Multigrid Bilateral Filter

Zongqing Lu

Abstract—It has proved that nonlinear diffusion and bilateral filtering (BF) have a closed connection. Early effort and contribution are to find a generalized representation to link them by using adaptive filtering. In this paper a new further relationship between nonlinear diffusion and bilateral filtering is explored which pays more attention to numerical calculus. We give a fresh idea that bilateral filtering can be accelerated by multigrid (MG) scheme which likes the nonlinear diffusion, and show that a bilateral filtering process with large kernel size can be approximated by a nonlinear diffusion process based on full multigrid (FMG) scheme.

Keywords—Bilateral filter, multigrid

I. INTRODUCTION

FILTERING is perhaps the most fundamental operation of computer vision. Simple smoothing operations such as low-pass filtering, which does not take into account intensity variations within an image, tend to blur edges. The more advanced methods for noise removal aim at preserving the signal details while removing the noise. This is achieved by a locally adaptive recovery paradigm, such as: anisotropic diffusion (AD)[1][2][3][4][5], weighted least squares (WLS)[6], robust estimation (RE)[7]. All these methods share the fact that local relations between the samples dictate the final result, and therefore, all these methods resort to an iterative algorithm. There is a solid theoretical bridge between these methods as well as to the line-process approach[8]. Recently, Tomasi and Manduchi proposed an alternative noniterative bilateral filter for removing noise from images[10]. This filter is merely a weighted average of the local neighborhood samples, where the weights are computed based on temporal (or spatial in case on images) and radiometric distances between the center sample and the neighboring samples. This filter is also locally adaptive, and it was shown to give similar and possibly better results to those obtained by the previously mentioned iterative approaches.

The nature of bilateral filtering resembles that of nonlinear diffusion. It is therefore suggested the two are related and a unified viewpoint can reveal the similarities and differences between the two approaches. Adaptive smoothing serves as a link between the two approaches[19]. In nonlinear diffusion, several iterations are performed. In bilateral filtering, the window of the filter becomes much bigger in size than the one used in adaptive smoothing and there is no need to perform several iterations. In some research, it is shown that the bilateral filter is identical to the first iteration of the Jacobi algorithm (diagonal normalized steepest descent) with a specific cost function[21].

Z. Q. Lu. Author is with Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China (e-mail: zq.lu@ siat.ac.cn)

Unlike AD, bilateral filtering is a non-iterative process, so we do not associate bilateral filtering with multigrid (MG) scheme for acceleration customarily which often was applied to nonlinear diffusion. A new relationship between bilateral filtering and nonlinear diffusion is explored in this paper. Our work is motivated by a fresh idea that a bilateral filtering process with large kernel size can be approximated by a nonlinear diffusion process based on full multigrid (FMG) scheme. It is the first time to apply MG to accelerate bilateral filtering.

II. BILATERAL FILTER

The bilateral filter is technique to smooth images while preserving edges. It can be traced back to 1995 with the work of Aurich and Weule[11] on nonlinear Gaussian filters. It has been later rediscovered by Smith and Brady[12] as part of their SUSAN framework, and Tomasi and Manduchi[10] who gave it its current name. Since then, the use of bilateral filtering (BF) has grown rapidly and is now ubiquitous in image-processing applications. BF is a nonlinear filter which combines domain and range filtering (Fig. 1). Given an input image $i(\boldsymbol{x}), \boldsymbol{x} = [x_1, x_2] \in \mathbf{R}^2$, using a continuous representation notation, the output image $i'(\boldsymbol{x})$ is obtained by

$$i'(\boldsymbol{x}) = k^{-1}(\boldsymbol{x}) \int_{\boldsymbol{\tau} \in \Omega(\boldsymbol{x})} i(\boldsymbol{\tau}) c(\boldsymbol{\tau}, \boldsymbol{x}) s(i(\boldsymbol{\tau}), i(\boldsymbol{x})) d\boldsymbol{\tau} \quad (1)$$

Where $\boldsymbol{\tau} = [\tau_1, \tau_2] \in \Omega(\boldsymbol{x})$. The convolution mask is the product of the functions c and s, which represent 'closeness' (in the domain) and 'similarity' (in the range), respectively. In the bilateral filter, the domain weight c is given by

$$c(\boldsymbol{\tau}, \boldsymbol{x}) = \exp\left(-\frac{\|\boldsymbol{x} - \boldsymbol{\tau}\|^2}{2\sigma_c^2}\right)$$

Where σ_c is the standard deviation in the domain and $\|\boldsymbol{\tau} - \boldsymbol{x}\|$ is the Euclidean distance between the pixels $\boldsymbol{\tau}$ and $\boldsymbol{x}$. For the range weights between pixels $\boldsymbol{\tau}$ and $\boldsymbol{x}$, we use

$$s(i(\boldsymbol{\tau}), i(\boldsymbol{x})) = \exp\left(-\frac{\|i(\boldsymbol{x}) - i(\boldsymbol{\tau})\|^2}{2\sigma_s^2}\right)$$

Where σ_s is the standard deviation in the range and $\|i(\boldsymbol{\tau}) - i(\boldsymbol{x})\|$ is a measure of the distance between the intensities at pixels $\boldsymbol{\tau}$ and $\boldsymbol{x}$. If the two pixels vary widely in their intensities, as is the case at an edge, then the corresponding weight in the range filter is small, thus reducing the smoothing. $k(\boldsymbol{x})$ is a normalization factor:

$$k(\boldsymbol{x}) = \int_{\Omega(\boldsymbol{x})} c(\boldsymbol{\tau}, \boldsymbol{x}) s(i(\boldsymbol{\tau}), i(\boldsymbol{x})) d\boldsymbol{\tau} \quad (2)$$

The bilateral filter has several qualities that explain its attraction:

♣ It's formulation is simple: each pixel is replaced by an average of its neighbors.

World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:5, No:7, 2011

♣ It depends only on two parameters that indicate the size and contrast of the features to preserve.

♣ It can be used in a non-iterative manner. This makes the parameters easy to set since their effect is not cumulative over several iterations.

Fig. 1 Left: original image; Right: filtered by BF [10]

III. MULTIGRID: THE NEW LINK BETWEEN BILATERAL FILTERING AND NONLINEAR DIFFUSION

A. Downsampling and bilateral filtering

One research interest of bilateral filter is the acceleration of computation speed. Multiresultion or downsampling scheme is often used efficient way[20][21][22]. Paris and Durand [20] analyzed accuracy in terms of bandwidth and sampling, and derive criteria for downsampling in space and intensity to accelerate the bilateral filter by extending an earlier work on high dynamic range images. Their method approximates the bilateral by filtering subsampled copies of the image with discrete intensity kernels, and recombining the results using linear interpolation.

Zhang and Gunturk[21] used multiresolution scheme and wavelet to eliminate noise. It avoids over-smoothing texture regions and to effectively eliminate blocking and ringing artifacts, and accelerate filtering process at the same time.

Downsampling based bilateral filtering techniques need two indispensable steps: downsampling and interpolation. The former insures the low computation time, and the latter eliminates the artifact noise produced by downsampling.

B. Multiscale based bilateral filtering

Borrow the idea of downsampling scheme, we try to extend bilateral filtering to multiscale based version. Downsample the original input image $i^{(0)}(x)$ to $i^{(1)}(x) = i^{(0)}(x)_\downarrow$. In order to reduce computational cost, for $i^{(1)}(x)$ it is not necessary to use the original kernel $ke = c \cdot s$, but a smaller size version ke'. Let $\tilde{i}^{(0)}(x)$ be the wanted result of $i^{(0)}(x)$, $\tilde{i}^{(1)}(x)$ is the filtered result of $i^{(1)}(x)$ and $e^{(1)}(x) = i^{(1)}(x) - \tilde{i}^{(1)}(x)$ is the noise of $i^{(1)}(x)$. Let $e^{(1)}(x)_\uparrow$ be the interpolated version of $e^{(1)}(x)$ which is the low-frequency noise of $i^{(0)}(x)$. Here we have the following formula (Fig. 2):

$$i^{(0)}(x) - e^{(1)}(x)_\uparrow = \tilde{i}^{(0)}(x) + e^{(0)}(x)$$

Where $\uparrow, \downarrow, \otimes$ denote downsampling, interpolation and convolution operators respectively. $e^{(0)}$ is the residue noise or high-frequency noise of $i^{(0)}$. In order to eliminate $e^{(0)}$ from $i^{(0)}(x)$, it needs a further filtering step. High frequency noise $e^{(0)}$ can be approximately described as $N(0, \sigma)$, therefore the final result $\tilde{i}^{(0)}(x)$ is insensitive to the choice of $c(\tau, x)$, in other words we can still choose the smaller kernel and simply suggest the filtering application several times. The nature of multiscale idea is to replace large kernel convolution by several times filtering with small kernel over more scale levels. Let $i^{(0)}(x)$ be original input image, then $i^{(1)}(x) = i^{(0)}(x)_\downarrow$ is the downsampled one, so we have $i^{(n+1)}(x) = i^{(n)}(x)_\downarrow, 0 \le n < N$. We can make the kernel smaller when N increases. The smallest kernel size (the size of Ω) is 3×3, the same choice in this article. We give the pseudo-code of the multiscale frame for bilateral filtering.

Input image: $i(x) = i^{(0)}(x)$.
Output image: $\tilde{i}(x)$.
1. Obtain $i^{(n)}(x), 0 \le n < N$ by downsampling steps.
2. For ($n > 0$)
      Obtain error: $e^{(n+1)}(x) = i^{(n+1)}(x) - \tilde{i}^{(n+1)}(x)$;
      Interpolation: $\hat{e}^{(n)}(x) = e^{(n+1)}(x)_\uparrow$;
      Compensation: $i^{(n)}(x) \leftarrow i^{(n)}(x) - \hat{e}^{(n)}(x)$;
      Filtering: $\tilde{i}^{(n)}(x) \leftarrow i^{(n)}(x) \otimes ke'$;
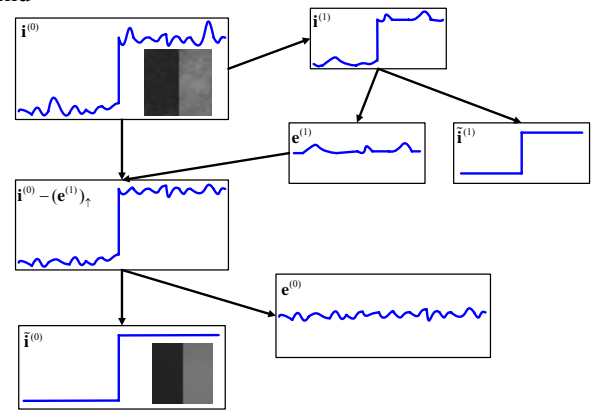 $n = n - 1$;
End

Fig. 2 Illustration of multiscale BF, it visualizes that image noise including high-frequency and low-frequency components which can be removed at different scales.

C. Multigrid based bilateral filtering

For a small ke', it is easy and not computational expensive to apply several iterations. Here we introduce a notation τ denoting the iterative times, so we have $i_\tau^{(n)}(x), 0 \le n < N, 1 \le \tau < \infty$, one iteration of the filter (1) (the case of discrete and kernel size 3×3) can be rewritten as:

$$i_{\tau+1}^{(n)}(x, y) = \sum_{-1 \le u, v \le 1} w_{(x,y)}^{(n)}(u, v) i_\tau^{(n)}(x+u, y+v) \quad (3)$$

Where

$$w_{(x,y)}^{(n)}(u, v) =$$

$$\frac{c((x,y),(x+u, y+v)) \cdot s\left(i_\tau^{(n)}(x, y), i_\tau^{(n)}(x+u, y+v)\right)}{\sum_{-1 \le u, v \le 1} c((x,y),(x+u, y+v)) \cdot s\left(i_\tau^{(n)}(x, y), i_\tau^{(n)}(x+u, y+v)\right)},$$

$$1 = \sum_{-1 \le u, v \le 1} w_{(x,y)}^{(n)}(u, v)$$

World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:5, No:7, 2011

Equation (3) is a typical adaptive smoothing filter with 3×3 window, and also a nonlinear diffusion equation evidently[19]. Reconsider (3), we have

$$i_{\tau+1}^{(n)}(x,y) - i_{\tau}^{(n)}(x,y) = \sum_{-1 \le u,v \le 1} w_{(x,y)}(u,v) i_{\tau}^{(n)}(x+u,y+v) - i_{\tau}^{(n)}(x,y)$$

$$= \frac{d i_{\tau}^{(n)}(x,y)}{d\tau}$$

(4)

Equation (4) can be expressed by a generalized form (5).

$$\frac{d\mathbf{I}^{(n)}}{d\tau} = \mathbf{A}^{(n)}(\mathbf{I}^{(n)})$$

(5)

Where $\mathbf{I}^{(n)}$ is the vector expression of $i_{\tau}^{(n)}(x,y)$, $\mathbf{A}^{(n)}(\square)$ is a nonlinear function of $\mathbf{I}^{(n)}$ derived from w. Now we find the original bilateral filtering (with large kernel) can be approximated by a multiscale nonlinear diffusion. The nature of equation (5) is to solve a nonlinear equation (6), which can be finished by relaxation iteration.

$$\mathbf{A}^{(n)}(\mathbf{u}^{(n)}) = 0$$

(6)

Equation (6) can be accelerated, it is easy to suggest the application of MG scheme[13], then the equation (6) at different scales can be viewed as (6) at different grids. Next we will demonstrate that a fast version of multiscale BF is equivalent to a full multigrid (FMG) based nonlinear diffusion filter. First we simply introduce the multigrid scheme.

Multigrid

Within the mathematical community, there has been widespread recent interest in multigrid methods[13]. Multigrid techniques have already been used to expedite relaxation problems in image processing[17]. The multigrid methods can be used to provide numerical solutions to the linear and nonlinear iterative problems of image processing. For linear problems of the form $\mathbf{A}(\mathbf{u}) = \mathbf{f}$, we denote by $\mathbf{v}$ an approximation to the exact solution $\mathbf{u}$ and by $\mathbf{e}$ the error, $\mathbf{e} = \mathbf{u} - \mathbf{v}$. Defining the residual to be $\mathbf{r} = \mathbf{f} - \mathbf{A}(\mathbf{v}) = \mathbf{A}(\mathbf{u} - \mathbf{v}) = \mathbf{A}(\mathbf{e})$. In brief, multigrid is the recursive application of a two-grid process. An iterative method, such as Gauss-Seidel or Jacobi relaxation is applied to the fine-grid problem. These iterations have the property that after relaxation the error will be smooth. At the fine grid, only high frequency error is eliminated quickly. The low frequency error can be accurately represented on a coarse grid. Since the coarse grid is much smaller than the fine grid, it is much less expensive to work on the coarse grid. The fine grid residual $\mathbf{r}$ is computed and restricted to the coarse grid $\mathbf{r}_{\downarrow} = P_{\downarrow}(\mathbf{r})$, $P_{\downarrow}$ is downsampling or restriction operator, where it is used as the right-hand side of the coarse grid residual equation $\mathbf{A}_{\downarrow}(\mathbf{e}_{\downarrow}) = \mathbf{r}_{\downarrow}$. This equation is solved, and the error thus determined is then interpolated back to the fine grid where it is used to correct the fine grid approximation, $\mathbf{v} \leftarrow \mathbf{v} + P_{\uparrow}(\mathbf{e}_{\downarrow})$, $P_{\uparrow}$ is interpolation operator. By recursively solving the coarse grid equation with this two-grid process, a multigrid algorithm is defined. This is called the multigrid V-cycle, as the algorithm starts with an initial estimate, telescopes down to the coarsest grid, and then returns in order to the finest grid[14].

In this article $\mathbf{A}(\square)$ is nonlinear and $\mathbf{A}(\mathbf{u} - \mathbf{v}) \ne \mathbf{f} - \mathbf{A}(\mathbf{v})$. There are two basic approaches for nonlinear MG. The first is to apply a linearization scheme, such as the Newton's method, and to employ multigrid for the solution of the Jacobian system in each iteration. The second is to apply multigrid directly to the nonlinear problem by employing the so called Full Approximation Scheme (FAS)[24]. In FAS a nonlinear iteration is applied to smooth the error. The full equation is solved on the coarse grid, after which the coarse-grid error is extracted from the solution. This correction is then interpolated and applied to the fine grid approximation. We adopt V-cycle MG and FAS schemes to accelerate the form of (6) for every scale in this article. For multiscale BF the initial input is $\mathbf{I}^{(N-1)}$, the result is the solution of $\mathbf{A}^{(0)}(\mathbf{I}^{(0)}) = 0$. In this paper, the size of test image is 640×480, we choose $N = 4$, so the input is $\mathbf{I}^{(3)}$. In the following we give detail operations for every scale.

Downsampling operator: $P_{\downarrow}$

The full-weighting restriction operator produces at a coarse-grid point a value that is just an average of the values at the corresponding fine-grid point and its eight nearest neighbors:

$$v_{\downarrow}(i,j) = \tfrac{1}{16}[v(2i-1,2j-1) + v(2i-1,2j+1) + v(2i+1,2j-1) + v(2i+1,2j+1)$$
$$+ 2(v(2i,2j-1) + v(2i,2j+1) + v(2i-1,2j) + v(2i+1,2j)) + 4v(2i,2j)]$$

Interpolation operator: $P_{\uparrow}$

The easiest and most natural for this problem are to use a linear interpolation operator. The linear interpolation operator can be defined by $\mathbf{v} = P_{\uparrow}(\mathbf{v}_{\downarrow})$, with components of $\mathbf{v}$ given.

$$v(2i,2j) = v_{\downarrow}(i,j),$$
$$v(2i+1,2j) = \tfrac{1}{2}(v_{\downarrow}(i,j) + v_{\downarrow}(i+1,j)),$$
$$v(2i,2j+1) = \tfrac{1}{2}(v_{\downarrow}(i,j) + v_{\downarrow}(i,j+1)),$$
$$v(2i+1,2j+1) = \tfrac{1}{4}(v_{\downarrow}(i,j) + v_{\downarrow}(i+1,j) + v_{\downarrow}(i,j+1) + v_{\downarrow}(i+1,j+1)).$$

Scale-3:

For scale-3, namely the coarsest grid, we only apply a few relaxations for solving equation (7).

$$\mathbf{A}^{(3)}(\mathbf{u}^{(3)}) = 0$$

(7)

Let the result of (7) is $\tilde{\mathbf{I}}^{(3)}$. For scale-3 it does not need compensation.

Scale-n<3:

For scale-n<3, it needs to solve equation (8).

$$\mathbf{A}^{(n)}(\mathbf{u}^{(n)}) = 0$$

(8)

The initial input is:

$$\mathbf{u}^{(n)}\big|_{\tau=0} = \mathbf{I}^{(n)} - \mathbf{e}^{(n)}, \quad \mathbf{e}^{(n)} = P_{\uparrow}(\mathbf{I}^{(n+1)} - \tilde{\mathbf{I}}^{(n+1)})$$

Use V-cycle multigrid (Fig.3(a)) and FAS to accelerate equation (8). Suppose we have found an approximation $\mathbf{v}$ for problem (9).

$$\mathbf{A}(\mathbf{u}) = \mathbf{f} = 0$$

(9)

The coarse-grid version of (9) is:

$$\mathbf{A}_{\downarrow}(\mathbf{v}_{\downarrow} + \mathbf{e}_{\downarrow}) - \mathbf{A}_{\downarrow}(\mathbf{v}_{\downarrow}) = \mathbf{r}_{\downarrow}$$

(10)

Where $\mathbf{v}_{\downarrow} = P_{\downarrow}(\mathbf{v}), \mathbf{r}_{\downarrow} = P_{\downarrow}(-\mathbf{A}(\mathbf{v}))$. $\mathbf{A}_{\downarrow}$ ($\ne P_{\downarrow}(\mathbf{A})$) is the coarser grid version of $\mathbf{A}$, but is not the downsampled result of $\mathbf{A}$. $\mathbf{A}_{\downarrow}(\square)$ are nonlinear function and determined by its variable and (σ_c, σ_s), for all scale-levels of (σ_c, σ_s) are same. Making these substitutions in the coarse-grid residual equation yields

World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:5, No:7, 2011

$$\mathbf{A}_\downarrow \underbrace{(\mathbf{v}_\downarrow + \mathbf{e}_\downarrow)}_{\mathbf{u}_\downarrow} = \underbrace{\mathbf{A}_\downarrow (\mathbf{v}_\downarrow) + \mathbf{r}_\downarrow}_{\mathbf{f}_\downarrow} \qquad (11)$$

The right side of this nonlinear system is known, and the equation is of the same form as the fine-grid equation (9). Assume we can find a solution to this system, which we denote $\mathbf{u}_\downarrow$. The coarse-grid error can be extracted from the solution by $\mathbf{e}_\downarrow = \mathbf{u}_\downarrow - \mathbf{v}_\downarrow$, and can then be interpolated up to the fine grid and used to correct the fine-grid approximation $\mathbf{v}$:

$$\mathbf{v} \leftarrow \mathbf{v} + P_\uparrow (\mathbf{e}_\downarrow)$$

Now it is not difficult to find that combining V-cycle MG and FAS with multiscale-BF (section 3.2) is equivalent to a nonlinear diffusion (5) accelerated by FMG (Fig.3(b)).
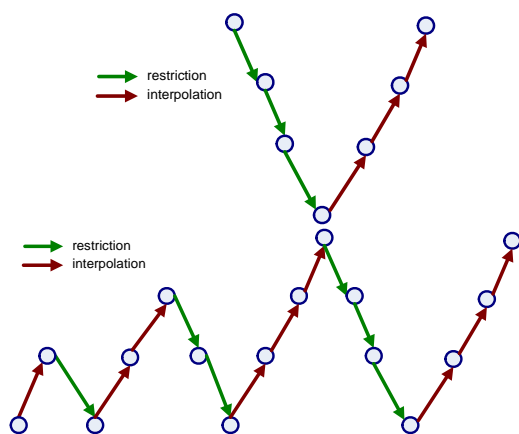

Fig.3 Schedule of grids for V-cycle (a) and FMG (b) schemes, all on four levels

IV. EXPERIMENTS

Fig.4-6 shows three experiments, the test data are LED, Cat and Piece-wise constant test images (the size is 640×480), the scale level is $N = 4$. For equation (7) we apply two iterations, for scale $N = 0,1,2$ we use V-cycle multigrid computation, only one relaxation is applied. Fig.4-6 shows the comparisons between two different ways of bilateral filtering. The result of using BF [10] and multigrid bilateral filtering (MGBF) is basically identical, but the time cost has large difference. Table.1 gives the computational performance comparison.

TABLE I
PERFORMANCE COMPARISON

Symbol	BF (one iteration)	MGBF
LED	Time=1.83s	Time=0.28s
Cat	Time=1.83s	Time=0.25s
Piece-wise constant test	Time=1.80s	Time=0.27s
image	SNR=17.3	SNR=46.8

V. CONCLUSION

In [19], a conclusion was given that the nature of bilateral filtering resembles that of anisotropic diffusion firstly, it use adaptive filter to build a novel link between bilateral filtering and nonlinear diffusion. But for numerical calculus, they are still two distinct computation ways: convolution based and

iteration based under the relationship by [19]. In this paper we give a new closer relationship between bilateral filtering and nonlinear diffusion: the bilateral filtering (1) can be approximated by a nonlinear diffusion based on FMG scheme. Additional contribution is that we propose a MG based bilateral filter, it is first time to show that bilateral filter can be accelerated by MG scheme. Fig.7 gives an visual expression and comparison between our works and [19]. In [19], the crucial link is that both bilateral filtering and nonlinear diffusion have same generalized form of adaptive filtering, but the new relationship from this paper focuses more on numerical calculus.
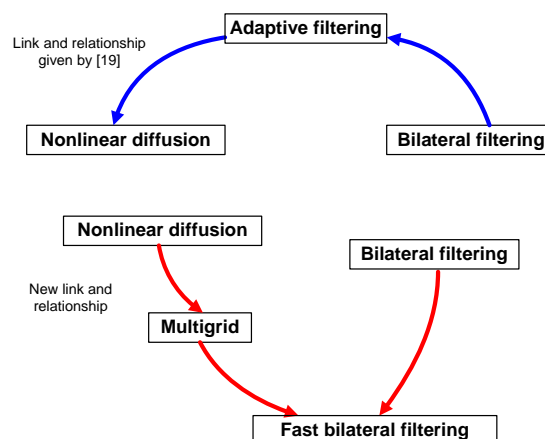

Fig.7 Visual expression and comparison of the new relationship and that from [19]

REFERENCES

[1] Z. Lin, , and Q. Shi. An anisotropic diffusion PDE for noise reduction and thin edge preservation. In *Proc. 10th Int. Conf. Image Analysis and Processing, IEEE Computer Society*, Los Alamitos, CA, 102-107, 1999.

[2] N. Sochen, R. Kimmel, and R. Malladi. A geometrical framework for low level vision. *IEEE Trans. Image Processing*, 7(3):310-318, 1998.

[3] J. Weickert. Anisotropic Diffusion in Image Processing, ser. *ECMI Series*. Stuttgart, Germany: Teubner, 1998.

[4] G. Sapiro and D. L. Ringach. Anisotropic diffusion of color images. *Proc. SPIE,* 471:382-2657, 1996.

[5] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans on PAMI.* 12(7):629-639, 1990.

[6] R. L. Lagendijk, J. Biemond, and D. E. Boekee. Regularized iterative image restoration with ringing reduction. *IEEE Trans. Acoust., Speech, Signal Processing*, 36(12):1874-1887, 1988.

[7] M. E. Zervakis. Nonlinear image restoration techniques. *Ph.D. dissertation*, Univ. Toronto, Toronto, ON, Canada, 1990.

[8] M. J. Black and G. Sapiro. Edges as outliers: Anisotropic smoothing using local image statistics. In *Scale-Space Theories in Computer Vision, Second International Conference, Scale-Space. Proceedings* (Lecture Notes in Computer Science), 1682: 259-270, 1999.

[9] M. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int. J. Comput. Vis.*, 19(1): 57-92, 1996.

[10] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. 6th Int. Conf. Computer Vision,* New Delhi, India, 839-846, 1998.

[11] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *Journal of Computational Physics*, 118(2):269-277, 1995.

[12] S. M. Smith and J. M. Brady. SUSAN – a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45-78, 1997(b).

[13] J. H. Bramble, Multigrid Methods. New York: Wiley, 1993.
[14] W. L. Briggs. A Multigrid Tutorial. Philadelphia, PA: SIAM, 1988.
[15] W. Hackbush and U. Trottenberg, Eds., Multigrid Methods. New York: Springer-Verlag, 1982.
[16] S. F. McCormick, Ed., Multigrid Methods. Philadelphia, PA: SIAM, 1987.
[17] P. Saint-Marc, J. Chen, and G. Medioni. Adaptive smoothing: A general tool for early vision. *IEEE Trans on PAMI*. 13(6):514-529, 1991.
[18] D. Terzopoulos. Image analysis using multigrid relaxation methods. *IEEE Trans on PAMI*. 8(2):129-139, 1986.
[19] Danny Barash. A Fundamental Relationship between Bilateral Filtering, Adaptive Smoothing and the Nonlinear Diffusion Equation. *IEEE Trans on PAMI*, 24(6):1-5, 2002.
[20] S. Paris, F. Durand. A fast approximation of the bilateral filter using a signal processing approach. *J. Comput. Vis.*, 81(1):24-52, 2009.
[21] Ming Zhang, Gunturk. B. K. Multiresolution Bilateral Filtering for Image Denoising. *IEEE Trans on Image processing*. 17(12):2324-2333, 2008.
[22] B. Weiss. Fast median and bilateral filtering. *Proc. SIGGRAPH*, 25(3):519-526, 2006.
[23] F. Porikli. Constant Time O (1) Bilateral Filtering. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR), 1-8, 2008.
[24] Van E. H. Multigrid methods nonlinear problems: an overview. *Proc. SPIE*, 5016:36-38, 2003.

World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
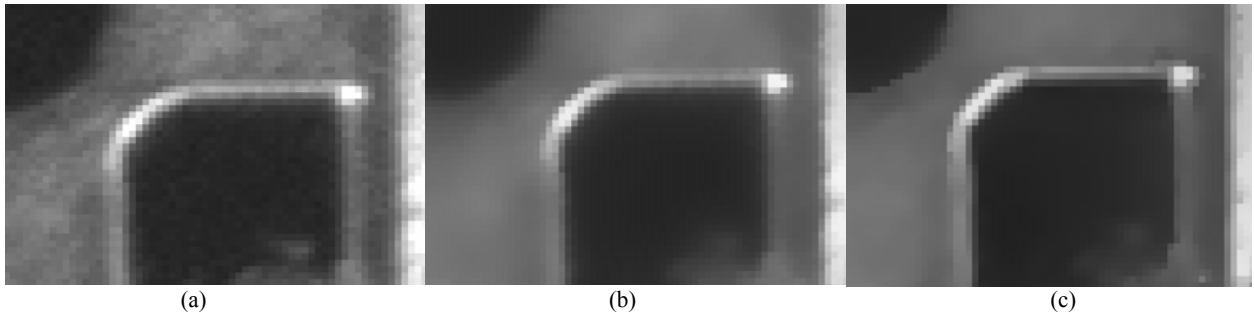Vol:5, No:7, 2011

(a)   (b)   (c)

Fig.4 (a) the original LED image; (b) the result of (a) by BF (one iteration), the time cost is 1.83s, the kernel size is 33×33 and $(\sigma_c, \sigma_s) = (2.5, 31.62)$; (c) the result of (a) by MGBF, the time cost is 0.28s, $(\sigma_c, \sigma_s) = (2.5, 16)$.
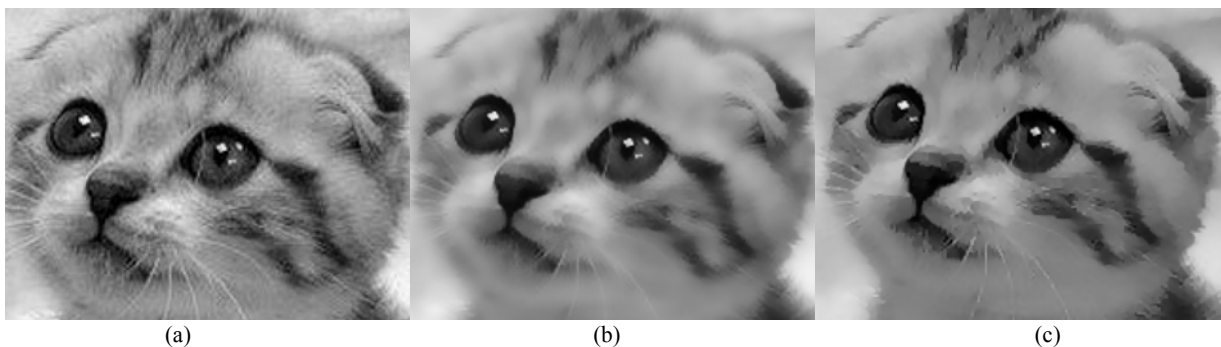
LED

(a)   (b)   (c)

Fig.5 (a) the original Cat image; (b) the result of (a) by BF (one iteration), the time cost is 1.83s, the kernel size is 33×33 and $(\sigma_c, \sigma_s) = (2.5, 31.62)$; (c) the result of (a) by MGBF, the time cost is 0.25s, $(\sigma_c, \sigma_s) = (2.5, 10)$.

Cat

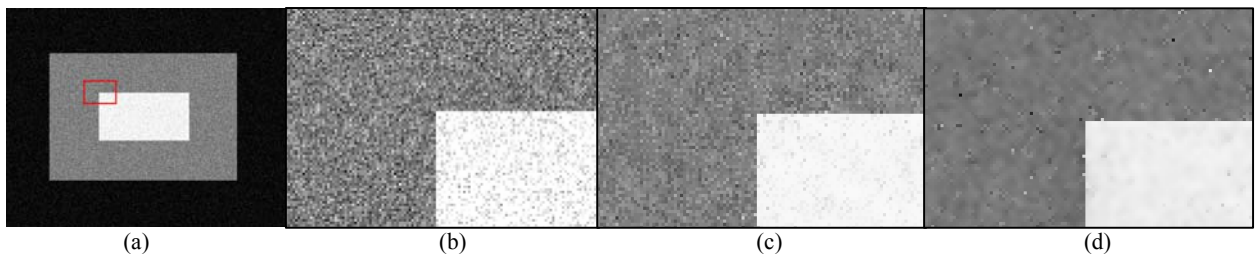(a)   (b)   (c)   (d)

Fig.6 (a) the original test image; (b) local details from (a) within the small red frame; (c) the result by BF (one iteration), the time cost is 1.80s, SNR=17.3, the kernel size is 33×33 and $(\sigma_c, \sigma_s) = (2.5, 31.62)$; (d) the result by MGBF, the time cost is 0.27s, SNR=46.8, $(\sigma_c, \sigma_s) = (2.5, 10)$.

Piece-wise constant test image