SCIENCE CHINA
Information Sciences

• Supplementary File •

Nonlocal image denoising using edge-based similarity metric and adaptive parameter selection

Linwei FAN [1], Xuemei LI [1], Qiang GUO [2,3] & Caiming ZHANG [1,3*]

[1]*School of Computer Science and Technology, Shandong University, Jinan* 250101*, China;*
[2]*School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan* 250014*, China;*
[3]*Shandong Provincial Key Laboratory of Digital Media Technology, Jinan* 250014*, China*

Appendix A Design of the anti-noise edge detection operator

Eight templates can be defined by a value function $F = z(x, y)$, namely, the weight values are the sampling points of the function. Hence, it is vital to construct the value function F, the function which satisfies that the center value is the biggest and the value strictly decreases with the farther away from the center. The function F can be constructed by the analysis of 25 sample values of 0° template. Due to symmetry, the other direction templates can be obtained by rotating the value function F. For the convenience, the lower-left coordinate of the 0° template is assumed to be (0, 0), the upper-right coordinate is (4, 4). The *a, b, c* and *d* of 0° template must statisfy the following conditions: (1) In the case of a smooth signal, we expect each column to produce zero when difference in the direction of columns. Therefore, the values of middle line are zero; (2) Distribution weight values according to the distance from the center, let the same distance have identical weights; (3) In order to find local differences, we should subtract the signal "in front" of the center pixel from the signal "behind" it.

Due to the correlation between weight values, we set the weight value with maximum distance to 1. Finally, we obtain the 0° template, as shown in the Figure A1 (a). The 90° template can be produced from the 0° template with a 90° rotation, as shown in Figure A1 (b). Following , the relationship between *a, b, c* and *d* are discussed. The idea is trying to specify parameters of the filter in Figure A1, so that, the output of the operator is as faithful as possible to the true values of α which correspond to the non-discretized image in Figure A2. Edges are characterized by their orientation, defined as:

$$\alpha(i, j) = \tan^{-1} \frac{\Delta f_y(i, j)}{\Delta f_x(i, j)}$$

-1	-a	-b	-a	-1
-a	-d	-c	-d	-a
0	0	0	0	0
a	d	c	d	a
1	a	b	a	1

(a)

-1	-a	0	a	-1
-a	-d	0	d	a
-b	-c	0	c	b
-a	-d	0	d	a
-1	-a	0	a	1

(b)

Figure A1 The weight values of 0° and 90° template: (a) 0°; (b)90°.

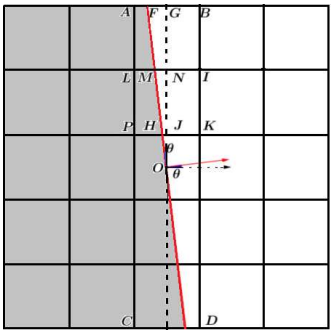
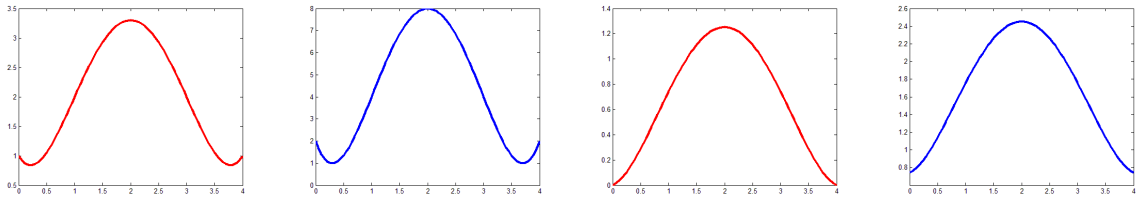


Figure A2 Zooming into a 5×5 patch of an image around pixel (i, j). (The orientation of the edge, is measured from the horizontal axis anticlockwise.)

Consider a straight step edge in the scene, passing through the middle of a pixel [1]. Each pixel is assumed to be a tile of size 1×1. Assume that the edge orientation θ is small enough so that the edge cuts lines AB and CD (see Figure A2). Also,



(a) (b) (c) (d)

Figure A3 Lagrangian interpolation curves.

assume that a dark region with grey value G1 on the left of the edge and a bright region with G2 on the right. Clearly, the pixels in the central column have mixed values. If we assume S_{LMHP} be the area of a polygon LMHP, then the gray-value of the pixel that is formed by the polygon LIKP is $f(i, j-1)$:

$$f(i,j-1) = G_1(S_{LMPH}) + G_2(S_{MIHK}) = G_1[\frac{1}{2} - (S_{MNHJ})] + G_2[\frac{1}{2} + (S_{MNHJ})] = G_1(\frac{1}{2} - \tan\theta) + G_2(\frac{1}{2} + \tan\theta)$$

Similarly, we can also obtain: $f(i,j+1) = G_2(\frac{1}{2} - \tan\theta) + G_1(\frac{1}{2} + \tan\theta)$; $f(i,j-2) = G_1(\frac{1}{2} - 2\tan\theta) + G_2(\frac{1}{2} + 2\tan\theta)$; $f(i,j+2) = G_2(\frac{1}{2} - 2\tan\theta) + G_1(\frac{1}{2} + 2\tan\theta)$. On the base of the template, we introduce the notion of edge orientation. The 90° template calculates Δf_x and the 0° template calculates Δf_y. The horizontal and vertical differences are obtained by template convolution between two templates in Figure A1 and 5×5 image patches in Figure A2.

$$\tan\alpha = \frac{\Delta f_{y1}}{\Delta f_{x1}} = \frac{(4b+2c)(G_2-G_1)\tan\theta}{(2+4a+2d+b+c)(G_2-G_1)} = \frac{4b+2c}{2+4a+2d+b+c}\tan\theta \tag{A1}$$

Similarly, we can calculate the orientation of the edge in a 3×3 template:

$$\tan\alpha = \frac{\Delta f_{y2}}{\Delta f_{x2}} = \frac{2c}{2d+c}\tan\theta \tag{A2}$$

Note that the template parameters a, b, c and d in Figure A1 should be chosen appropriately, so that the calculated orientation α of the edge is equal to the true orientation θ. For these reasons, we set coefficients of Eq. (A1) and Eq. (A2) to 1. Therefore the weights shall meet the following relationships:

$$\begin{cases} b = \frac{4}{3}a + \frac{2}{3} \\ c = 2d \end{cases} \tag{A3}$$

From Eq. (A3), we get only two independent variables a and d which are needed for discussion, at this time, the weights of 0° template are as follows:

We hope that select the unknown parameters a and d to make a double quartic Lagrangian interpolation value function F, which can generate the rest templates. No matter what the value of a and d, the two quartic polynomial curves are constructed by the weights of lines 3 and 4 in Figure A3 (a)-(b). The function F is nonconvex for not strictly decreasing on both sides of the maximum, while the peak point lies at the central location. Therefore, the convex interpolation function F cannot be constructed from the 25 weights directly. To satisfy the convexity, the logarithms of 25 weights were taken as input to get the ideal curves by adjusting the unknown parameters a and d (see Figure A3 (c)-(d)). The Lagrangian interpolation function of lines 3 and 4 in 0° template as shown in Eq. (A4) and Eq. (A5) respectively:

$$f_1(x) = l_2(x)\ln a + l_3(x)\ln(\frac{4}{3}a + \frac{2}{3}) + l_4(x)\ln a \tag{A4}$$

$$f_2(x) = l_1(x)\ln a + l_2(x)\ln d + l_3(x)\ln(2d) + l_4(x)\ln d + l_5(x)\ln a \tag{A5}$$

where $l_n(x)$ is basic function defined in 0, 1, 2, 3, 4 independently, satisfying $l_n(n) = 1$.

Since the convex function satisfies the necessary condition is that the first derivative at the zero point should greater than zero, so we get the derivatives at the zero points to meet the condition.

$$\begin{cases} f_1'(0) = \ln \frac{a^{\frac{64}{3}}}{(\frac{4}{3}a+\frac{2}{3})^{12}} > 0 \\ f_2'(0) = \ln \frac{a^{-\frac{28}{3}} d^{\frac{28}{3}}}{2^{12}} > 0 \end{cases} \tag{A6}$$

Experiments proved $f'(0)$ equal to one gives better results and the weights in template are integers. Thus, we obtain the 0° template defined by the value function F after rounded and the 8-directional templates are shown in Figure A4.

Appendix B Experimental Results

Appendix B.1 Parameter of search-window S setting

Our method needs to set some parameters, including the size of similarity-window and search-window to balance the effect of denoising. The first stage uses the weighted version of all patches in a search-window to obtain the central patch. In order

* Corresponding author (email: czhang@sdu.edu.cn)

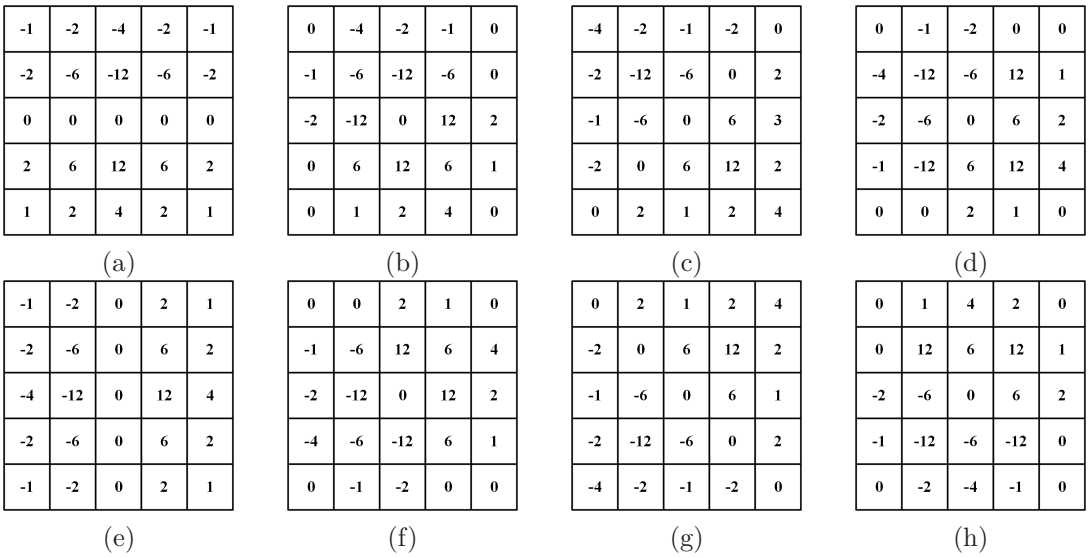


Figure A4 8-directional templates: (a) 0° direction; (b) 22.5° direction; (c) 45° direction; (d) 67.5° direction; (e) 90° direction; (f) 112.5° direction; (g) 135° direction; (h) 157.5° direction.

to improve the denoising performance, we should search for similar patches as much as possible, so the search-window size is also affected. Most of the nonlocal denoising methods such as [2, 3] select a fixed-size search-window, while [9] proposes an adaptive search-window method based on the estimation of the error and the variance analysis. In this subsection, we discuss how the size of search-window will affect the denoising performance based on image patches.

Figure B1 shows the relationship between the PSNR of our method under $\sigma = 30$ and the search-window size. Figure B2 gives a comparison of the results of the *Barbara*'s scarf section with different search-window sizes. From Figure B1 and Figure B2, the performance of our method is improved with the increasement of the search-window size and then is stabilized at about size 13×13.

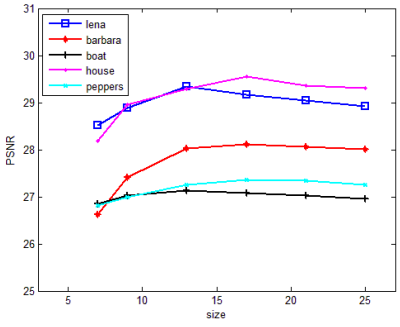


Figure B1 The relationship between denoising performance and the size of search-window.

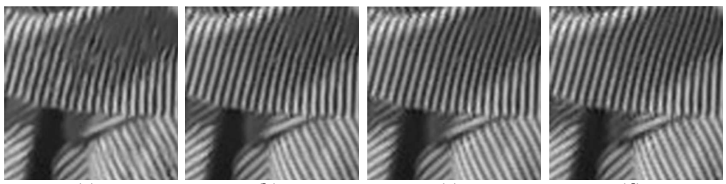


Figure B2 Performances for different patch sizes when the method is applied to the noisy *Barbara* image($\sigma = 30$): (a) 7×7 patch (PSNR=27.63); (b) 9×9 patch (PSNR=27.82); (c) 13×13 patch (PSNR=28.12); (d) 17×17 patch (PSNR=28.03).

In general, with the increasing noise level, the search-window S_i should become large for finding more similar pixels. We empirically find that the best search-window size is 11×11, 13×13, 21×21 for $\sigma = 20, 30, 50$, respectively. In the following experiments, the similarity-window is fixed at size 3×3 and the search-windows in other comparison algorithms are 17×17 while our algorithm use an adaptive search-window.

Appendix B.2 Denoising results

In this subsection, we use 7 standard images (*Boat*, *Lena*, *Monarch*, *Barbara*, *C. man*, *House*, *Peppers*) as test images. By setting the AWGN standard deviation $\sigma \in \{20, 30, 50\}$, we validate the performance of the proposed scheme and compare it with state-of-the-art NLM-based methods, including the spatial domain methods (NLM [2], PND [3], NLM-BDPCA [6], NLM-SVB [7]) and hybrid methods (K-SVD [5], BM3D [4], SAIST [8]).

We first compare the denoised results with spatial domain methods. As shown in Figure B3, the visual quality of the results by the proposed methods outperform those spatial domain methods. Figure B3 (a) illustrates the results for *Boat* image, which is corrupted by the noise with $\sigma = 50$. It can be observed that NLM-BDPCA and our method are better in

denoising and preserving details than other methods. However, our method gives the better denoising result than NLM-BDPCA in keeping details, especially for the text portion of the hull. It maintains more details than the other methods, without over-smoothing at the edge of the wings of *Monarch* in Figure B3 (b), which gains the denoised results much closer to the ground truth image. To further validate the efficiency of our method in keeping the details, we apply the above six algorithms to filter the *Barbara* image with delicate texture, and the results are shown in Figure B3 (c). The denoised image by our method does not have obvious fuzzy phenomenon and distinctly maintains more image details, such as the texture details on the pants and scarf. In contrast, the denoised images by the other four algorithms all have the ambiguous phenomena. We also compare our denoising results with hybrid denoising methods, including K-SVD [5], BM3D [4] and SAIST [8]. Figure B4 shows the examples for *House* and *C. man* image with noise level $\sigma = 35$. The visual quality by our method outforms K-SVD and is similar to BM3D. In Figure B5, we further provide a comparison of some enlarged parts of the denoising results for *Boat*, *House* and *Peppers* images based on different methods for $\sigma = 25$. It is obvious that our method is more accurate in detail preservation, and closely to BM3D.

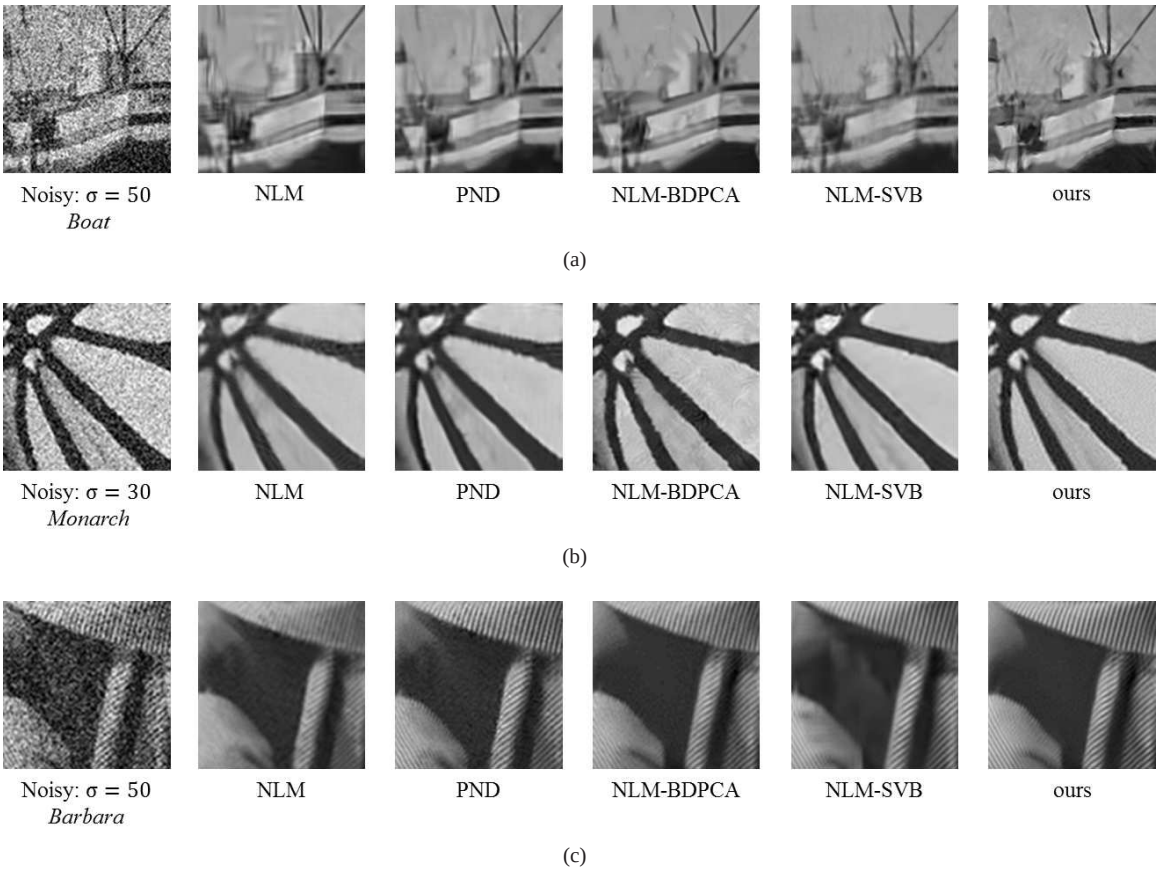


Figure B3 Enlarged parts of image denoising results by spatial domain methods: (a) denoising with noise level $\sigma = 50$; (b) denoising with noise level $\sigma = 30$; (c) denoising with noise level $\sigma = 50$.

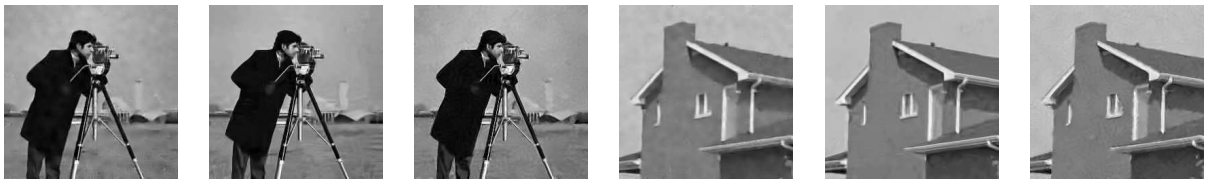


Figure B4 *C. man* and *House* denoised by hybrid methods (K-SVD, BM3D) and our method: (a) K-SVD; (b) BM3D; (c) ours; (d) K-SVD; (e) BM3D; (f) ours.

For quantitative comparisons, the PSNR and SSIM metrics are used as the evaluation criterion. Table B1 lists the quantitative results of competing algorithms. The best denoising results on every piece of test images under the same noise levels are bold to obvious comparison. From Table B1, we know the PSNR and SSIM of our proposed method are mainly

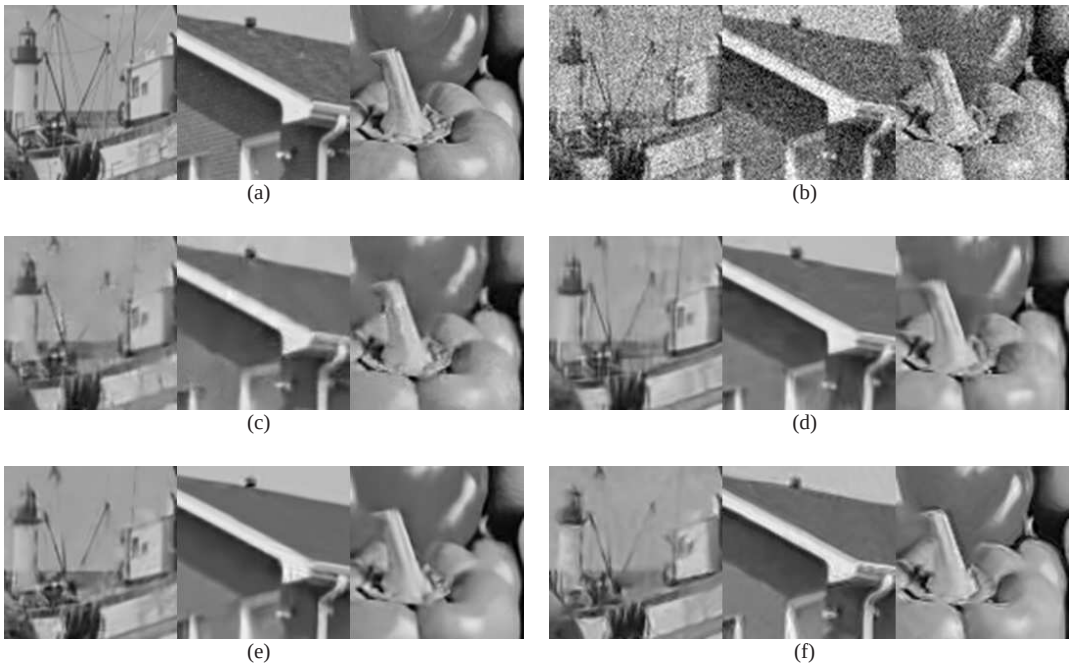


Figure B5 Enlarged parts of image denoising results (from left to right: *Boat*, *House* and *Peppers*): (a) original image; (b)noisy image; (c)denoised by K-SVD; (d) denoised by BM3D; (e)denoised by SAIST; and (f)denoised by our method.

Table B1 Denoising PSNR and SSIM results by different denoising methods

Image	σ	Spatial Domain								Spatial + Transform Domain						ours	
		NLM		PND		NLM-BDPCA		NLM-SVB		K-SVD		BM3D		SAIST			
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Boat	20	29.10	0.751	29.68	0.772	29.81	0.802	29.72	0.790	30.37	0.863	30.54	0.878	**30.81**	**0.904**	29.98	0.813
	30	27.16	0.703	27.66	0.731	28.02	0.763	27.94	0.757	27.93	0.801	28.55	0.829	**28.93**	**0.856**	28.10	0.787
	50	24.40	0.624	25.63	0.674	25.87	0.711	25.31	0.697	25.38	0.710	26.19	0.758	**26.66**	**0.772**	25.93	0.722
Lena	20	29.53	0.811	29.94	0.837	30.32	0.854	30.16	0.848	31.42	0.882	32.13	0.890	**33.08**	**0.906**	30.46	0.851
	30	28.45	0.720	29.21	0.740	29.42	0.783	29.13	0.760	29.86	0.832	30.13	0.847	**31.27**	**0.864**	28.61	0.795
	50	25.81	0.635	26.48	0.655	26.96	0.691	26.84	0.673	27.79	0.745	28.86	0.760	**29.01**	**0.794**	27.02	0.690
Barbara	20	28.82	0.817	29.22	0.842	29.39	0.870	29.34	0.864	30.71	0.878	31.74	0.923	**32.10**	**0.941**	29.51	0.872
	30	27.26	0.762	27.75	0.790	28.03	0.822	27.94	0.818	28.36	0.815	29.77	0.872	**30.09**	**0.902**	28.12	0.833
	50	24.54	0.630	25.14	0.663	25.34	0.687	25.21	0.664	25.47	0.698	27.17	0.828	**27.54**	**0.845**	25.50	0.688
C. man	20	27.94	0.830	28.47	0.816	28.69	0.823	28.72	0.830	29.97	0.863	30.43	0.873	**30.50**	**0.896**	28.88	0.839
	30	25.36	0.725	27.79	0.755	28.05	0.771	27.91	0.758	27.95	0.813	28.64	0.836	**28.96**	**0.861**	28.26	0.771
	50	23.35	0.687	24.36	0.716	25.08	0.723	25.31	0.726	25.35	0.739	26.09	0.781	**26.19**	**0.799**	25.72	0.734
House	20	30.46	0.783	32.03	0.791	32.32	0.832	32.45	0.845	33.02	0.859	33.85	0.871	**33.90**	**0.896**	32.76	0.855
	30	29.62	0.76	30.30	0.782	30.65	0.805	30.38	0.776	30.82	0.829	32.23	0.847	**32.39**	**0.872**	30.53	0.792
	50	26.40	0.675	28.10	0.703	28.56	0.722	28.52	0.731	27.95	0.763	29.37	0.803	**30.20**	**0.829**	28.70	0.766
Peppers	20	28.64	0.791	29.34	0.827	30.11	0.856	30.06	0.843	30.68	0.875	31.39	0.903	**31.69**	**0.926**	30.28	0.872
	30	27.90	0.756	28.26	0.774	28.73	0.803	28.43	0.788	28.70	0.836	29.33	0.868	**29.60**	**0.891**	28.67	0.801
	50	24.49	0.544	26.10	0.631	27.03	0.702	26.89	0.689	26.90	0.769	27.27	0.810	**27.76**	**0.834**	27.02	0.781
Monarch	20	27.97	0.867	28.64	0.889	28.85	0.895	28.75	0.891	29.82	0.902	30.39	0.921	**30.81**	**0.945**	29.00	0.902
	30	26.64	0.741	28.01	0.783	28.42	0.837	28.38	0.826	28.80	0.867	29.38	0.887	**29.68**	**0.903**	28.51	0.845
	50	22.89	0.696	23.66	0.751	24.05	0.779	24.15	0.782	25.18	0.795	25.64	0.824	**25.89**	**0.846**	25.39	0.799

higher than the other spatial domain methods. With the increasement of noise standard deviation, the denoising effect of our method is much better. However, because the NLM-BDPCA can make full use of the spatial structure of the image, the PSNR is higher than our algorithm when the image with high spatial structure similarity, e.g. *House* image and *Peppers*

image. Besides, our method is also implemented for fair comparison with the methods K-SVD [5], BM3D [4], SAIST [8]. These methods, denoted as hybrid methods. From Table B1, we can also see that the superiority of our method exists in some cases with respect to K-SVD. For example, for *House* image, the PSNR gains of our method over K-SVD is 0.75, when $\sigma = 50$. Though the results of BM3D and SAIST are always the best, our method is close to BM3D when $\sigma = 50$ for images *Monarch* and *Peppers*. Since the main goal of this paper is to offer an improvement to the spatial filters, we do not claim to offer a better PSNR than hybrid method.

References

1 Petrou M, Petrou C. Image Processing: The fundamentals. 2nd ed. Chichester: John Wiley and Sons, 2010.
2 Buades A, Coll B, Morel J M. A review of image denoising algorithms, with a new one. Multiscale Modeling and Simulation, 2005, 4(2): 490-530
3 Tasdizen T. Principal neighborhood dictionaries for nonlocal means image denoising. IEEE Trans. Image Process., 2009, 18(12): 2649-2660
4 Dabov K, Foi A, Katkovnik V, et al. Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process., 2007, 16(8): 2080-2095
5 Elad M, Aharon M. Image denoising via sparse and redundant representations over learned dictionaries. IEEE Trans. Image Process., 2006, 15(12): 3736-3745
6 Chen H H, Ding J J. Nonlocal means image denoising based on bidirectional principal component analysis. In: Proceedings of ICASSP, 2015:1265-1269
7 Sharifymoghaddam M, Beheshti S, Elahi P, et al. Similarity validation based nonlocal means image denoising. IEEE Signal Process. Letters, 2015, 22(12): 2185-2188
8 Dong W, Shi G, Li X. Nonlocal image restoration with bilateral variance estimation: a low-rank approach. IEEE Trans. Image Process., 2013, 22(2): 700-711
9 Kervrann C, Boulanger J. Optimal spatial adaptation for patch-based image denoising. IEEE Trans. Image Process., 2006, 15(10): 2866-2878