23.73	23.72	<b>23.82</b>

patch group. We can assume the matrix of the vectorized patches from a patch group is of low rank and model the matrix as follows:

$$Y_i = Z_i + V_i, \quad (13)$$

where  $Z_i$  is the corresponding low-rank matrix and  $V_i$  denotes the Gaussian noise matrix. Then, an energy function can be proposed to estimate  $Z_i$  [23]:

$$\hat{Z}_i = \operatorname{argmin}_{Z_i} \frac{1}{\sigma_n^2} \|Y_i - Z_i\|_F^2 + \|Z_i\|_{\omega,*}, \quad (14)$$

where  $\sigma_n^2$  is the noise variance. Denote the  $j$ -th singular value of  $Z_i$  by  $\beta_j(Z_i)$  and let  $U\Sigma V^T$  be the SVD of  $Y_i$ . The optimal solution to this problem is given by Ref. [23]:

$$\hat{Z}_i = U(\Sigma - \tau \operatorname{diag}(\alpha)_+)V^T, \quad (15)$$

where  $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_j, \dots, \alpha_n]$ ,  $\alpha_j = \frac{1}{(\beta_j(Z_i) + \varepsilon)}$ , and  $(*)_+ = \max(*, 0)$ .

The value of each pixel within the reconstructed image is calculated by averaging the values of pixels contained by the overlapped

recovered patches.

### 3.3. The proposed denoising algorithm

The PCD-based patch grouping method and the low-rank approximation process are combined to form our proposed denoising algorithm. To enhance the output of this denoising algorithm, we iterate these procedures. Given a noisy image  $y$ , we start with some initial guess  $x^0$ . The standard deviation of noise in iteration  $t$  is estimated by

$$\sigma^{t+1} = \eta \times \sqrt{\sigma_n^2 - \|y - x^t\|_2^2}. \quad (16)$$

The process iterates until convergence occurs. Algorithm 1 outlines the complete optimization process.

**Algorithm 1.** PCD-based patch grouping for image denoising

- 1: **Input:** Noisy image  $y$ , noise standard deviation  $\sigma_n$ , a reasonable scaling factor  $\eta$ , learned PCD model  $\{\Phi_k\}$ .
- 2: **Output:** Denoised image  $x^{IterNum}$ .
- 3: **Initialization:**  $x^0 = y, y^0 = y, \sigma^0 = \sigma_n$ .
- 4: **for**  $t = 1: IterNum$  **do**
- 5:   Iteration regularization:  

$$y^t = x^{t-1} + \delta(y - y^{t-1}); \sigma^t = \eta \times \sqrt{\sigma_n^2 - \|y - x^{t-1}\|_2^2};$$
- 6:   **for** each reference patch  $y_i^t$  **do**
- 7:     Select principal component subspace for each reference patch;
- 8:     Search for similar patches;
- 9:     Construct low-rank matrices  $Y_i^t$ ;

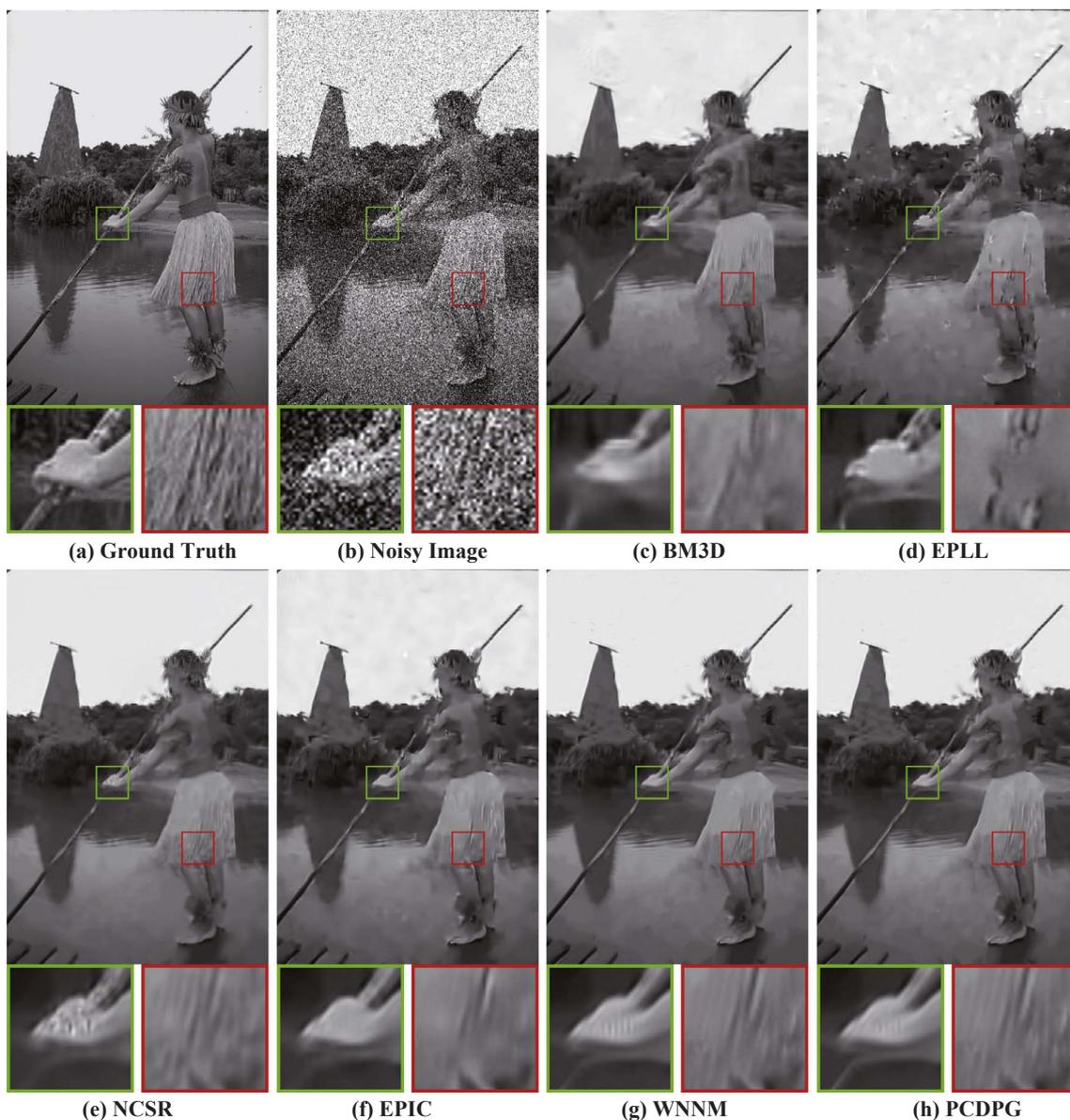


Fig. 11. Denoising results on image from BSD dataset by different methods (noise level  $\sigma_n = 50$ ). (a) Ground truth. (b) Noisy image. (c) BM3D, PSNR = 26.70 dB. (d) EPLL, PSNR = 26.84 dB. (e) NCSR, PSNR = 26.61 dB. (f) EPIC, PSNR = 27.03 dB. (g) WNNM, PSNR = 26.90 dB. (h) PCDPG, PSNR = 27.00 dB. The figure is better viewed in zoomed PDF.

```

10:     Compute the latent matrix  $\hat{Z}_i^t$  from  $Y_i^t$  via Eq. (13);
11:   end for
12:   Average the overlapped recovered patches to reconstruct the
    estimated images  $x^t$ ;
13: end for

```

#### 4. Experimental evaluation

In the ensuing experiments, our algorithm is called PCD-based patch grouping (PCDPG) for image denoising. We compared our algorithm with state-of-the-art modern image denoising techniques, including BM3D [21], NCSR [19], expected patch log likelihood (EPLL) [34], external prior-guided internal clustering (EPIC) [28], and weighted nuclear norm minimization (WNNM) [23]. The codes of these algorithms were obtained from the websites of the authors and the default parameter settings were used.

In the model learning stage, there are 3 parameters:  $p, K, m$ . The patch size ( $p \times p$ ) was set as  $p = (6, 7, 8, 9)$ , the number ( $K$ ) of

principal component subspaces as  $K = (5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100)$ , the number ( $m$ ) of principal components as  $m = p$ . We extracted about 10 million patches from about 200 training images in the BSD to learn the PCD model. In the denoising stage, the parameter values of  $p$  were set as  $p = 6$  for  $0 < \sigma_n \leq 20$ ,  $p = 7$  for  $20 < \sigma_n \leq 40$ ,  $p = 8$  for  $40 < \sigma_n \leq 60$ , and  $p = 9$  for  $60 < \sigma_n \leq 100$ . The parameter values of  $\eta$  were set as  $\eta = 0.54$  for  $0 < \sigma_n \leq 20$ ,  $\eta = 0.56$  for  $20 < \sigma_n \leq 40$ ,  $\eta = 0.58$  for  $40 < \sigma_n \leq 100$ . The other parameter settings were the same as WNNM.

##### 4.1. Denoising on commonly used images

We validated the performance of the proposed denoising algorithm on 16 commonly used images. White Gaussian noise with zero mean and standard deviation  $\sigma_n = (30, 40, 50, 60, 75, 100)$  was added to the clean images to evaluate the performances.

We analyze the suitable value of parameter  $K$  for each noise level  $\sigma_n$ . In Fig. 7, the vertical coordinate of each subfigure denotes the averaged value of peak signal-to-noise ratio (PSNR) at a certain noise level, and the horizontal coordinate denotes the value of parameter  $K$ . As can be

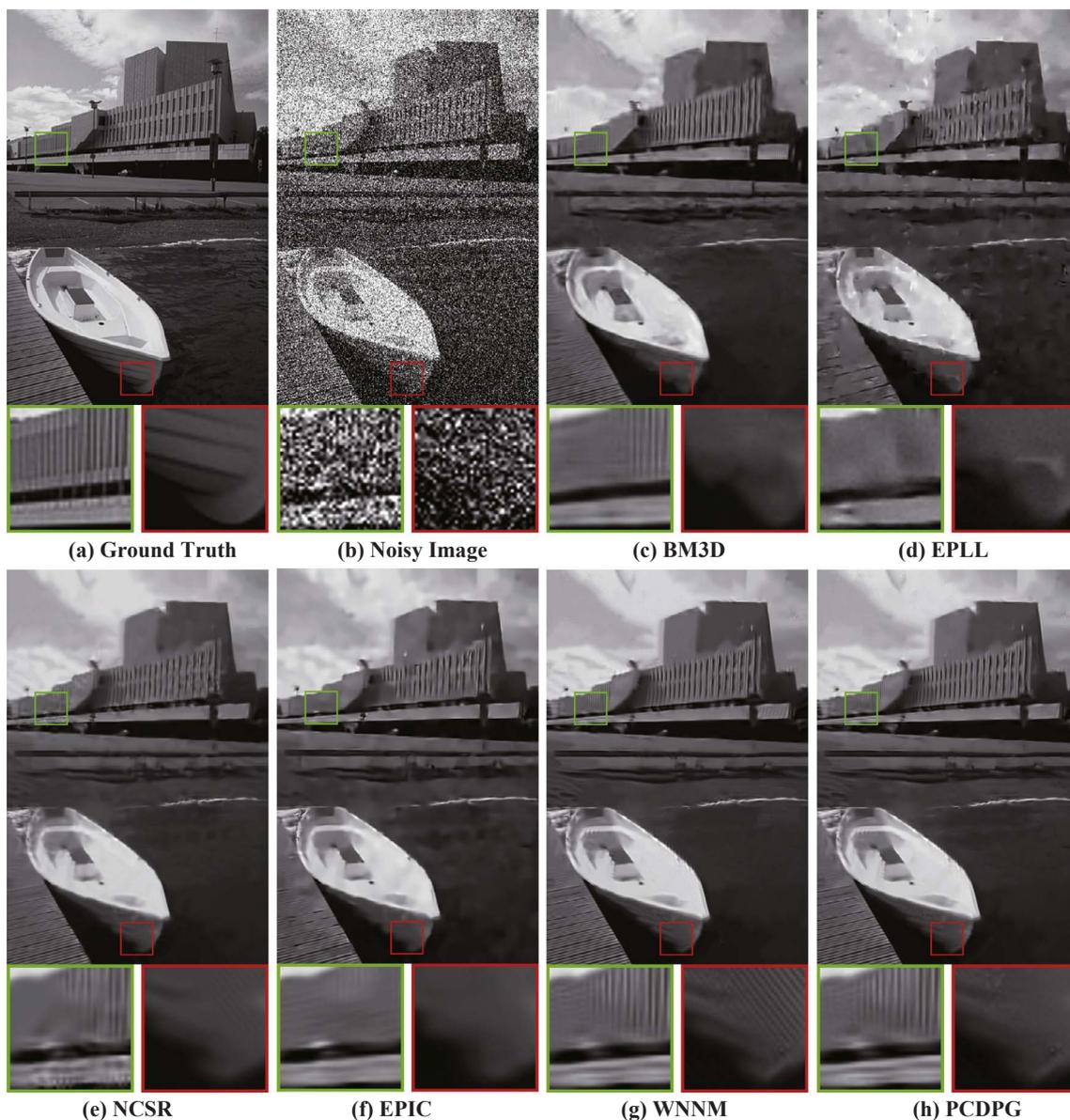


Fig. 12. Denoising results on image from BSD dataset by different methods (noise level  $\sigma_n = 75$ ). (a) Ground truth. (b) Noisy image. (c) BM3D, PSNR = 25.12 dB. (d) EPLL, PSNR = 24.30 dB. (e) NCSR, PSNR = 24.90 dB. (f) EPIC, PSNR = 25.13 dB. (g) WNNM, PSNR = 25.64 dB. (h) PCDPG, PSNR = 25.76 dB. The figure is better viewed in zoomed PDF.

Table 3

Average denoising results (PSNR (dB)) by CBM3D and our algorithm on 68 randomly selected color images from the BSD under several noise levels.

$\sigma_n$	10	20	30	40
CBM3D	35.82	31.78	29.57	27.90
PCDPG	<b>35.88</b>	<b>32.00</b>	<b>29.84</b>	<b>28.45</b>

seen, the best value of parameter  $K$  can be set as 10. Even through the best value of parameter  $K$  are separately 5 and 30, when  $\sigma_n$  is set to be 50 and 100. The PSNR results only change in a slight range with different values of  $K$ . Table 1 reports the PSNR performance of seven competing algorithms. It is clear from Table 1 that our proposed algorithm achieved the highest PSNR in almost all cases. The PSNR performance of our proposed algorithm is an improvement over WNNM from 0.03 dB to 0.06 dB to 0.09 dB on average, when the noise deviation increases from 30 to 60 to 100. Our algorithm outperformed all the other methods. The improvement became more significant with

increasing noise level.

The visual quality of denoised images is critical when evaluating a denoising algorithm. Figs. 8–10 show the denoised images of *House*, *Peppers*, and *Straw* by the respective competing algorithms. It is clear that EPLL and EPIC do not handle smooth areas well; further, WNNM is likely to generate artifacts when noise is strong and BM3D and NCSR slightly over-smooth the image. Our proposed denoising method is more robust than those other methods and better preserves structures. For example, for *House*, our algorithm produces a much clearer edge with fewer artifacts. For *Peppers*, our algorithm recovers the interspaces between peppers and the textures on peppers clearer than all the competing methods, including WNNM. For *Straw*, the structures of the straws are reconstructed more faithfully by our algorithm.

#### 4.2. Denoising the BSD test dataset

We validated the performance of our proposed denoising algorithm on 40 images randomly selecting from the BSD test dataset. The value of  $K$  was set to 10, same as in the above experiment. Table 2 shows the PSNR performance for the competing denoising methods. It is clear that

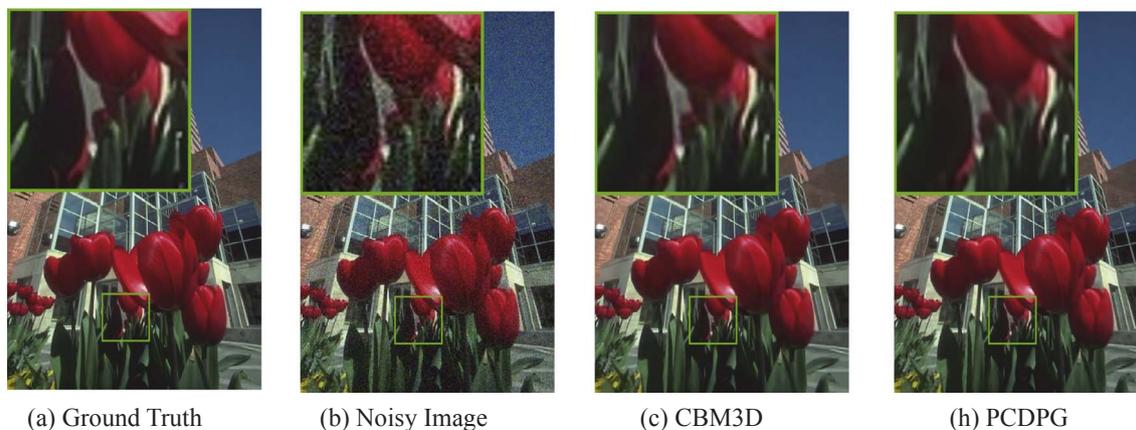


Fig. 13. Denoising results on color image from BSD dataset by different methods (noise level  $\sigma_n = 20$ ). (a) Ground truth. (b) Noisy image. (c) CBM3D, PSNR = 32.33 dB. (d) PCDPG, PSNR = 32.80 dB. The figure is better viewed in zoomed PDF. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

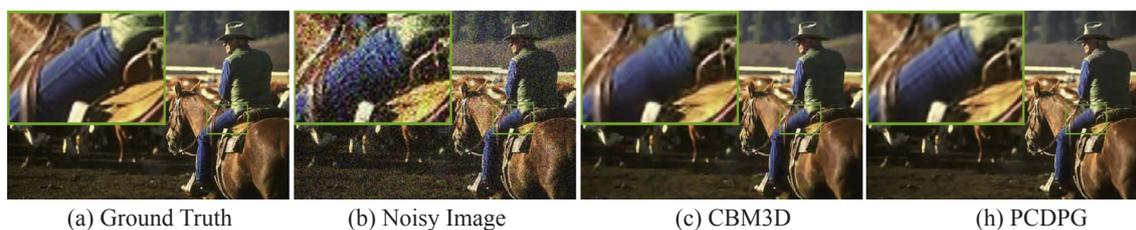


Fig. 14. Denoising results on color image from BSD dataset by different methods (noise level  $\sigma_n = 40$ ). (a) Ground truth. (b) Noisy image. (c) CBM3D, PSNR = 28.95 dB. (d) PCDPG, PSNR = 29.54 dB. The figure is better viewed in zoomed PDF. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

our algorithm performed best on all the cases of eight noise levels ( $\sigma_n = 10, 20, 30, 40, 50, 60, 75, 100$ ). Our algorithm is better than WNNM from 0.06 dB to 0.1 dB on average, when the noise deviation increases from 10 to 100. Figs. 11 and 12 show examples from BSD test dataset. In Fig. 11, our algorithm recovers the structures of hand and skirt more faithfully than the other algorithms. In Fig. 12, the textures of the boat and the buildings are restored better with our algorithm than with the others.

We also compared our proposed algorithm with the color block-matching and 3D filtering (CBM3D) method [35] on color images from BSD test dataset. When our algorithm was applied to color images, the intensity data from 3 channels at a local patch was processed as a vector. The parameter value of  $\eta$  was set as  $\eta = 0.65$ . Table 3 shows the performance for CBM3D and our algorithm on color images under several noise levels. Our algorithm is better than CBM3D from 0.06 dB to 0.55 dB on average, when the noise deviation increases from 10 to 40. Figs. 13 and 14 show examples of color images from the BSD test dataset. In Fig. 13, the proposed denoising algorithm recovers the flowers more clearly than CBM3D. In Fig. 14, the textures on the jeans are restored better with our algorithm. In summary, our proposed algorithm outperformed the other methods both qualitatively and quantitatively. The stability of our proposed similar patches grouping method enhances the performance of the denoising algorithm, especially with heavy noise.

## 5. Conclusion

In this paper, we proposed PCD-based patch grouping for image denoising. We learned PCD from clean natural images and used it to guide similar patch grouping. In the proposed patch grouping scheme, noisy patches are projected onto PCD-based transform-domain to estimate their principal components, and grouped with their estimated principal components, which are robust to noise. Therefore, the PCD-based scheme enhances the patch grouping results, especially with heavy noise. The proposed patches grouping scheme enhances

denoising performance. We combined the PCD-based patch grouping method and a row-rank approximation process in a proposed denoising algorithm. External knowledge from clean natural images and internal NSS prior are used jointly in the proposed algorithm. Experimental results indicate that our denoising method outperforms many state-of-the-art algorithms, both qualitatively and quantitatively. The proposed patches grouping scheme also can be applied to other image restoration tasks, such as deblurring, super-resolution and so on.

## Acknowledgements

The authors would like to thank editor and anonymous reviewers who gave valuable suggestion that has helped to improve the quality of the paper. This work is supported by National Natural Science Foundation of China (NSFC) grant (61571207) and National Key Laboratory of Science and Technology on Multispectral Information Processing, School of Automation, Huazhong University of Science and Technology, China.

## References

- [1] D.L. Donoho, De-noising by soft-thresholding, *IEEE Trans. Inf. Theory* 41 (3) (1995) 613–627.
- [2] S.G. Chang, B. Yu, M. Vetterli, Adaptive wavelet thresholding for image denoising and compression, *IEEE Trans. Image Process.* 9 (9) (2000) 1532–1546.
- [3] J.L. Starck, E.J. Candes, D.L. Donoho, The curvelet transform for image denoising, *IEEE Trans. Image Process.* 11 (6) (2002) 670–684.
- [4] J. Portilla, V. Strela, M. Wainwright, E. Simoncelli, Image denoising using scale mixtures of Gaussians in the wavelet domain, *IEEE Trans. Image Process.* 12 (11) (2003) 1338–1351.
- [5] L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D* 60 (1) (1992) 259–268.
- [6] S. Osher, M. Burger, D. Goldfarb, J. Xu, W. Yin, An iterative regularization method for total variation-based image restoration, *Multiscale Model. Simul.* 4 (2) (2005) 460–489.
- [7] M. Elad, M. Aharon, Image denoising via sparse and redundant representations over learned dictionaries, *IEEE Trans. Image Process.* 15 (12) (2006) 3736–3745.
- [8] J. Mairal, M. Elad, G. Sapiro, Sparse representation for color image restoration, *IEEE Trans. Image Process.* 17 (1) (2008) 53–69.
- [9] J. Cai, E. Candes, Z. Shen, A singular value thresholding algorithm for matrix

- completion, *SIAM J. Optim.* 20 (4) (2010) 1956–1982.
- [10] E.J. Candes, Y. Plan, Matrix completion with noise, in: *Proceedings of the IEEE*, vol. 98, 2010, pp. 925–936.
- [11] A. Rajwade, A. Rangarajan, A. Banerjee, Image denoising using the higher order singular value decomposition, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (4) (2013) 849–862.
- [12] L. Rudin, S. Osher, E. Fatemi, Nonlinear total variation based noise removal algorithms, *Physica D* 60 (1) (1992) 259–268.
- [13] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1998, pp. 839–846.
- [14] L. Zhang, R. Lukac, X. Wu, D. Zhang, Pca-based spatially adaptive denoising of cfa images for single-sensor digital cameras, *IEEE Trans. Image Process.* 18 (4) (2009) 797–812.
- [15] M. Zontak, M. Irani, Internal statistics of a single natural image, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011, pp. 977–984.
- [16] Y. Chang, L. Yan, H. Fang, C. Luo, Anisotropic spectral-spatial total variation model for multispectral remote sensing image destriping, *IEEE Trans. Image Process.* 24 (6) (2015) 1852–1866.
- [17] Y. Chang, L. Yan, S. Zhong, Hyper-Laplacian regularized unidirectional low-rank tensor recovery for multispectral image denoising, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 4260–4268.
- [18] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Non-local sparse models for image restoration, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 2272–2279.
- [19] W. Dong, L. Zhang, G. Shi, X. Li, Nonlocally centralized sparse representation for image restoration, *IEEE Trans. Image Process.* 22 (4) (2013) 1620–1630.
- [20] A. Buades, B. Coll, J.M. Morel, A non-local algorithm for image denoising, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2, 2005, pp. 60–65.
- [21] 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.
- [22] H. Ji, C. Liu, Z. Shen, Y. Xu, Robust video denoising using low rank matrix completion, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 1791–1798.
- [23] S. Gu, L. Zhang, W. Zuo, X. Feng, Weighted nuclear norm minimization with application to image denoising, in: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 2862–2869.
- [24] S. Grewenig, S. Zimmer, J. Weickert, Rotationally invariant similarity measures for nonlocal image denoising, *J. Vis. Commun. Image Represent.* 22 (2011) 117–130.
- [25] S. Wang, Y. Xia, Q. Liu, J. Luo, Y. Zhu, D.D. Feng, Gabor feature based nonlocal means filter for textured image denoising, *J. Vis. Commun. Image Represent.* 23 (2012) 1008–1018.
- [26] F. Chen, X. Zeng, M. Wang, Image denoising via local and nonlocal circulant similarity, *J. Vis. Commun. Image Represent.* 30 (2015) 117–124.
- [27] M. Nejati, S. Samavi, H. Derksen, K. Najarian, Denoising by low-rank and sparse representations, *J. Vis. Commun. Image Represent.* 36 (2016) 28–39.
- [28] F. Chen, L. Zhang, H. Yu, External patch prior guided internal clustering for image denoising, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 603–611.
- [29] L. Zhang, W. Dong, D. Zhang, G. Shi, Two-stage image denoising by principal component analysis with local pixel grouping, *Pattern Recogn.* 43 (2010) 1531–1549.
- [30] C.M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.
- [31] W. Dong, L. Zhang, G.S.X. Wu, Image deblurring and superresolution by adaptive sparse domain selection and adaptive regularization, *IEEE Trans. Image Process.* 20 (7) (2011) 1838–1857.
- [32] G.E. Hinton, P. Dayan, M. Revow, Modelling the manifolds of images of hand-written digits, *IEEE Trans. Neural Networks* 8 (7) (1997) 65–74.
- [33] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, vol. 2, 2001, pp. 416–423.
- [34] D. Zoran, Y. Weiss, From learning models of natural image patches to whole image restoration, in: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011, pp. 479–486.
- [35] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian, Color image de-noising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space, in: *Proceedings of the IEEE International Conference Image Processing*, 2007, pp. 1-313–1-316.