PATCH-BASED BILATERAL FILTER: ALGORITHMS AND PERFORMANCE

Hisashi Shimodaira

Home address: 2-2-16 Katsuradai, Aoba-Ku, Yokohama-City, 227-0034
Japan

ABSTRACT

The bilateral filter has a weakness in its denoising ability, because the intensity similarity is evaluated by comparing just the intensity values of a given pixel and another pixel. To alleviate this problem, this paper proposes a patch-based bilateral (PBL) filter in which the intensity similarity is evaluated by comparing a whole patch around each pixel. Experimental results for real images show that this method outperforms the bilateral filter in all cases, is comparable to the nonlocal-(NL-)means filter, and outperforms the more elaborate variants of the NL-means filter in some cases.

Index Terms — Bilateral filter, edge-preserving denoising, image denoising, intensity similarity, nonlocal means filter.

1. INTRODUCTION

Noise can be introduced into a digital image in many ways, and thus reducing noise without degrading the underlining image has recently attracted much attention. If the noise in digital images is additive and white, it can be reduced by averaging the intensity values of neighboring pixels that are similar to that of a given pixel. In the averaging process, pixels with intensity values more similar to that of the given pixel are given larger weights. Since the original intensity value is lost, these filters use various methods for selecting pixels with a similar intensity value in the neighborhood of each pixel. We shall refer to filters that use this strategy as neighborhood filters.

The bilateral (BL) filter proposed in [1], [2] is a kind of neighborhood filter in which the weight is evaluated by combining the terms for a pixel's spatial closeness and intensity similarity. The spatial closeness term is designed so that pixels spatially closer to a given pixel are given larger weights. The intensity similarity is evaluated by only the difference between the intensity value of a given pixel and that of the other pixel in the window. Since its algorithm is simple and its denoising performance is relatively high, it has been popular and widely used.

Several papers [3]–[5] have pointed out the problem with the structure-preserving ability of the BL filter when the variations in intensity of the underlining image are of the order of or less than the noise in the image. In such a case, the filter inevitably blunts or blurs the image structure because it cannot separate the noise from the image structures. Some papers have proposed methods to alleviate this problem, for example, based on linear regression [3] and the gradient of the image intensity [4].

Furthermore, many papers have proposed variants of the BL filter and ways to combine it with other methods [6]–[9]. For example, one proposal [9] is to use BL filters with different strategies and parameters according to the edge and non-edge regions. Another [6] is to apply the BL filter to the signal decomposed by a wavelet filter bank. Another [8] is first to perform a wavelet-based denoising and then use the BL filter to remove artifacts produced by the wavelet method. On the other hand, several attempts [10]–[12] have been made to understand the BL filter theoretically and extend it to more general methods.

The pixels most similar to a given pixel are not necessarily the pixels close to it spatially. Based on this idea, [13] has extended neighborhood filters to the non-local (NL) means filter, in which the intensity similarity is evaluated by comparing the intensity values of all pixels in the two patches around a given pixel and another pixel in a relatively large fixed-size window. This leads to using image self-similarity to select pixels with similar intensity values. Spatial closeness is no longer taken into account. The NL-means filter achieves better denoising performance than the BL filter at the cost of a large increase in computation time.

Since the study in [13] was released, several variants of the NL-means filter or patch-based methods that achieve better denoising performance than the original NL-means filter have been published [14]–[16]. Lately, several papers have proposed variational approaches for the BL and NL-means filters [17]–[21]. Although these filters achieve better denoising performance, their computational complexity becomes much higher.

Against this background, we have developed the patch-based bilateral (PBL) filter by combining the ideas of the BL filter and the NL-means filter of [13]. The PBL filter uses both spatial closeness and intensity similarity of a pixel in a relatively small fixed-size window, in which the intensity similarity is evaluated by comparing the intensity values of all pixels in the two patches around a given pixel and another pixel in the window as with the NL-means filter. To the best of the author's knowledge, such a method has not been tested and reported in any previous papers. Its basic idea is to select pixels with similar intensity values effectively and economically by using the image self-similarity around pixels closer to the given pixel. Since this helps alleviate the problem of the BL filter mentioned above, this method can significantly improve the performance of the BL filter.

We conducted experiments on simulated noisy real images to investigate the performance of the proposed method. The results show that the PBL filter outperforms the BL filter in all cases, is comparable to the NL-means filter, and outperforms the more elaborate variants of the NL-means filter in some cases.

2. BL FILTER AND PBL FILTER

Consider an image of size $M \times N$ pixels, where M and N are the total number of pixels in the horizontal and vertical dimensions, respectively. Let $I(x, y)$ denote the observed intensity value of a pixel of a noisy image at a location (x, y). The intensity value of a filtered image at a given pixel (x_0, y_0) is represented by $J(x_0, y_0)$. As the neighborhood of the pixel (x_0, y_0), we use a square search window $W(x_0, y_0)$ centered at (x_0, y_0). Consider a pixel (x, y) within the window and two square patches of size $(2p+1) \times (2p+1)$ pixels centered at (x_0, y_0) and (x, y), respectively. Let (i, j) denote the relative position of a pixel within the patch from its center pixel.

In the bilateral (BL) filter [1], [2], the filtered value can be written as

$$J(x_0, y_0) = \frac{1}{K(x_0, y_0)} \sum_{(x, y) \in W(x_0, y_0)} C_g(x, y)\, C_s(x, y)\, I(x, y)\,, \tag{1}$$

$$C_g(x, y) = \exp\left(-\frac{d^2}{2\, h_g^2} \right), \tag{2}$$

$$C_s = \exp\left(-\left(\frac{I(x_0, y_0) - I(x, y)}{hs} \right)^2 \right), \tag{3}$$

and

$$K(x_0, y_0) = \sum_{(x, y) \in W(x_0, y_0)} C_g(x, y)\, C_s(x, y)\,, \tag{4}$$

where d is the Euclidean distance between the pixels (x_0, y_0) and (x, y), h_g is the parameter that controls the spatial smoothing, and h_s is the parameter that controls the intensity smoothing.

One reason for the success of the NL-means filter [13] is the following. Because the intensity values of an image contain noise, the intensity similarity between a pixel and another pixel can be more accurately evaluated by comparing a whole patch around each pixel, not just the intensity of the pixel itself. This leads to using self-similarity, which most images have. However, if a given pixel is located on an edge, the most similar pixels can be the neighboring pixels along the edge direction. Therefore, even within a relatively small window around a given pixel, we can expect using the self-similarity of the image to improve selection of similar pixels. Based on this idea, we extend the BL filter to the patch-based bilateral (PBL) filter to improve its performance. In the PBL filter, the intensity similarity of a pixel (x, y) is evaluated by the following equation in place of (3):

$$C_s(x, y) = \exp\left(-\sum_{i=-p}^{p} \sum_{j=-p}^{p} \left(\frac{I(x_0 + i, y_0 + j) - I(x + i, y + j)}{hs} \right)^2 \right). \tag{5}$$

According to the experimental results, (5) does not use Gauss kernel weighting, though the NL-means filter does use it.

3. EXPERIMENTAL RESULTS AND COMPARISON

We conducted experiments on the performance of the PBL and BL filters on the following five test images commonly used in the previous papers: Lena (512×512), Barbara (512×512), Boats (512×512), House (256×256), and Peppers (256×256). For each test image, eight noisy versions were made by adding computer-generated white Gaussian noise with standard deviations σ_a = 5, 10, 15, 20, 25, 50, 75, and 100. The denoising performances were quantitatively measured by the peak signal-to-noise ratio (PSNR) defined as

Table 1 Best PSNR values obtained by the PBL filter

σ / PSNR	Lena	Barbara	Boats	House	Peppers
5 / 34.13	38.16	37.16	36.65	38.06	37.72
10 / 28.12	34.97	33.22	33.17	35.03	34.27
15 / 24.61	33.09	30.54	31.19	33.17	32.21
20/ 22.13	31.78	28.97	29.83	31.80	30.76
25 / 20.23	30.75	27.91	28.80	30.69	29.59
50 / 14.61	27.30	24.02	25.32	26.77	25.64
75 / 11.78	24.92	21.91	23.17	24.43	23.14
100 / 10.15	23.08	20.67	21.71	22.68	21.40

Table 2 Best PSNR values of the related methods for the noisy image with σ_a = 5 (PSNR = 34.13)

Method	Lena	Barbara	Boats	House	Peppers
PBL filter	38.16	37.16	36.65	38.06	37.72
BL filter	37.12	36.05	36.15	37.39	37.39
[16]	37.91	37.12	36.14	37.62	37.34
[15]	38.72	38.31	37.28	39.83	38.12

Table 3 Best PSNR values of the related methods for the noisy image with σ_a = 20 (PSNR = 22.13)

Method	Lena	Barbara	Boats	House	Peppers
PBL filter	31.78	28.97	29.83	31.80	30.76
BL filter	30.21	27.16	28.74	30.35	29.62
[6]	31.28	27.74	29.25	31.37	30.20
[13]	31.78	30.31	29.34	32.49	29.62
[14]	31.79	30.14	29.54	32.59	29.75
[17]	31.95	30.20	29.89	32.55	30.28
[20]	31.95	30.41	29.57	32.72	30.17
[18]	32.12	30.46	29.94	32.34	30.67
[21]	32.05	30.64	29.80	32.78	30.22
[19]	32.08	30.33	29.69	32.74	30.04
[16]	32.64	30.37	30.12	32.90	30.59
[15]	33.05	31.78	30.88	33.77	31.29

Table 4 Best PSNR values of the related methods for the noisy image with σ_a = 50 (PSNR = 14.61)

Method	Lena	Barbara	Boats	House	Peppers
PBL filter	27.30	24.02	25.32	26.77	25.64
BL filter	26.60	23.28	24.48	25.78	24.80
[16]	28.38	24.09	25.93	28.67	25.29
[15]	28.86	27.17	26.64	29.37	26.41

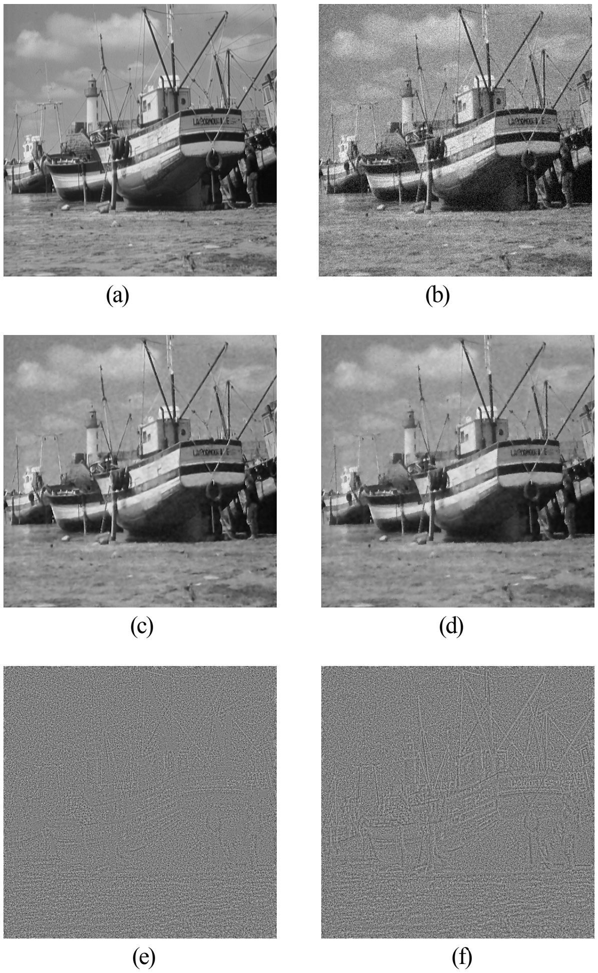

Fig. 1. Boats image: (a) original image; (b) noisy image with $\sigma_a = 20$ (PSNR = 22.18); (c) image filtered by the PBL filter (PSNR = 29.83); (d) image filtered by the BL filter (PSNR = 28.74); (e) difference image for the PBL filter; (f) difference image for the BL filter.

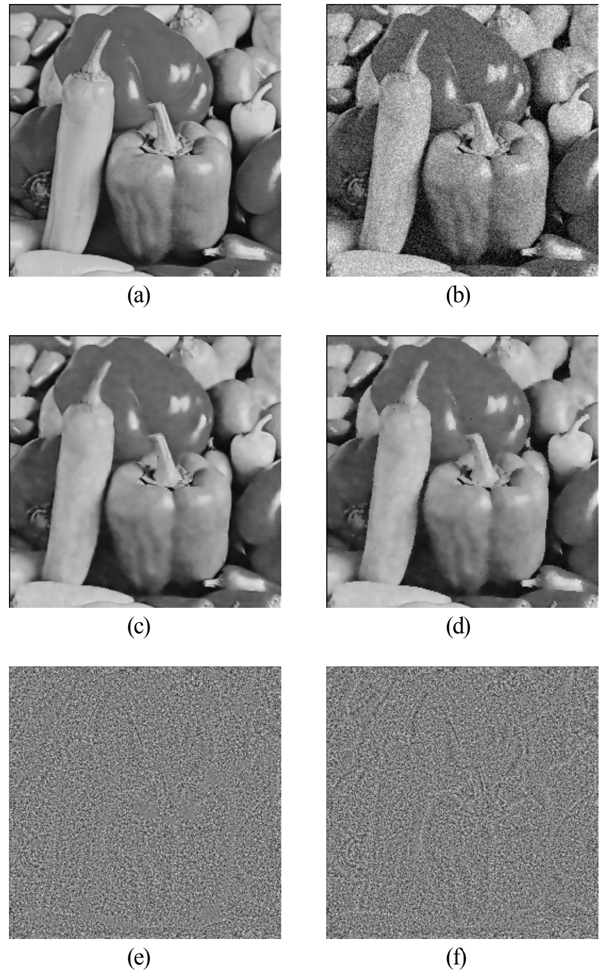

Fig. 2. Peppers image: (a) original image; (b) noisy image with $\sigma_a = 20$ (PSNR = 22.17); (c) image filtered by the PBL filter (PSNR = 30.76); (d) image filtered by the BL filter (PSNR = 29.62); (e) difference image for the PBL filter; (f) difference image for the BL filter.

$$\text{PSNR} = 10\log_{10}\frac{255^2}{\frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}\left(J(x,y)-I_0(x,y)\right)^2}, \tag{6}$$

where $I_0(x, y)$ and $J(x, y)$ denote the pixel intensity values of the uncorrupted original image and the filtered image, respectively.

To obtain the highest PSNR, the PBL and BL filters can be iterated by using the filtered image obtained by the previous iteration as the input image for the present iteration. According to the noise level of a given image, there is an optimal number of iterations that maximizes the PSNR, and any further iteration will start to degrade the structure of the image gradually. In the experiments, we ran both filters in such a way as to change the values of h_g and h_s through trial and error.

The PBL filter used a square search window of size 7 × 7 pixels for $W(x_0, y_0)$ and a square patch of size 5 × 5 pixels, which were empirically found optimal. The BL filter used a square search window of size 7 × 7 pixels for $W(x_0, y_0)$, which was empirically found optimal. Table 1 shows the best PSNR values achieved by the PBL filter using the optimal values of h_g and h_s, where σ_a is the standard deviation of the added Gaussian noise.

Tables 2, 3, and 4 show the best PSNR values of several methods related to the BL and NL-means filters, which are taken from the literatures [6], [13]–[21], and compare them with those of the PBL and BL filters. The results for [13], [14], and [17] are taken from [19]. That for [20] is taken from [21]. From these results, we can see the following. The method of [15] achieves the highest PSNR in all cases. Compared with the other methods, the PBL filter has the following advantages. In all cases, it outperforms the BL filter and the method in [6]. It is comparable to the NL-means filter [13]. Note that in the following cases, our simple PBL filter outperforms the more elaborate methods: For $\sigma_a = 5$, it outperforms the method of [16] for all images; for the Peppers image with $\sigma_a = 20$, it outperforms all the methods except the method in [15]. For the Boats image with $\sigma_a = 20$, it outperforms other methods in some cases.

Figs. 1 and 2 show the visual quality of the best filtered images obtained by the PBL and BL filters for the Boats and Peppers images with $\sigma_a = 20$. Each figure shows the uncorrupted original images, the noisy image (its PSNR value), the image filtered by the PBL filter (its best PSNR value), those by the BL filter, the difference image for the PBL filter, and that for the BL filter. The difference image was made by computing the difference in intensity values between the uncorrupted original image and the best filtered image, which was magnified by a factor of 2 and centered on 128.

From Figs. 1 and 2, we can see that the structure-preserving ability of the PBL filter is superior to that of the BL filter, especially in the edges representing outlines of objects and in the textured region. This can be understood from (3) and (5) as follows: Consider the case where the noise level is high and a given pixel is located on an edge whose height is relatively low. The proper similar pixels for the given pixel are on a line passing through the given pixel in the edge direction perpendicular to the profile. However, pixels with an intensity value almost equal to that of the given pixel other than proper similar pixels (false similar pixels) can exist in the search window. For false similar pixels, (3) can produce almost the same value as for proper similar pixels, whereas (5) can produce a smaller value than for proper similar pixels because all pixels in the patches are compared. As a result, the PBL filter can take the edge structure into account, thus reducing edge blurring.

Both methods were implemented in Java and run on a PC with a 3.2-GH_z Pentium 4 processor. For the Lena image with $\sigma_a = 20$, the PBL filter achieved the best PSNR value with two iterations, whereas the BL filter obtained it with three iterations. Under such conditions, the computation time required to achieve the best PSNR value was 27.8 seconds for the PBL filter and 3.1 seconds for the BL filter. For the NL-means filter, the study in [13] used a search window of size 21×21 pixels and a patch of size 7×7 pixels. In the patch-based method, the total number of times for the computation of the difference in intensity values of each pixel, which can be estimated from these conditions, is roughly proportional to the computation time. Therefore, the computation time required for the PBL is about one tenth that of the NL-means filter.

4. CONCLUSIONS

This paper has proposed the PBL filter. Experimental results for real images have shown that the PBL filter outperforms the BL filter in all cases, is comparable to the NL-means filter, and outperforms the more elaborate variants of the NL-means filter in some cases. The features of the PBL filter are that its algorithm is simple and its computation time is relatively small. The PBL filter is useful from a practical point of view.

REFERENCES

[1] S.M. Smith and J.M. Brady, "SUSAN–A New approach to low level image processing," *Int'l J. Computer Vision*, 23(1), pp. 45-78, 1997.

[2] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," *Proc. IEEE Int'l Conf. Computer Vision*, pp. 839-846, 1998.

[3] A.Buades, B. Coll, and J.M. Morel, "The staircasing effect in neighborhood filters and its solution," *IEEE Trans. Image Processing*, vol. 15, no. 6, pp. 1499-1505, 2006.

[4] P. Choudhury and J. Tumblin, "The trilateral filter for high contrast images and meshes," *Proc. the 14th Eurographics Workshop on Rendering*, pp.186-196, 2003.

[5] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graphics*, vol. 27, Is. 3, 2008,

[6] M. Zhang and B.K. Gunturk, "Multiresolution bilateral filtering for image denoising," *IEEE Trans. Image Processing*, vol. 17, no.12, pp. 2324-2333, 2008.

[7] W. Meng, Z. Huisheng, and H. He, "A pseudo cross bilateral filter for image denoising based on Laplacian pyramid," *Proc. IEEE Int'l Symp. Knowledge Acquisition and Modeling Workshop*, pp. 235- 238, 2008.

[8] H. Yu, L. Zhao, and H. Wang, "Image denoising using trivariate shrinkage filter in the wavelet domain and joint bilateral filter in the spatial domain," *IEEE Trans. Image Processing*, vol. 18, no. 10, pp. 2364-2369, 2009.

[9] H. Yu, L. Zhao, and H. Wang, "An efficient edge-based bilateral filter for restoring real noisy image," *IEEE Trans. Consumer Electronics*, vol. 57, no. 2, pp. 682-686, 2011.

[10] M. Elad, "On the origin of the bilateral filter and ways to improve it," *IEEE Trans. Image Processing*, vol. 11, no. 10, pp. 1141-1151, 2002.

[11] D. Barash and D. Comaniciu, "A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift," *Image and Vision Computing*, vol. 22, pp. 73-81, 2004.

[12] G. Dong and S.T. Acton, "On the convergence of bilateral filter for edge-preserving image smoothing," *IEEE Signal Processing Letters*, vol. 14, no. 9, pp. 617-620, 2007.

[13] A. Buades, B. Coll, and J.M. Morel, "A non-local algorithm for image denoising," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, pp. 60-65, 2005.

[14] S.P. Awate and R.T. Whitaker, "Unsupervised, information-theoretic, adaptive image filtering for image restoration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vo. 28, no. 3, pp. 364-376, 2006.

[15] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Processing*, vol. 16, no. 8, pp. 2080-2095, 2007.

[16] C. Kervrann and J. Boulanger, "Local adaptability to variable smoothness for exemplar-based image regularization and representation," *Int'l J. Computer Vision*, vol. 79, pp. 45-69, 2008.

[17] G. Gilboa and S. Osher, "Nonlocal linear image regularization and supervised segmentation," *SIAM Multiscale Modeling & Simulation*, 6 (2), pp. 595-630, 2007.

[18] N. Azzabou, N. Paragios, F. Guichard, and F. Cao, "Variable bandwidth image denoising using image-based noise models," *Proc. IEEE Int'l Conf. computer Vision and Pattern recognition*, pp. 1-7, 2007.

[19] T. Brox O. Kleinschmidt, and D. Cremers, "Efficient nonlocal means for denoising of textural patterns," *IEEE Trans. Image Processing*, vol. 17, no. 7, pp. 1083-1092, 2008.

[20] S. Bougleux, A. Elmoataz, and M. Melkemi, "Local and nonlocal discrete regularization on weighted graphs for image and mesh processing," *Int'l J. Computer vision*, vol. 84, pp. 220-236, 2009.

[21] L. Pizarro, P. Mrazek, S. Didas, S. Grewenig,, and J. Weickert, "Generalised nonlocal image smoothing," *Int'l J. Computer Vision*, vol. 90, pp. 62-87, 2010.