Adaptive Guided Image Filtering for Sharpness Enhancement and Noise Reduction

Cuong Cao Pham, Synh Viet Uyen Ha, and Jae Wook Jeon*

School of Information and Communication Engineering,
Sungkyunkwan University, Suwon, South Korea
cuongpc@skku.edu, synhha@ece.skku.edu, jwjeon@yurim.skku.ac.kr
http://micro.skku.ac.kr

Abstract. Sharpness enhancement and noise reduction play crucial roles in computer vision and image processing. The problem is to enhance the appearance and reduce the noise of the digital images without causing halo artifacts. In this paper, we propose an adaptive guided image filtering (AGF) able to perform halo-free edge slope enhancement and noise reduction simulaneously. The proposed method is developed based on guided image filtering (GIF) and the shift-variant technique, part of adaptive bilateral filtering (ABF). Experiments showed the results produced from our method are superior to those produced from unsharp masking-based techniques and comparable to ABF filtered output. Our proposed AGF outperforms ABF in terms of computational complexity. It is implemented using a fast and exact linear-time algorithm.

Keywords: Edge-preserving smoothing, guided image filter, sharpness enhancement, noise reduction.

1 Introduction

Enhancing the sharpness and reducing the noise of the digital images have attracted much interest during the last decades. These pre-processing techniques play crucial roles in computer vision and image processing. However, how to simultaneously reduce noise and increase the slope of edges without creating halo artifacts is still a challenging issue.

Conventional linear filter effectively smooths noise in homogeneous regions, however, blurring the edges of an image. Conversely, edge-preserving smoothing techniques only filter noise, while preserving edge structures. Existing techniques that feasibly perform this kind of operation include anisotropic diffusion (AD) [14], bilateral filtering (BLF) [17] and guided image filtering (GIF) [9]. However,

* This research was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC (Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency)" (NIPA-2011-(C1090-1121-0008)), and by Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology(2011-0018397).

Y.-S. Ho (Ed.): PSIVT 2011, Part I, LNCS 7087, pp. 323–334, 2011.

none of them can be directly applied to achieve sharpness enhancement and noise reduction simultanenously, as is our stated goal.

Anisotropic diffusion is able to preserve and sharpen edges, but both noise and fine details are unexpectedly removed due to its over-smooth characteristic. Although BLF is widely used and has become the *de facto* standard for computer vision and image processing, its ability to enhance the sharpness of an image is limited. While GIF proposed in [9] outperforms BLF in a variety of computer vision applications, it shares the same limitation as does BLF.

In terms of image sharpening, the unsharp masking technique (USM) is popularly used due to its simplicity. A high-pass filter (HPF) is applied to the input image under the guidance of the unsharp mask, obtained by subtracting the input image and its blurred version. Thus, the contrast along the edges is increased in the sharpened output. However, as discussed in [3,10,19], USM has two major drawbacks. First, the overshoot and undershoot artifacts occur around the edges of the sharpened image due to the large boost of high contrast areas. Second, HPF not only enhances the edges but also significantly amplifies the noise in the input image. This reduces image quality.

Investigations have been conducted to improve these two limitations of USM [3,10,15]. Especially, Kim et al. proposed the optimal unsharp mask (OUM) [10] to reduce noise in the homogeneous regions, while achieving the equivalent level of sharpness as USM does. The Laplacian of Gaussian (LoG) filter is used to determine the locally adaptive optimal λ value, instead of a fixed λ for HPF. However, the halo artifacts have not been overcome completely.

In summary, state-of-the-art edge-preserving smoothing techniques cannot be used to achieve the goal directly, while unsharp masking-based approaches create overshoot and undershoot artifacts during the sharpening process. In a notable recent work, Zhang et al. [19] made use of the shift-variant technique to propose an adaptive bilateral filter (ABF) able to enhance the sharpness and remove the noise simultaneously. Unfortunately, the introduction of locally adaptive optimal parameters make this approach infeasible to fully adapt with the existing BLF acceleration schemes. It must be implemented using the two nested loops brute-force approach, whose computational complexity is $O(|w|^2)$, where $|w|$ is the size of the filter kernel.

In this paper, we present adaptive guided image filtering (AGF) for image sharpening and de-noising. Our proposed AGF method is based on GIF and the shifting technique proposed in [19]. The optimal training parameters produced from [19] are slightly modified and reused in our method to visually compare to ABF and OUM. However, we will prove the participation of these adaptive parameters does not corrupt the acceleration scheme of GIF - the O(N) time exact algorithm can still be applied to achieve the speed up. Experiments show the results produced from our method are superior to those produced from USM and OUM and comparable to ABF filtered results.

The remainder of this paper is organized as follows. Section 2 presents the connection between bilateral filter and guided image filter. Section 3 examines the adaptive bilateral filter with the shift-variant technique and adaptive optimal parameters. Section 4 presents the adaptive guided image filtering using the shift-variant technique. Section 5 presents the experimental results to compare our method to methods from the literature. Finally, this paper is drawn to a conclusion and future work outlined in Section 6.

2 Bilateral and Guided Image Filtering

In this section, we present the relationship between BLF and GIF in terms of the filter kernel. These two edge-preserving smoothing techniques play a central role in ABF and our proposed AGF.

2.1 Bilateral Filtering

As we briefly mentioned above, BLF is widely used due to its appealing characteristics. The name bilateral filter was first termed in [17] based on the work [1,16]. It is a non-iterative, non-linear filter that smooths low gradient regions, while preserving strong edges. Each output pixel is computed as a weighted mean of its neighbors. The weight is computed based on the spatial domain, like other linear filters, and on the intensity range domain. Let I_p be the intensity value at pixel p, w_k be the kernel window centered at pixel k, BLF is given by:

$$BLF(I)_p = \frac{1}{\sum\limits_{q \in w_k} W_{BLF_{pq}}(I)} \sum_{q \in w_k} W_{BLF_{pq}}(I) I_q \tag{1}$$

where the division term normalizes the weights sum to 1 and the kernel weights function $W_{BLF_{pq}}(I)$ can be expressed by:

$$W_{BLF_{pq}}(I) = \exp\left(-\frac{\|p-q\|^2}{2\sigma_s^2}\right) \exp\left(-\frac{|I_p - I_q|^2}{2\sigma_r^2}\right) \tag{2}$$

where the standard deviations parameters σ_s and σ_r control the decrement of weights in the spatial and intensity range domains, respectively. Each domain is represented by a Gaussian function. The spatial domain gives higher weight to pixels closer to the center pixel, whilst lower weight is assigned to distant pixels. Correspondingly, the same rule can be applied to the intensity range domain. Higher or lower weight will be assigned to the pixels that are similar to or different from the center pixel in terms of intensity value. The degree of smoothing can be adjusted by changing the value of σ_r. In most applications, this value must be sufficiently small to avoid filtering meaningful features, because BLF becomes equivalent to the Gaussian filter when σ_r increases.

Excessive time consumption is one of BLF's disadvantages, although it is efficient to implement. The brute-force approach consists of two nested loops. The computational complexity is $O(|w|^2)$, where $|w|$ is the size of the spatial domain. Studies have investigated reducing the time-taken [5,7,11,13]. The main concepts of these acceleration schemes can be found in [12]. Notably, the fast approximation approach proposed in [11] has been proved to be a very useful technique. It has been applied to a variety of bilateral-based applications [12].

2.2 Guided Image Filtering

He et al. [9] proposed GIF to overcome the gradient reversal artifacts occurring, using BLF in detail manipulation technique that is not mentioned in this paper. Instead, we focus on its ability of edge-preserving and fast implementation. It has been analyzed and proved that GIF shares the good edge-preserving characteristic compared to BLF. Furthermore, its fast and exact linear-time algorithm outperforms BLF in terms of computational complexity.

The filtering process of GIF is originally done under the guidance of an image G that can be another image or the input image I itself. It is similar to the joint bilateral filter [12] which is used to denoise the no-flash image I using the flash image G. When I and G are identical, joint bilateral filter becomes bilateral filter naturally. We first express GIF in terms of the filter kernel to establish the connection between BLF and GIF. Let I_p and G_p be the intensity value at pixel p of the input and guided image, w_k be the kernel window centered at pixel k, to be consistent with BLF. GIF is then formulated by:

$$GIF(I)_p = \frac{1}{\sum\limits_{q \in w_k} W_{GIF_{pq}}(G)} \sum_{q \in w_k} W_{GIF_{pq}}(G) I_q \tag{3}$$

where the kernel weights function $W_{GIF_{pq}}(G)$ can be expressed by:

$$W_{GIF_{pq}}(G) = \frac{1}{|w|^2} \sum_{k:(p,q) \in w_k} \left(1 + \frac{(G_p - \mu_k)(G_q - \mu_k)}{\sigma_k^2 + \varepsilon}\right) \tag{4}$$

where μ_k and σ_k^2 are the mean and variance of guided image G in local window w_k, $|w|$ is the number of pixels in this window. The key to understanding the edge-preserving ability of GIF lies in the term $1+[(G_p - \mu_k)(G_q - \mu_k)]/(\sigma_k^2 + \varepsilon)$ in this equation. When both G_p and G_q are concurrently on the same side of an edge (smaller or larger than the mean), the weight assigned to pixel q is large. Conversely, a small weight will be assigned to pixel q when they are on different sides (one is smaller and one is larger than the mean). Some further computations in [9] confirm the normalization term in equation (3) equals 1. The filter kernel of GIF can be shortened as follows:

$$GIF(I)_p = \sum_{q \in w_k} W_{GIF_{pq}}(G) I_q \tag{5}$$

The degree of smoothing of GIF is adjusted via parameter ε. The larger the value of ε is, the smoother the filtered image will be. It plays an equivalent role to σ_r in BLF. Some further experiment and demonstration in [9] prove that BLF and GIF yield approximately equivalent smoothing results, by setting $\varepsilon = \sigma_r^2$ in the normalized $[0; 1]$ intensity range value. Of course, the guided image G is identical to the input image I in this relation. This property is crucial, because it is going to be used to convert the optimal parameters of ABF to our proposed AGF, as shown in Section 4.

The $O(N)$ exact algorithm of GIF is performed by applying a chain of box filters using the $O(N)$ time integral image technique [6]. The linear translation-variant takes the place of the filter kernel (4) when computing this fast and exact linear-time algorithm. We will discuss this issue in more detail in Section 4.2.

3 Adaptive Bilateral Filtering

In this section, we examine ABF for sharpness enhancement and noise removal. We mainly focus on the shift-variant technique, because it will be applied to our method. This method was proposed in [19] based on the work [18]. The main differences of ABF compared to BLF is the introduction of the shifting technique and locally adaptive optimal parameters. These modifications make ABF outperforms conventional BLF in terms of image sharpening and de-noising. The filter kernel and weighting function of ABF are expressed by:

$$ABF(I)_p = \frac{1}{\sum\limits_{q\in w_k} W_{ABF_{pq}}(I)} \sum_{q\in w_k} W_{ABF_{pq}}(I) I_q \tag{6}$$

$$W_{ABF_{pq}}(I) = \exp\left(-\frac{\|p-q\|^2}{2\sigma_s^2}\right)\exp\left(-\frac{|(I_p+\xi_p)-I_q|^2}{2\sigma_r^2}\right) \tag{7}$$

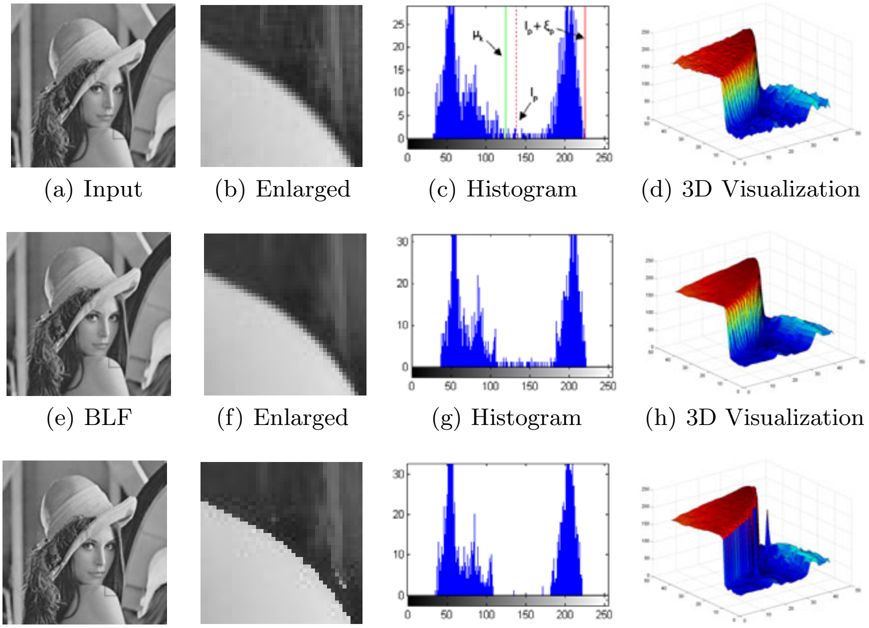
where ξ_p is the introduced offset that enables ABF to sharpen the image. The näive strategy for choosing this value is guided by:

$$\xi_p = \begin{cases} \text{MAX}(w_k) - I_p & \text{if } \Delta_p > 0 \\ \text{MIN}(w_k) - I_p & \text{if } \Delta_p < 0 \\ 0 & \text{if } \Delta_p = 0 \end{cases} \tag{8}$$

where $\Delta_p = I_p - \mu_k$ is the intensity difference between pixel p and the mean of local window w_k. While $\text{MAX}(w_k)$ and $\text{MIN}(w_k)$ are the maximum and minimum values of local window w_k, respectively.

This strategy is due to the histogram analysis, as shown in Fig. 1. For an input image shown in Fig. 1(a), the histogram and 3-D visualization of its enlarged window (Fig. 1(b)) are shown in Fig. 1(c) and 1(d), respectively. For the conventional BLF, the intensity range domain normally computes the affinities between the center pixel p and its neighbors q. This center value I_p is represented by the dotted-red line in the histogram. Thus, the slope of the edge in the filtered output is only just preserved, but not sharpened. The second row of Fig. 1 represents the corresponding conventional BLF output. In contrast, the edge is extremely enhanced by applying ABF with the näive offset choosing strategy. The center value I_p has been shifted to $\text{MAX}(w_k)$ (red line), because its intensity value (dotted-red line) is larger than the mean μ_k (green line). However, as we can see in the third row, the aliasing effect and unexpected outliers occur in the sharpened output.

Zhang et al. proposed a more reliable strategy for choosing offset value to overcome this problem. They estimated both the offset ξ and standard deviation



(a) Input (b) Enlarged (c) Histogram (d) 3D Visualization

(e) BLF (f) Enlarged (g) Histogram (h) 3D Visualization

(i) Näive ABF (j) Enlarged (k) Histogram (l) 3D Visualization

Fig. 1. Illustration of the effect of ABF ($\sigma_s = 1.0, \sigma_r = 20$) with the näive offset choosing strategy compared to conventional BLF ($\sigma_s = 1.0, \sigma_r = 20$)

σ_r of the intensity range domain via a training procedure. Given the N sets of training images, where each set S consists of a high-quality original image I, a degraded image J and its restored output $\hat{J}$, the optimal parameters are obtained by solving the following minimum mean squared error estimation problem:

$$\{\xi_i^*, \sigma_{r,i}^*\} = \underset{\{\xi_i, \sigma_{r,i}\}}{\arg\min} \sum_{n=1}^{N} \left\| I_{n,p} - \hat{J}_{n,p} \right\|_{S^{(n)}}^2 \tag{9}$$

where $i = 1, 2, \ldots, T$ is the pixel classified number obtained by applying a 9×9 Laplacian of Gaussian filter (LoG) with $\sigma_{LoG} = 1.5$. The resultant parameters are locally adaptive, making ABF more robust. Zhang et al. [19] show how to find these parameters; we refer the reader to their paper for further details.

The main concern when applying ABF with optimal parameters is the large computational cost of its brute-force implementation. The standard deviation σ_r must be fixed in order to accelerate it using the method [11]. Otherwise, it will degrade the 3-D convolution model of method [11] when applying both adaptive offset and standard deviation parameters. Our proposed AGF with the use of these optimal parameters can achieve comparable results to ABF, while still keeping the generality of the linear translation-variant of the GIF. That is, the exact and linear-time algorithm is easily applied to achieve the acceleration.

4 Proposed Adaptive Guided Image Filtering for Image Sharpening and De-noising

4.1 Proposed Adaptive Guided Image Filtering

In this section, we present our proposed method using the shifting technique. As we have seen when we analyzed the relationship between BLF and GIF in terms of the filter kernel in Section 2, the main difference between them lies in their weighting functions of the filter kernel, as shown in equation (2) and (4). However, the intensity range domain of BLF and kernel function of GIF are similar in principle, because each of them takes the intensity value of center pixel p, local neighbors q and a smoothing parameter (σ_r in BLF, ε in GIF) in the computation process.

This is based on the shifting technique of ABF, in which the offset ξ_p is added to the intensity value of center pixel p in the intensity range domain of BLF. The same strategy is applied to our proposed AGF - the offset is added to the intensity value of center pixel p in the kernel weights function of GIF. Formally, the filter kernel and weighting function of our proposed AGF are given by:

$$AGF(I)_p = \sum_{q \in w_k} W_{AGF_{pq}}(G) I_q \tag{10}$$

$$W_{AGF_{pq}}(G) = \frac{1}{|w|^2} \sum_{k:(p,q) \in w_k} \left(1 + \frac{\left((G_p + \xi'_p) - \mu_k \right) (G_q - \mu_k)}{\sigma_k^2 + \varepsilon} \right) \tag{11}$$

where ξ'_p is the added offset and ε is the smoothing parameter. The näive offset choosing strategy is also applied to our proposed AGF, as does ABF. That is:

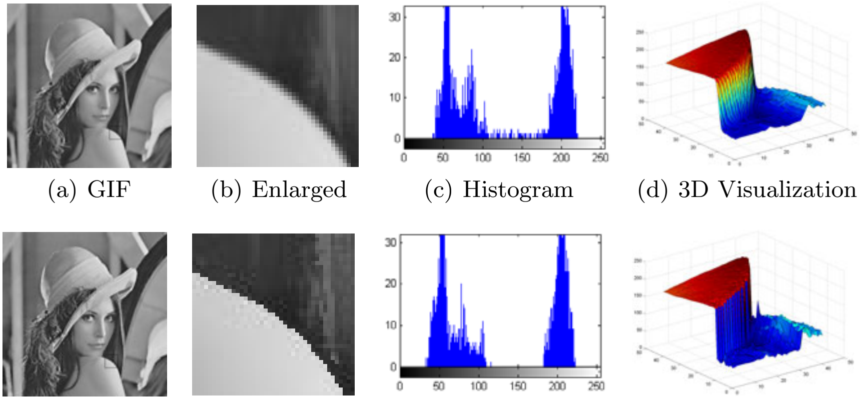
$$\xi'_p = \begin{cases} \text{MAX}(w_k) - G_p & \text{if } \Delta'_p > 0 \\ \text{MIN}(w_k) - G_p & \text{if } \Delta'_p < 0 \\ 0 & \text{if } \Delta'_p = 0 \end{cases} \tag{12}$$

where the intensity difference is defined by $\Delta'_p = G_p - \mu_k$.

The same histogram analysis is applied to GIF and our proposed AGF, as shown in Fig. 2. GIF only preserves the edges during the smoothing process, while the sharpened result produced from our proposed AGF with the näive offset choosing strategy contains the aliasing effect and unexpected outliers, as did näive ABF. In order to achieve the better result by applying the adaptive optimal parameters produced from [19], the values of ε in AGF need to be computed based on the corresponding optimal values of σ_r in ABF. In Section 2, we showed these two parameters can be converted by the following expression:

$$\varepsilon = \sigma_r^2 / 255 \tag{13}$$

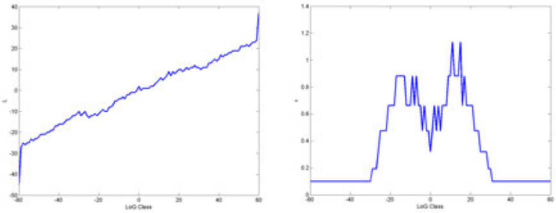
where both ε and σ_r are in the range $[0; 255]$ intensity value. Fig. 3 shows the corresponding offset and converted epsilon values we will use in AGF. The offset tends to be unchanged. However, to make sure the term $G_p + \xi'_p$ is still within the range $[\text{MIN}(w_k); \text{MAX}(w_k)]$, it is constrained by the following equation:



(a) GIF (b) Enlarged (c) Histogram (d) 3D Visualization

(e) näive AGF (f) Enlarged (g) Histogram (h) 3D Visualization

Fig. 2. Illustration of the effect of our proposed AGF ($\varepsilon = 1.5686$) with the näive offset choosing strategy compared to conventional GIF ($\varepsilon = 1.5686$)



(a) Offset ξ (b) Epsilon ε

Fig. 3. Optimal offset and converted epsilon correspoding to each LoG class

$$\xi'_p = \begin{cases} \mathrm{MAX}(w_k) - G_p & \text{if } \mathrm{A}_p > \mathrm{MAX}(w_k) \\ \mathrm{MIN}(w_k) - G_p & \text{if } \mathrm{A}_p < \mathrm{MIN}(w_k) \\ \xi_p & \text{otherwise} \end{cases} \tag{14}$$

where ξ_p is the optimal offset obtained from [19] and $\mathrm{A}_p = G_p + \xi_p$. It's noted that, each pixel p is classified by the corresponding LoG class number obtained by applying a LoG filter. The rounded LoG class is limited within the range $[-60; 60]$ as does ABF.

4.2 Linear Transform Model of AGF

In this section, we present AGF in terms of the linear translation-variant, because the $O(N)$ time exact algorithm takes advantage of this model to implement it. First, we will show the linear transform model of GIF, and then apply it to the proposed AGF. As described in [9], the filtered output $\hat{I}$ of GIF is represented by a linear transform of guided image G within a local window w_k centered at pixel k as follows:

$$\hat{I}_p = a_k G_p + b_k, \forall p \in w_k \tag{15}$$

where a_k and b_k are constant linear coefficients determined by solving the optimization problem that seeks to minimize the difference between the output and input image. Formally, it is expressed by:

$$E(a_k, b) = \sum_{p \in w_k} \left((a_k G_p + b_k - I_p)^2 + \varepsilon_k a_k^2 \right) \tag{16}$$

where ε_k is unchanged over the entire image. It controls the degree of smoothing of GIF. These coefficients are formally determined using linear regression method:

$$a_k = \frac{\frac{1}{|w|} \sum_{p \in w_k} G_p I_p - \mu_k \bar{I}_k}{\sigma_k^2 + \varepsilon_k} \tag{17}$$

$$b_k = \bar{I}_k - a_k \mu_k \tag{18}$$

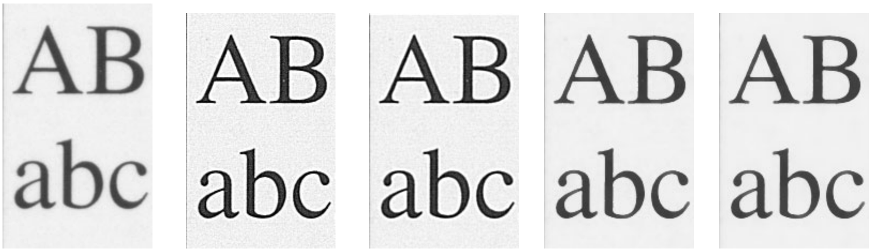
where $\bar{I}_k$ is the mean of I in w_k. To ensure the value of $\hat{I}_p$ does not vary when computed in different windows, the final output is computed by:

$$\hat{I}_p = \left(\frac{1}{|\mathrm{w}|} \sum_{k \in w_p} a_k \right) G_p + \left(\frac{1}{|w|} \sum_{k \in w_p} b_k \right) \tag{19}$$

For our proposed AGF, the question is how to include the adaptive optimal parameters into the linear transform-variant of the GIF. First, we can clearly see the varying adaptive ε^* obviously fits well to equation (17) when computing linear coefficient a_k. Second, the function

$$\hat{I}_p = \left(\frac{1}{|\mathrm{w}|} \sum_{k \in w_p} a_k \right) (G_p + \xi'_p) + \left(\frac{1}{|w|} \sum_{k \in w_p} b_k \right) \tag{20}$$

is the linear transform model of AGF with the participation of the adaptive offset. The appendix presented at the end of this paper shows the correspondence between this linear transform model and its filter kernel expressed in equation (10) and (11). Hence, the algorithm can be implemented by applying a chain of box filters using $O(N)$ integral image technique, as does GIF.

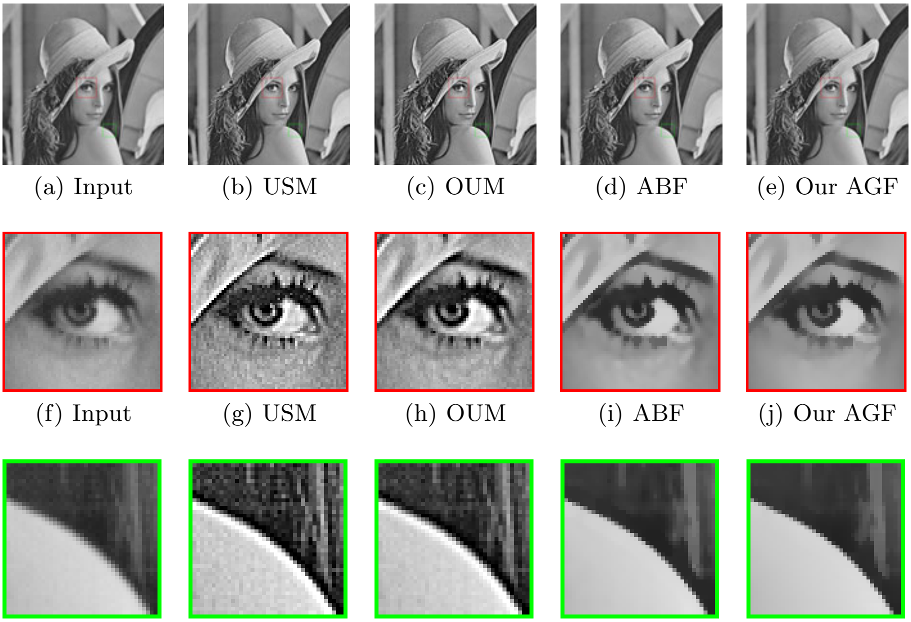


(a) Input text (b) USM (c) OUM (d) ABF (e) Our AGF

Fig. 4. Scanned text image rendered by our proposed AGF and existing methods. Parameters are configured as follows: (b) USM: $r = 5, \lambda = 4$; (c) OUM: $r = 5$; (d) ABF: $r = 3, \sigma_s = 1.0, r_{LoG} = 4$; (e) AGF: $r = 3, r_{LoG} = 4$.

5 Experimental Results

We evaluate the performance of AGF and existing methods with a scanned text image and the Lena image. The text image scanned at 600 dpi was obtained from [19] and cropped due to space limitations. For the text image, as shown in Fig. 4, the contrast of restored outputs produced from USM and OUM increase; but visible halos occur around the edges. Conversely, restored texts produced from ABF and our AGF do not suffer such artifacts, and the contrast is nearly identical to that of the input image. For the Lena image, the difference between these methods can be seen more clearly, as shown in Fig. 5. USM produces visible halos around the edges, and the noise is also significantly enhanced. OUM reduces noise but suffers from the artifacts. Both ABF and our AGF with the use of optimal adaptive parameters effectively remove noise and significantly enhance the sharpness. We used a PC with an AMD Athlon 64 X2 Dual Core Processor 3800+ 2.00 Ghz to measure the processing time of both AGF and ABF with a kernel radius $r = 5$. Our proposed AGF takes about 1.4s to process a 1-megapixel gray-scale image, while the $O(|w|^2)$ time ABF [19] takes about 12.7s to process it.



(a) Input (b) USM (c) OUM (d) ABF (e) Our AGF

(f) Input (g) USM (h) OUM (i) ABF (j) Our AGF

(k) Input (l) USM (m) OUM (n) ABF (o) Our AGF

Fig. 5. Lena image rendered by our proposed AGF and existing methods. Parameters are configured as follows: (b), (g), (l) USM: $r = 5, \lambda = 4$; (c), (h), (m) OUM: $r = 5$; (d), (i), (n) ABF: $r = 3, \sigma_s = 1.0, r_{LoG} = 4$; (e), (j), (o) AGF: $r = 3, r_{LoG} = 4$.

6 Conclusion

In this paper, we presented an adaptive guided image filtering (AGF) for sharpness enhancement and noise reduction. The proposed method is developed based on guided image filtering and the shift-variant technique. The relationship between the conventional bilateral filter and the guided image filter is presented to convert optimal parameters from ABF to our proposed AGF.

Experiments showed the results produced from our method to be superior to those produced from unsharp masking-based techniques and comparable to ABF filtered output. It effectively removes noise and sharpens the edges simultaneously, without producing overshoot and undershoot artifacts as the ideal approach. Our method outperforms ABF in terms of computation cost, where the computational complexity is $O(N)$ compared to $O(|w|^2)$ of ABF.

References

1. Aurich, V., Weule, J.: Non-linear gaussian filters performing edge preserving diffusion. In: Proceedings of the DAGM Symposium, pp. 538–545 (1995)
2. Barash, D.: A Fundamental Relationship Between Bilateral Filtering, Adaptive Smoothing, and the Nonlinear Diffusion Equation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(6), 844–847 (2002)
3. Bilcu, R.C., Vehvilainen, M.: Constrained Unsharp Masking for Image Enhancement. In: Proc. of Intl. Conf. on Image and Signal Processing, pp. 10–19 (2008)
4. Buades, A., Coll, B., Morel, J.M.: The staircasing effect in neighborhood filters and its solution. IEEE Trans. Image Processing 15(6), 1499–1505 (2006)
5. Chen, J., Paris, S., Durand, F.: Real-time edge-aware image processing with the bilateral grid. ACM Transactions on Graphics 26(3) (2007)
6. Crow, F.C.: Summed-area tables for texture mapping. In: SIGGRAPH (1984)
7. Durand, F., Dorsey, J.: Fast Bilateral Filtering for the Display of High-Dynamic-Range Images. ACM Transactions on Graphics 21(3), 257–266 (2002)
8. Elad, M.: On the bilateral filter and ways to improve it. IEEE Transactions on Image Processing 11(10), 1141–1151 (2002)
9. He, K., Sun, J., Tang, X.: Guided Image Filtering. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 1–14. Springer, Heidelberg (2010)
10. Kim, S., Allebach, J.P.: Optimal unsharp mask for image sharpening and noise removal. Journal of Electronic Imaging 14, 023007-1–023007-13 (2005)
11. Paris, S., Durand, F.: A Fast Approximation of the Bilateral Filter using a Signal Processing Approach. International Journal of Computer Vision 81(1), 24–52 (2009)
12. Paris, S., Kornprobst, P., Tumblin, J., Durand, F.: Bilateral Filtering: Theory and Applications. In: Foundations and Trends in Computer Graphics and Vision (2009)
13. Pham, T.Q., Van Vliet, L.J.: Separable bilateral filtering for fast video preprocessing. In: Proceedings of the IEEE Intl. Conf. on Multimedia and Expo (2005)
14. Penora, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–639 (1990)

15. Polesel, A., Ramponi, G., Mathews, V.G.: Image Enhancement via Addaptive Unsharp Masking. IEEE Trans. Image Processing 9(3), 505–510 (2000)
16. Smith, S.M., Brady, J.M.: SUSAN - A new approach to low level image processing. International Journal of Computer Vision 23(1), 45–78 (1997)
17. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the IEEE Intl. Conf. on Computer Vision (ICCV), pp. 839–846 (1998)
18. Zhang, B., Allebach, J.P.: Adaptive Bilateral Filter for Sharpness Enhancement and Noise Removal. In: Proc. Intl. Conf. on Image Processing (ICIP), vol. 4, pp. 417–420 (2007)
19. Zhang, B., Allebach, J.P.: Adaptive Bilateral Filter for Sharpness Enhancement and Noise Removal. IEEE Transactions on Image Processing 17(5), 664–678 (2008)

Appendix: Derivative of the AGF Filter Kernel

This is based on the proof that shows the filter kernel of GIF corresponds to its linear translation-variant in [9], we shortly present the correspondence between the filter kernel and linear transform model of AGF with the introduction of optimal offset ξ^* and ε^* in this part.

First, we rewrite equation (10) by $\hat{I}_p = \sum_{q \in w_k} W_{AGF_{pq}}(G) I_q$. So, the filter kernel $W_{AGF_{pq}}(G)$ is computed by taking the partial derivative of $\hat{I}_p$ with respect to I_q. Formally, it is expressed by:

$$W_{AGF_{pq}}(G) = \frac{\partial \hat{I}_p}{\partial I_q} \tag{21}$$

Replacing b_k in (20) by (18), we have:

$$\hat{I}_p = \frac{1}{|w|} \sum_{k \in w_p} \left[a_k \left((G_p + \xi'_p) - \mu_k \right) + \bar{I}_k \right] \tag{22}$$

So, the partial derivative of $\hat{I}_p$ with respect to I_q is formulated by:

$$\frac{\partial \hat{I}_p}{\partial I_q} = \frac{1}{|w|} \sum_{k \in w_p} \left[\frac{\partial a_k}{\partial I_q} \left((G_p + \xi'_p) - \mu_k \right) + \frac{\partial \bar{I}_k}{\partial I_q} \right] \tag{23}$$

From [9], we already had:

$$\frac{\partial a_k}{\partial I_q} = \frac{1}{\sigma_k^2 + \varepsilon_k} \left(\frac{1}{|w|} G_q - \frac{1}{|w|} \mu_k \right) \delta_{k \in w_q} \tag{24}$$

$$\frac{\partial \bar{I}_k}{\partial I_q} = \frac{1}{|w|} \delta_{q \in w_k} = \frac{1}{|w|} \delta_{k \in w_q} \tag{25}$$

where $\delta_{q \in w_k}$ equals 1 when q is in w_k, and equals 0 otherwise.

Placing (24) and (25) into (23), we get:

$$\frac{\partial \hat{I}_p}{\partial I_q} = \frac{1}{|w|^2} \sum_{k:(p,q) \in w_k} \left(1 + \frac{\left((G_p + \xi'_p) - \mu_k \right) (G_q - \mu_k)}{\sigma_k^2 + \varepsilon_k} \right) \tag{26}$$

This is exactly the filter kernel $W_{AGF_{pq}}(G)$ that we expressed in equation (11).