ICIC Express Letters
Part B: Applications
Volume **7**, Number **7**, July **2016**

ICIC International ©2016 ISSN 2185-2766
pp. 1467–1474

SEPARABLE GEODESIC FILTERING FOR FAST IMAGE DENOISING

Dawei Guo, Min Cai and Aiqi Wang

School of Science
Dalian Jiaotong University
No. 794, Huanghe Road, Shahekou Dist., Dalian 116028, P. R. China
542511470@qq.com; dlcaim@126.com

Received December 2015; accepted March 2016

Abstract. *This paper proposes a fast edge-preserving smoothing filter. Based on the intrinsic geometry of an image surface-geodesic distance which is used to measure the similarity of two pixels, this method can filter out noise with preserved edges and fine-scaled details. Moreover, its separable implementation is simple and fast, and can process efficiently color images as well as gray images. Simulation results of image denoising demonstrate the efficiency of the proposed algorithm.*
Keywords: Image denoising, Geodesic transform, Edge-preserving

1. **Introduction.** Filtering is perhaps the most important operation of image processing and computer vision, and it is used extensively in the wide range of applications, including image smoothing and sharpening, noise removal, resolution enhancement, edge detection and so on. The simplest filtering should be linear translation-invariant (LTI), which calculates the filtering output according to the weight of neighbor pixels. Since the LTI filtering uses the same weighted model in each neighborhood, it can implement fast by using the simple convolution. For instance, Gaussian filtering which is widely used in image processing, computes the weights according to the Gaussian function. Although LTI filtering is very simple and is used extensively in early vision processing, it still has some disadvantages. LTI filtering smoothes the noise, but blurs important structures along with noise, and outliers exert large influence on filtered outputs.

In order to overcome the shortcomings of linear filtering, many nonlinear filtering methods are proposed by researchers. Currently, bilateral filter (BF) is a relatively popular filtering method for preserving edge. BF was first proposed by Aurich and Weule [1] and called as nonlinear Gaussian filter. From then on, Tomasi and Manduchi [2] rediscussed this method and called it bilateral filter (BF). Similar to the Gaussian filtering, BF's output at one point $P(x, y)$ is also the weighted average of local neighbor pixels. However, BF's weight does not only depend on the distance between center pixel and neighborhood pixels, but also depend on the grayscale (color) distance among them. In the smooth area, since the pixel values in the local neighborhood are close to each other, the weight is mainly decided by the space distance. Nonetheless, in the non-smooth area, the pixel values changed dramatically, therefore weights are greatly influenced by the grayscale (color) distance. Bilateral filtering can effectively remove noise with well-preserved details of original images [3-7]. However, implementing BF method directly will be slow. Reducing the complexity of calculating will be still challenging [8,9].

Lately, based on the kernel regression (attenuation) framework, Takeda et al. [10] unified the Gaussian filtering and BF method, and proposed a novel method of image filter-locally adaptive regression kernels (LARK). Compared with traditional filtering methods, LARK uses the structure tensor as the kernel regression to measure similarities between the pixels.

In fact, the structure tensor is similar to the geodesic distance. Geodesic distance is the intrinsic geometry of the image surface, which can reflect the characteristics of the image structures better and measure similarities among pixels more accurately compared with Euclidean distance. Therefore, the geodesic distance filtering method will be more effective. However, geodesic distance's calculation is much more complex than the Euclidean distance. From the point of view of graph theory, the essence of calculating geodesic distance should be how to get the shortest path. Although, some fast algorithms have been proposed [11,12], the target of researchers is still to reduce the amount of calculation.

In this paper, firstly we discuss the geodesic distance in one-dimensional space and propose a fast algorithm based on the additivity of one-dimensional distance. Then it will be generalized to two-dimensional images, and a fast separable geodesic filtering algorithm will be proposed. The filtering method is simple and has a linear time implementation. The experimental results show that competitive results can be achieved in image denoising and joint filtering.

2. The Fast Separable Algorithm of the Geodesic Distance.

2.1. The geodesic distance.

Suppose $I(x) : \Omega \to R^d$ is an image ($d = 1$ for grayscale, $d = 3$ for color), where $\Omega \subset R^2$ is a spatial coordinates domain. The geodesic distance $D(x, y)$ between pixel x and pixel y is defined as [13]:

$$D(x,y) = \min_{\Gamma \in P_{x,y}} d(\Gamma) = \min_{\Gamma \in P_{x,y}} \int_0^1 \sqrt{\|\Gamma'(s)\|^2 + \gamma^2 \left(\nabla I(s) \cdot \mathbf{u}\right)^2} ds \tag{1}$$

where $P_{x,y}$ is a set of all paths connecting x and y, and Γ is such a representation of arc-length parameterization. ∇I and $\Gamma'(s)$ separately mean derivatives of image gradient and the path. The unit vector $\mathbf{u}$ is along the tangent direction path of Γ. The weight of balance image gradient and space distance is γ. For digital images, Γ is a connection between x and y in 8-connected space neighborhood pixels of discrete sequence. $\{p_0 = x, p_1, \cdots, p_{n-1}, p_n = y\}$, so

$$d(\Gamma) = \sum_{i=1}^{n} d_{i,i-1} = \sum_{i=1}^{n} \sqrt{1 + \gamma^2 \left\|I(p_i) - I(p_{i-1})\right\|_2} \tag{2}$$

where $\|\cdot\|_2$ is L_2 norm.

Simply, the geodesic distance between two points – x and y is the shortest distance along the surface. Directly, if the geodesic distance between two points is smallest on image I, definitely there will be a shortest path about color changes. The small geodesic distance shows more similarities between two points.

No matter the classic Gaussian filtering, bilateral filtering, or geodesic filtering, they are used to measure similarities among pixels and to combine neighborhood information for filtering. So they could be a unified form as the following [10]:

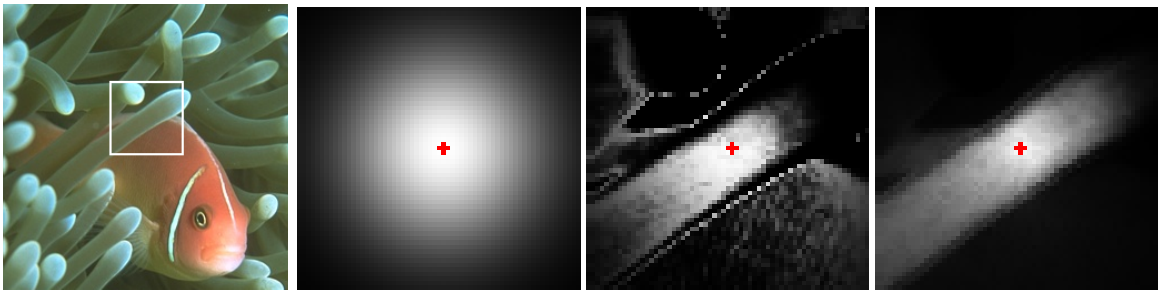
$$\widetilde{I}(x) = \sum_{y \in U(x)} K\left(x, y, I(x), I(y)\right) I(y) \tag{3}$$

where $\widetilde{I}(x)$ is the filtering output on the point of x, $U(x)$ is the neighborhood using the point x as the center point. (For the simplicity, square neighborhood is chosen.) $K(x, y, I(x), I(y))$ is the kernel function. It shows that the Gaussian Filter is $K(x, y, I(x), I(y)) = \exp\left(\frac{\|x-y\|^2}{\sigma_s^2}\right)$, if the kernel function is only related with space distance; it shows that the bilateral filter is

$$K\left(x, y, I(x), I(y)\right) = \exp\left(\frac{\|x-y\|^2}{\sigma_s^2}\right) \exp\left(\frac{\|I(x) - I(y)\|^2}{\sigma_r^2}\right),$$

if the kernel function is only related with space distance and grayscale distance; it shows that geodesic filter is $K(x, y, I(x), I(y)) = \exp\left(\frac{D(x,y)}{\sigma_r^2}\right)$, if the kernel function is the geodesic distance function, where σ_s, σ_r control the weight.

Figure 1 compares the results of the weight for different filtering methods. It is summarized that the geodesic distance could affect images' structures and measure the similarity between pixels better. Therefore, the filtering method based on geodesic distance can get better effective results.



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

Figure 1. The comparison of weight for different methods: (a) original image; (b) Gaussian weight; (c) bilateral filtering weight; (d) geodesic distance weight

2.2. **The fast algorithm.** Although the geodesic distance weight could enhance the filtering effectiveness, Formula (1) shows that the calculation of geodesic distance is more complex compared with the Euclidean distance. In order to work out all of possible paths between two points, the complexity of calculating geodesic distance must be reduced. Next let us discuss it in two conditions as follows.

(a) **One-dimensional situation.** Firstly, let us consider Formula (1) in the situation of 1-D, and that is to say a 2-D image should be disintegrated to curves in 1-D.

Suppose $u(x) : [a, b] \to R$ is random, from Formula (1), for the random point $y \in [a, b]$, the geodesic distance is between the point a to y. In order to distinguish Formula (1) in two-dimensional situation, here $d(x, y)$ is used for expression:

$$d(a, y) = d(\Gamma) = \sum_{i=1}^{n} d_{i,i-1} = \sum_{i=1}^{n} \sqrt{1 + \gamma^2 \left\| u(p_i) - u(p_{i-1}) \right\|_2} \tag{4}$$

where Γ is a discrete series $\{p_0 = a, p_1, \ldots, p_{n-1}, p_n = y\}$ between point a and point y. On 1-D curve, the connection of discrete series between point a and point y is unique. So the minimum operation is not necessary.

Since on 1-D curve the distance between two points is with the feature of additivity, the random geodesic distance $x, y \in [a, b]$ can be expressed as: (suppose $x \le y$)

$$d(x, y) = d(a, y) - d(a, x) \tag{5}$$

According to Formula (5) for computing the geodesic distance, firstly the geodesic distances among point a and other points should be computed, and then subtraction should be programmed. Nevertheless, it is simple to calculate the geodesic distances among point a and other points, and that means two points in neighborhood should be calculated firstly:

$$d_{i,i-1} = \sqrt{1 + \gamma^2 \left\| u(p_i) - u(p_{i-1}) \right\|_2} \tag{6}$$

Then the cumulative sum should be done. Therefore, according to Formulas (4) and (5) for computing the geodesic distance, the amount of computing is N (the number of points on the curve of discrete series). However, according to Formula (1) or (2), the amount of computing is $\frac{1}{2}N(N-1)$.

(b) **Two-dimension situation.** In the previous section, Formula (5) provides the fast algorithm of the geodesic distance in the situation of 1-D, but it cannot be generalized to 2-D. Because in the situation of 2-D the path is not unique, the distance could not be added together.

For applying the fast algorithm obtained in the situation of 1-D, one of effective methods is to process each dimension of 2-D image separately. For digital images, operation should be done according to rows firstly and columns secondly. In order to get the effective information of other pixels in neighborhood, iteration should be done three times in the experiments. (One time means one row's processing firstly and one column's processing secondly)

Although in the program mentioned above combination of the arc length was used instead of the geodesic distance, it is not the true geodesic distance. However, the following experiments will show that the results could be more similar to the geodesic distance with smaller amount of calculations.

The conclusion of FSGF is as the following.

Fast separable geodesic filtering algorithm FSGF
Suppose $I(x)$ is the input image, make $k = 0, \ldots, K$ (iterations), $I^0(x) = I(x)$.
1 Filter $I^k(x)$ row by row by using 1-D geodesic filtering method, where geodesic distance follows Formulas (4) and (5), and record the filtering result as $I_r^k(x)$;
2 Rotate $I_r^k(x)$ clockwise about $\pi/2$, repeat the step 1 column by column, and record the filtering result as $I_c^k(x)$;
3 Contrarotate $I_c^k(x)$ about $\pi/2$, record it as $I^{k+1}(x)$;
4 If $k+1 = K$, stop the operation; otherwise, repeat the step 1;

3. **Joint Geodesic Distance Filtering.** On the same occasion, when there are several unideal images, one of logical methods is to joint these images for better results. For instance, when the photos are taken in dark environment, flash is needed for strong lighting to get high signal noise ratio (SNR). However, the color of images will be darker, such as Figure 5(a). If the photo is taken with non-flash, colorful information will be provided with high SNR, such as Figure 5(b). For getting more vivid and colorful images, one of best methods is to use guided flash images function by using the feature of clear edge, dark color and high SNR for getting non-flash images. This joint filtering theory could be implemented simply and rapidly by taking advantages of the separable geodesic distance method.

In order to be able to monitor image edge information, the local neighborhood filtering Formula (3) can be changed as follows:

$$\widetilde{I}(x) = \sum_{y \in U(x)} K\left(x, y, g(x), g(y)\right) I(y) \tag{7}$$

where $\widetilde{I}(x)$ is the result of joint filtering, $I(x)$ is the input, and $g(x)$ is the guided image. Joint filtering Formula (7) can be explained as the following: denosing images with no significant fuzzy edge could be implemented by using the information provided by guided image $g(x)$ and the weight structured self-adaptively.

4. **Results and Discussions.** Based on geodesic distance, the filtering method provides better edge-preserving features, and it is obviously better than traditional methods on balanced filtering and running time with the simple algorithm and smaller amount of calculations. The experiments in this section are all implemented in the environment of Matlab7.0. All the experimental test images are given in Figure 2.

FIGURE 2. All experimental test images

4.1. **Image denoising.** Denoising is the most basic and important step in image processing. In experiment 1, the proposed algorithm is evaluated and compared with many other existing techniques, including BF [2] and LARK [10] for enhancing the feature of better denoising effects and shorter running time. In the experiments, the size of neighborhood uses the unified 11*11 square window. Other parameters will be taken artificially for making each method most optimal. The simulation of noises is all Gaussian noises added by the value of zero average, and standard deviation is separate $\sigma = [10, 15, 20, 25]$. For comparing the effectiveness of the denoising objectively, PSNR (peak signal-to-noise ratio) is taken as evaluation criterion, and it is defined as follows:

$$\mathrm{PSNR} = 10 \lg \frac{N \times 255^2}{\sum_{i=1}^{N} (u(i) - u^*(i))^2} \tag{8}$$

where N means the number of images' pixels, u is the original vivid image, and u^* is the filtered image.

The relative PSNR results are summarized in Table 1 and Table 2 and calculating time of various methods has been considered in this paper. From Table 1, we can notice that our method is more excellent than other traditional methods for better denoising effectiveness. The average rising of PSNR value is 0.6dB. Furthermore, according to the geodesic distance Formula (4), the method could be implemented on color images directly. Since jointing 3 passageways' information – R, G, B the advantages of this method are more outstanding for color images. Although LARK could get the best result, Table 2 shows the running time is much longer than the others.

For visually comparing each method's effectiveness of denoising, some visual examples for the "peppers" and "butterfly" test image are shown in Figure 3, where the value of

TABLE 1. Comparison of filtering results for different methods (dB)

σ	10	15	20	25	10	15	20	25	10	15	20	25
Input PSNR	28.13	24.61	22.11	20.17	28.13	24.61	22.11	20.17	28.13	24.61	22.11	20.17
Grayscale	Lena 256 × 256				Peppers 256 × 256				Cameraman 256 × 256			
BF	31.67	29.57	28.26	27.10	33.09	30.84	29.22	27.89	32.53	30.22	28.69	27.20
LARK	32.15	30.24	29.12	28.35	33.78	31.59	30.32	29.26	32.79	31.05	29.21	29.30
FSGF	32.09	29.84	28.38	27.42	33.28	31.31	29.72	28.79	32.63	30.24	28.70	27.50
Color	Bird 481 × 321				Butterfly 481 × 321				Starfish 481 × 321			
BF	35.50	32.72	30.98	29.40	31.97	29.75	28.19	27.10	31.25	29.53	28.05	26.77
LARK	36.89	34.86	33.75	32.89	33.65	31.75	30.26	29.23	33.21	31.35	30.10	29.05
FSGF	36.71	34.58	33.08	32.01	33.40	31.28	29.83	28.71	32.93	30.79	29.35	28.31

TABLE 2. Comparison of running time for different methods (second)

Size of image	BF	LARK	FSGF
128 × 128	0.2791	15.37	0.2256
256 × 256	0.9647	61.45	0.6323
215 × 512	3.6897	242.37	1.8854
1024 × 1024	13.2384	1138.51	6.7402

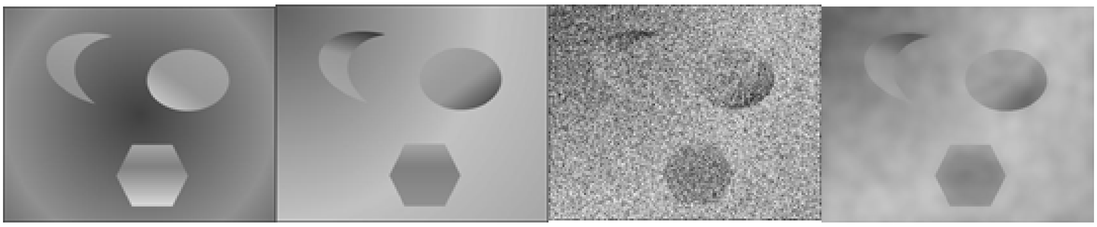
(a) Noise image (b) BF (c) LARK (d) FSGF

FIGURE 3. Results for test images of peppers and butterfly

Gaussian noise is $\sigma = 25$. It shows that the method mentioned in the paper could filter the noise without obvious fuzzy edge.

4.2. **Joint denoising.** On the same occasion, when there are several unideal images, the most common method is to use the joint filtering for better images. Let us take below two cases for verifying the effectiveness of the method in joint filter application. Firstly, Figure 4 shows the simulation results of composite images. Where, Figure 4(a) is a guided image, which composes of geometry with different shapes and different grayscales; Figure 4(b) is the original image with the same edge structure and different grayscales' comparison; Figure 4(c) is the noised image pulsed by the standard deviation $\sigma = 100$ of Gaussian noise with the value of zero, so edge information is almost submerged in strong noise; Figure 4(d) is the joint filtering output by using Figure 4(a) as the supervisory signals to do joint filtering for Figure 4(c). Obviously, although the image was influenced by strong noises, the joint filtering method mentioned in this paper could restore the original signal better.

Next for the true flash and no-flash images, Figure 5 compares the filtering results with the guided image function and without the guided image function. Figure 5(a) is a flash image, Figure 5(b) is a no-flash image, Figure 5(c) is the filtering result without using the conclusion of Figure 5(a), and Figure 5(d) is the filtering result with guided image of Figure 5(a). Obviously, the joint filtering result shows better effects. Because the joint filtering used the information provided by guided function, more adapted weights average has been gotten for image's structure.



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

Figure 4. Performance of the joint filtering. (a) Guided image. (b) Original image. (c) Noised image. (d) Denoised results using joint filtering.

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

Figure 5. Flash/no-flash denoising. (a) Flash image. (b) No-flash image. (c) Denoised results without using (a). (d) Denoised results using joint filtering.

5. **Conclusion.** This paper proposed the fast image-filtering method based on the separable geodesic distance. Firstly, the fast separable algorithm of the geodesic distance in one-dimensional space was discussed. Then it was generalized to two-dimensional images according to using the simple method in one-dimensional space repeatedly. Finally, the fast image filtering method – the separable geodesic filtering method was concluded. This method is not only simple with small amount of calculation, but also with a good approximation of geodesic distance. Simulation results show that the proposed method is obviously superior to traditional methods both on the balanced filtering effectiveness and the running time.

For the future work, we would like to investigate more sophisticated schemes to apply geodesic filtering method to other noise, for example, impulse noise. And we would also like to extend our approach to more application, including image matting, image colorization and so on.

Acknowledgment. This work is supported by the Scientific Research Project of Liaoning Province Educational Department, P. R. China (No. L2014186). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] V. Aurich and J. Weule, Non-linear Gaussian filters performing edge preserving diffusion, *Proc. of DAGM Symposium*, vol.17, pp.538-545, 1995.
[2] C. Tomasi and R. Manduchi, Bilateral filtering for gray and color images, *Proc. of the 6th International Conference on Computer Vision*, New Delhi, pp.839-846, 1988.
[3] S. A. Jose and C. S. Seelamantula, Bilateral edge detectors, *IEEE Int. Conf. Acoust., Speech and Signal Proces.*, pp.1449-1453, 2013.
[4] G. Li and W. Yu, SAR image despeckling based on adaptive bilateral filter, *Journal of Electronics & Information Technology*, vol.34, no.5, pp.1076-1081, 2012.

[5] J. W. Han, J. H. Kim, S. H. Cheon et al., A novel image interpolation method using the Bilateral filter, *IEEE Trans. Consumer Electronics*, vol.56, no.1, pp.175-181, 2010.

[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] Y. Chen and Y. Shu, Optimization of bilateral filter parameters via chi-square unbiased risk estimate, *IEEE Signal Process. Lett.*, vol.21, no.1, pp.97-100, 2014.

[8] F. Porikli, Constant time $o(1)$ bilateral filtering, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.1-8, 2008.

[9] Q. Yang, K. H. Tan and N. Ahuja, Real-time $o(1)$ bilateral filtering, *IEEE Conference on Computer Vision and Pattern Recognition*, pp.557-564, 2009.

[10] H. Takeda, S. Farsiu and P. Milanfar, Kernel regression for image processing and reconstruction, *IEEE Trans. Image Processing*, vol.16, no.2, pp.349-366, 2007.

[11] P. J. Toivanen, New geodesic distance transforms for gray-scale images, *Pattern Recognition Letters*, vol.17, no.5, pp.437-450, 1996.

[12] L. Yatziv, A. Bartesaghi and G. Sapiro, $O(n)$ implementation of the fast marching algorithm, *Journal of Computational Physics*, vol.212, no.1, pp.393-399, 2006.

[13] A. Criminisi, T. Sharp et al., Geodesic image and video editing, *ACM Trans. Graphics*, vol.29, no.5, pp.1-15, 2010.