ICIC Express Letters
Volume **11**, Number **4**, April **2017**

ICIC International ©2017 ISSN 1881-803X
pp. 773–779

FAST EDGE-PRESERVING SMOOTHING FILTER BASED ON IMAGE GRADIENT

Takanori Koga[1], Shota Furukawa[2], Noriaki Suetake[2] and Eiji Uchino[2]

[1]Department of Computer Science and Electronic Engineering
National Institute of Technology, Tokuyama College
Gakuendai, Shunan 745-8585, Japan
koga@tokuyama.ac.jp

[2]Graduate School of Science and Technology for Innovation
Yamaguchi University
Yoshida 1677-1, Yamaguchi 753-8512, Japan
{ t002wa; nsuetake; uchino }@yamaguchi-u.ac.jp

Received October 2016; accepted January 2017

Abstract. *As noise removal methods for digital images, various kinds of edge-preserving smoothing filters employing normalized convolution or Poisson image editing have been proposed. Generally, those filters require high computational cost in return for their superior performance of noise removal. Therefore, many acceleration methods for those filters have been proposed so far. In this paper, especially inspired by the gradient-domain image processing of Poisson image editing, we propose a fast and simple infinite impulse response (IIR) edge-preserving smoothing filter by focusing particularly on gradients of an input image. Through some experiments, we verify that the proposed method with a small filter window is able to provide reasonably practicable edge-preserving smoothing performance compared to those of finite impulse response (FIR) filters such as the ordinary bilateral filter and guided filter, and requires only complexity of linear order computation.*

Keywords: Edge-preserving smoothing, Image gradient, IIR filter

1. **Introduction.** Edge-preserving smoothing is a fundamental technique in the fields of digital image processing [1, 2, 3] and computer graphics [4]. It smooths unnecessary local small-amplitude signal with keeping strong edges in the image. The bilateral filter [5] and ε-filter [6] are well-known and frequently-used FIR filters. The processing of those filters is conducted based on normalized convolution. Thus, their computational cost is relatively high. To cope with this problem, many acceleration methods, especially for bilateral filter, have been proposed so far. Furthermore, the filters with low computational cost such as guided filter [7] and domain transform filter [8] have been proposed.

On the other hand, edge-preserving smoothing methods based on gradient-domain image processing such as Poisson image editing [9] have been proposed and applied to various kinds of applications [10, 11]. However, in gradient-domain image processing, a resulting image in the original space must be reconstructed from a gradient domain by solving a Poisson's equation under a given boundary condition. In this case, the size of an input image leads directly to computational cost for solving the Poisson's equation because it is a simultaneous linear equation.

To cope with the above-mentioned problems, in this study, we propose a fast and simple IIR edge-preserving smoothing filter by especially focusing on the gradient-domain of an input image. Concretely, the proposed method smooths weak gradients in an input image by reconstructing the image by recursive integration of the one-dimensional signal instead of solving Poisson's equation. Compared to FIR filters, the proposed IIR filter is able to provide powerful smoothing results with a small size of the filter window. We evaluate the edge-preserving smoothing performance and computational cost of the proposed method by using grayscale artificial and natural images.

2. **Proposed Method.** The proposed method is realized by IIR filtering in which an output value is used as an input value in the next step. First, a gradient $\nabla I(x, y)$ at pixel (x, y) is calculated with the input image I and the output value T as follows:

$$\nabla I(x, y) = (I_x(x, y),\ I_y(x, y)), \tag{1}$$

$$I_x(x, y) = I(x, y) - T(x - 1, y), \tag{2}$$

$$I_y(x, y) = I(x, y) - T(x, y - 1). \tag{3}$$

Next, for each of x and y, tentative values $P_x(x, y)$ and $P_y(x, y)$ are calculated by reducing and then integrating the gradients $I_x(x, y)$ and $I_y(x, y)$ as follows:

$$P_x(x, y) = \begin{cases} I(x, y) & \text{if } x = 1,\ M \\ T(x - 1, y) + \alpha I_x(x, y) & \text{otherwise,} \end{cases} \tag{4}$$

$$P_y(x, y) = \begin{cases} I(x, y) & \text{if } y = 1,\ N \\ T(x, y - 1) + \alpha I_y(x, y) & \text{otherwise.} \end{cases} \tag{5}$$

The average of those values $P(x, y)$ is calculated as follows:

$$P(x, y) = \frac{P_x(x, y) + P_y(x, y)}{2}. \tag{6}$$

In Equations (4) and (5), M and N are the numbers of pixels in row and column, respectively. The reduction coefficient α is a real value between 0 and 1. If $\alpha = 1$, the gradients are not reduced; thus, $P(x, y) = I(x, y)$. If $\alpha < 1$, the input image is smoothed by the reduction and integration of the gradients.

The output value $T(x, y)$ is calculated by using $I(x, y)$ and $P(x, y)$ as follows:

$$T(x, y) = \begin{cases} P(x, y) & \text{if } |e| \le \varepsilon \\ (1 - \beta) I(x, y) + \beta P(x, y) & \text{otherwise,} \end{cases} \tag{7}$$

$$e = I(x, y) - P(x, y). \tag{8}$$

If the absolute value of the difference between $I(x, y)$ and $P(x, y)$ is equal to or smaller than a threshold ε, the final output value $T(x, y)$ is equal to $P(x, y)$. Otherwise, $T(x, y)$ is obtained as a weighted average of $I(x, y)$ and $P(x, y)$. β is a weight coefficient calculated as follows:

$$\beta = \frac{\varepsilon}{|e|}. \tag{9}$$

If the absolute value $|e|$ is large, then β makes $I(x, y)$ dominant in the output image. On the other hand, if $|e| \simeq \varepsilon$, then β makes $P(x, y)$ dominant in the output image.

The above-mentioned processing is performed by raster scanning order. However, the above-mentioned processing causes strain in the output image because of the phase lag of IIR filtering. In order to cope with this strain caused by the phase lag, by reference to [12], the input image is processed separately four times with changing the scanning direction. Finally, the four obtained images $T_1(x, y)$, $T_2(x, y)$, $T_3(x, y)$, $T_4(x, y)$ are averaged in order to obtain an output image without strain. Specifically, the final output of the proposed method $V(x, y)$ is calculated as follows:

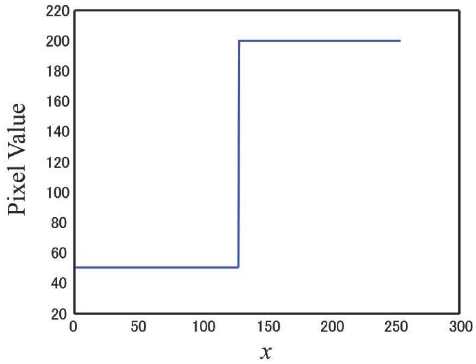
$$V(x, y) = \frac{1}{4} \sum_{i=1}^{4} T_i(x, y). \tag{10}$$

3. **Experiments.** We evaluated the edge-preserving smoothing performance and computation time of the proposed method by using an artificial image and images included in the standard image database (SIDBA) [13]. All the images have 256×256 pixels and are 8-bit grayscale. As targets for comparison, the bilateral filter [5] and guided filter [7] were used. As an evaluation index, the following mean square error (MSE) was used:

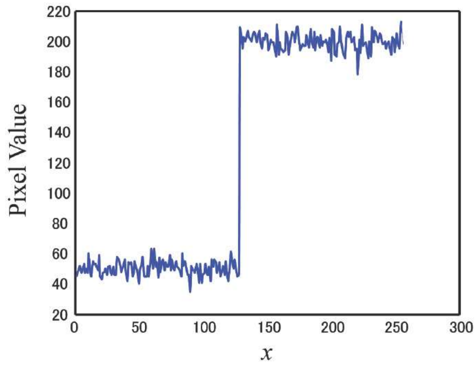
$$\text{MSE} = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} (I(x, y) - g(x, y))^2, \tag{11}$$

where I and g are the original image and the output image, respectively.

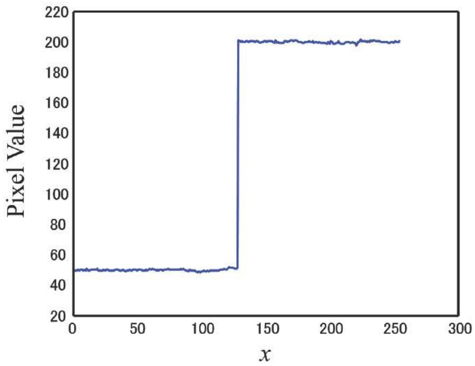
Figures 1(a)-1(e) show the results of Gaussian noise removal for the artificial image. The artificial test image without noise has two regions; the pixel values of the left-side and



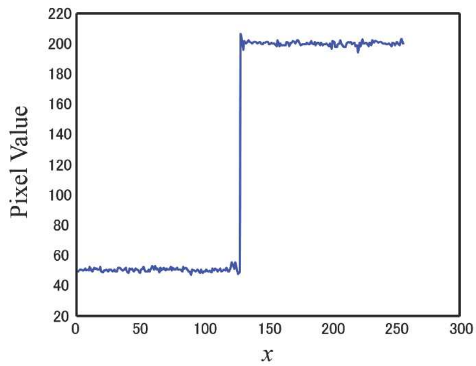
(a)



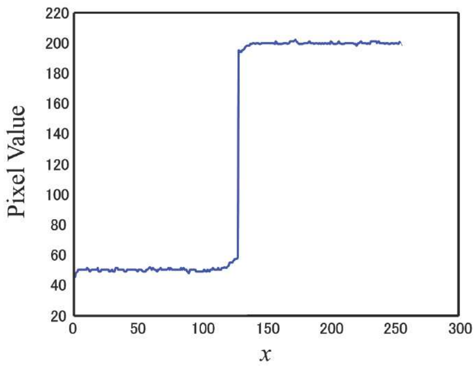
(b)



(c) $(\sigma_d, \sigma_r) = (3, 20)$



(d) $(r, \epsilon) = (4, 10^2)$



(e) $(\alpha, \varepsilon) = (0.1, 20)$

FIGURE 1. Smoothing results for the artificial image (cross-sectional 1-D signals are shown). (a) Input image, (b) noise-corrupted image, (c) smoothing result by the bilateral filter, (d) smoothing result by guided filter, and (e) smoothing result by the proposed method.

right-side areas are 50 and 200, respectively. Figures 1(a)-1(e) show the cross-sectional one-dimensional signal at the 129th row. The parameters of each method were empirically decided so that each one shows the best performance. The bilateral filter realized the ideal edge-preserving smoothing. On the other hand, both guided filter and the proposed method showed imperfect edge-preserving smoothing. The former method showed the

(a) $(\alpha, \varepsilon) = (0.25, 30)$ (b) $(\alpha, \varepsilon) = (0.2, 30)$ (c) $(\alpha, \varepsilon) = (0.15, 30)$ (d) $(\alpha, \varepsilon) = (0.1, 30)$

(e) $(\alpha, \varepsilon) = (0.25, 60)$ (f) $(\alpha, \varepsilon) = (0.2, 60)$ (g) $(\alpha, \varepsilon) = (0.15, 60)$ (h) $(\alpha, \varepsilon) = (0.1, 60)$

Figure 2. Filtering results for "Woman" by the proposed method with various parameters

(a) $(\alpha, \varepsilon) = (0.25, 30)$ (b) $(\alpha, \varepsilon) = (0.2, 30)$ (c) $(\alpha, \varepsilon) = (0.15, 30)$ (d) $(\alpha, \varepsilon) = (0.1, 30)$

(e) $(\alpha, \varepsilon) = (0.25, 60)$ (f) $(\alpha, \varepsilon) = (0.2, 60)$ (g) $(\alpha, \varepsilon) = (0.15, 60)$ (h) $(\alpha, \varepsilon) = (0.1, 60)$

Figure 3. Filtering results for "Barbara" by the proposed method with various parameters

remained small amplitude noise in the entire region. The latter one showed slight over-smoothing around the edge compared to the result by the bilateral filter. From these results, compared to the bilateral filter, it can be said that the other filters have different edge-preserving smoothing characteristics.

Figures 2 and 3 show the experimental results for "Woman" and "Barbara" in order to show the effects of the two parameters of the proposed method. From these results, it can be confirmed that the parameters α and ε control the smoothing level and the size of smoothing area, respectively.

(a) $\sigma = 0.05$ (b) $\sigma = 0.1$ (c) $\sigma = 0.05$ (d) $\sigma = 0.1$

(e) $(\sigma_d, \sigma_r) = (0.33, 12)$ (f) $(\sigma_d, \sigma_r) = (0.33, 28)$ (g) $(\sigma_d, \sigma_r) = (0.66, 15)$ (h) $(\sigma_d, \sigma_r) = (0.99, 22)$

(i) $(r, \epsilon) = (1, 9^2)$ (j) $(r, \epsilon) = (1, 19^2)$ (k) $(r, \epsilon) = (1, 7^2)$ (l) $(r, \epsilon) = (1, 15^2)$

(m) $(\alpha, \varepsilon) = (0.4, 5)$ (n) $(\alpha, \varepsilon) = (0.2, 10)$ (o) $(\alpha, \varepsilon) = (0.5, 5)$ (p) $(\alpha, \varepsilon) = (0.4, 10)$

Figure 4. Filtering result for "Woman" and "Barbara" corrupted by Gaussian noise. (a)-(d) Noise-corrupted input images, (e)-(h) filtering results by the bilateral filter, (i)-(l) filtering results by guided filter, (m)-(p) filtering results by the proposed method.

Figures 4(a)-4(p) and Table 1 show the results for the natural images corrupted by Gaussian noise. In each method, the parameters were set so that each one shows the best performance. From the results shown in figures, it seems that the visual qualities of the resulting images by all of the methods are almost equivalent on the surface. However, from Table 1, it can be quantitatively confirmed that the bilateral filter and guided filter showed slightly better performance than the proposed method and also that the performance of the proposed method is reasonably practicable.

Table 1. Evaluation by MSE for bilateral filter (BF), guided filter (GF), and the proposed method (Prop.). σ denotes a standard deviation of Gaussian noise.

	$\sigma = 0.05$				$\sigma = 0.1$			
Image Name	Input	BF	GF	Prop.	Input	BF	GF	Prop.
Airplane	24.5	13.1	13.0	14.9	98.6	38.0	40.0	45.0
Barbara	24.8	17.9	17.8	20.7	97.8	54.7	56.7	63.5
Boat	24.8	12.3	12.4	14.0	98.7	33.2	35.1	41.1
Bridge	24.4	22.0	21.8	23.7	97.4	69.3	70.1	75.1
Building	23.8	16.5	17.3	18.0	94.9	46.3	49.5	51.6
Cameraman	24.2	11.7	12.2	14.1	95.0	34.5	36.9	42.2
Girl	24.4	12.8	12.6	13.6	97.1	29.3	30.2	35.0
Lax	24.6	20.2	20.3	23.1	98.5	59.3	62.3	69.0
Lenna	24.8	12.4	12.1	14.0	97.9	33.1	34.3	41.3
Lighthouse	23.6	15.9	15.9	18.8	93.7	48.0	50.3	57.0
Text	23.4	17.7	17.7	18.8	93.3	52.3	52.6	56.6
Woman	24.5	13.3	13.4	14.9	98.3	34.8	37.2	43.0
Average	24.3	15.5	15.5	17.4	96.8	44.4	46.3	51.7

The computational complexities of each method are explained as follows. The computational complexity of the bilateral filter is $O(nr^2)$. Here, n and r are the total number of the pixels and the radius of the filter kernel, respectively. On the other hand, the computational complexity of the proposed method is $O(n)$. This is the same as that of guided filter which is a fast and sophisticated edge-preserving smoothing filter. In this regard, the proposed method requires four times filtering for the whole image in order to resolve the phase lag. Meanwhile, guided filter requires a pre-calculated integral image. It is difficult to compare the calculation speed accurately in the hardware level, and we show the numbers of the four arithmetic operations of each method in Table 2 just for reference. From the table, it can be said that both methods need almost the same numbers of operations, though the proposed method needs more multiplication and division operations.

Table 2. The numbers of operations of guided filter and the proposed method

	Guided filter	Proposed method (in each direction)	Proposed method (in 4 directions + averaging)
Addition and subtraction	47	8	$32 + 3$
Multiplication and division	13	6	$24 + 1$
Total	60	14	60

Actually, although the proposed method takes longer calculation time than that of guided filter, the proposed method can be parallelized smoothly; thus, this problem is easily solved. For reference, as a result of simple parallelization with OpenMP and four-core central processing unit (CPU), the calculation time of the proposed method was shorter than that of guided filter. However, this matter should be investigated more because the performance of parallel processing depends on the environment and implementation methods.

4. **Conclusions.** In this paper, we proposed a fast edge-preserving smoothing IIR filter by focusing on the gradient-domain image processing without solving Poisson's equation for the reconstruction. In the proposed method, the complexity of linear order computation was realized while it requires four times filtering in order to remove the phase lag. Through a series of experiments, the reasonably practicable edge-preserving smoothing performance of the proposed method was verified.

The future works are to develop an automatic parameter adjustment method and an acceleration method in the image reconstruction by using finely-tuned parallel computing technique.

REFERENCES

[1] P. Jain and V. Tyagi, A survey of edge-preserving image denoising methods, *Inf. Syst. Front.*, vol.18, no.1, pp.159-170, 2016.
[2] T. Koga, E. Uchino, N. Suetake, G. Hashimoto, T. Hiro and M. Matsuzaki, Speckle noise reduction and edge-enhancement of coronary plaque tissue in intravascular ultrasound image by using anisotropic diffusion filter, *Int. J. Circuits, Syst. Signal Process.*, vol.2, no.4, pp.239-248, 2008.
[3] T. Koga and N. Suetake, Image coarsening by using space-filling curve for decomposition-based image enhancement, *J. Vis. Commun. Image Rep.*, vol.24, pp.806-818, 2013.
[4] M. Sun and J. Zhang, Edge-preserving traveltime tomography with a sparse multiscale imaging constraint, *J. Appl. Geo-phy.*, vol.131, pp.179-190, 2016.
[5] C. Tomasi and R. Manduchi, Bilateral filtering for gray and color images, *Proc. of IEEE Int. Conf. Computer Vision*, pp.839-846, 1998.
[6] H. Harashima, K. Odajima, Y. Shishikui and H. Miyagawa, ε-separating nonlinear digital filter and its applications, *IEICE Trans. Fund.*, vol.J65-A, no.4, pp.297-304, 1982 (in Japanese).
[7] K. He, J. Sun and X. Tang, Guided image filtering, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol.35, no.6, pp.1397-1409, 2013.
[8] E. S. L. Gastal and M. M. Oliveira, Domain transform for edge-aware image and video processing, *ACM Trans. Graphics (SIGGRAPH 2011)*, vol.30, no.4, pp.69:1-69:12, 2011.
[9] P. Patrick, G. Michel and B. Andrew, Poisson image editing, *ACM Trans. Graphics (SIGGRAPH 2003)*, vol.22, no.3, pp.313-318, 2003.
[10] S. Miyaoka, Gradient-domain image filtering, *IPSJ Journal*, vol.52, no.2, pp.901-909, 2011 (in Japanese).
[11] R. Fattal, D. Lischinski and M. Werman, Gradient domain high dynamic range compression, *ACM Trans. Graphics (SIGGRAPH 2002)*, vol.21, no.3, pp.249-256, 2002.
[12] M. Muneyasu, S. Tsujii and T. Hinamoto, A 2-D IIR neural hybrid filter for image processing, *IEICE Trans. Fund.*, vol.J76-A, no.10, pp.1440-1449, 1993 (in Japanese).
[13] M. Sakauchi, Y. Ohsawa, M. Sone and M. Onoe, Management of the standard image database for image processing researches (SIDBA), *ITEJ Tech. Rep.*, vol.8, no.38, pp.7-12, 1984 (in Japanese).