

Adaptive Bilateral Filter for Sharpness Enhancement and Noise Removal

Buyue Zhang, *Member, IEEE*, and Jan P. Allebach, *Fellow, IEEE*

Abstract—In this paper, we present the adaptive bilateral filter (ABF) for sharpness enhancement and noise removal. The ABF sharpens an image by increasing the slope of the edges without producing overshoot or undershoot. It is an approach to sharpness enhancement that is fundamentally different from the unsharp mask (USM). This new approach to slope restoration also differs significantly from previous slope restoration algorithms in that the ABF does not involve detection of edges or their orientation, or extraction of edge profiles. In the ABF, the edge slope is enhanced by transforming the histogram via a range filter with adaptive offset and width. The ABF is able to smooth the noise, while enhancing edges and textures in the image. The parameters of the ABF are optimized with a training procedure. ABF restored images are significantly sharper than those restored by the bilateral filter. Compared with an USM based sharpening method—the optimal unsharp mask (OUM), ABF restored edges are as sharp as those rendered by the OUM, but without the halo artifacts that appear in the OUM restored image. In terms of noise removal, ABF also outperforms the bilateral filter and the OUM. We demonstrate that ABF works well for both natural images and text images.

Index Terms—Bilateral filter, de-blurring, noise removal, range filter, sharpness enhancement, slope restoration.

I. INTRODUCTION

IMAGE restoration [1], [2] refers to the genre of techniques that aim to recover a high-quality original image from a degraded version of that image given a specific model for the degradation process. This is in contrast to image enhancement techniques that seek to improve the appearance of an image without reference to a specific model for the degradation process. The restoration framework is particularly valuable because in conjunction with a training-based approach, it provides a context within which the free parameters of the restoration algorithm may be optimized.

Training-based approaches have been used to develop imaging algorithms for a variety of applications, including image interpolation [3]–[6], image restoration [7], digital halftoning [8]–[12], descreening [13], and color correction [14]. The ingredients that training-based approaches have in common when used for development of imaging algorithms

are: 1) a set of training pairs each consisting of an input image and a desired output image, 2) an architecture for the algorithm consisting of free parameters, and 3) a cost function under which those free parameters may be optimized. In many cases, the architecture contains a classifier that allows for parameter optimization separately within different pixel classes according to the value of an appropriately chosen feature vector.

In this paper, we propose a new training-based approach to image restoration. Once the restoration algorithm has been fully developed, we are, however, free to apply it to images for which the degradation process is unknown. This puts us back in the domain of enhancement. The success of this broader application of the restoration algorithm will depend on how general is the degradation model under which the algorithm was developed, as well as how robust is the overall structure of the algorithm to deviations from the assumed degradation model. The scope of this paper is to deal with images that are appropriate for digital photography. We do not consider images that are severely degraded.

The two most common forms of degradation an image suffers are loss of sharpness or blur, and noise. The degradation model we use consists of a linear, shift-invariant blur followed by additive noise, described in detail in [7]. The problem we are interested in is twofold. First we seek to develop a sharpening method that is fundamentally different from the unsharp mask filter (USM) [15], which sharpens an image by enhancing the high-frequency components of the image. In the spatial domain, the boosted high-frequency components lead to overshoot and undershoot around edges, which causes objectionable ringing or halo artifacts. Our goal is to develop a sharpening algorithm that increases the slope of edges without producing overshoot and undershoot, which renders clean, crisp, and artifact-free edges, thereby improving the overall appearance of the image. The second aspect of the problem we wish to address is noise removal. We want to present a unified solution to both sharpness enhancement and noise removal. In most applications, the degraded image contains both noise and blur. A sharpening algorithm that works well only for noise-free images will not be applicable in these situations.

In terms of noise removal, conventional linear filters work well for removing additive Gaussian noise, but they also significantly blur the edge structures of an image. Therefore, a great deal of research has been done on edge-preserving noise reduction. One of the major endeavors in this area has been to utilize rank order information [16]–[19]. Due to a lack of the sense of spatial ordering, rank order filters generally do not retain the frequency selective properties of the linear filters and do not suppress Gaussian noise optimally [20]. Hybrid schemes combining both rank order filtering and linear filtering have been proposed in order to take advantage of both approaches [20],

Manuscript received February 16, 2007; revised November 16, 2007. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stanley J. Reeves.

B. Zhang is with the Texas Instruments, Inc., TX 75243 USA (e-mail: buyue@ti.com).

J. P. Allebach is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: allebach@ecn.purdue.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TIP.2008.919949

[21]. These nonlinear rank order approaches in general improve the edge sharpness, but they are more complex to implement than a spatial linear filter.

In more recent years, a new concept in edge-preserving de-noising was proposed by Smith and Brady [22] and Tomasi and Manduchi [23]. Although their algorithms were developed independently, and named the “SUSAN” filter and the “bilateral filter,” respectively, the essential idea is the same: enforcing both geometric closeness in the spatial domain and gray value similarity in the range in the de-noising operation. The idea of bilateral filtering has since found its way into many applications not only in the area of image de-noising, but also computer graphics [24], [25], video processing [26], [27], image interpolation [28], illumination estimation [29], as well as relighting and texture manipulation, dynamic range compression, and several others pointed out in [30]. Several researchers have provided a theoretical analysis of the bilateral filter and connected it with the classical approaches to noise removal. Elad [31] demonstrated that the bilateral filter emerges from the well-known Bayesian approach [32], [33], when a novel penalty function is used. Based on this observation, he proposed methods to speed up the bilateral filtering and to implement a bilateral filter for piece-wise linear signals. In [29], Elad also pointed out that the bilateral filter is a discrete version of the short-time effective kernel of the Beltrami flow discussed in [34] and [35]. Barash and Comaniciu [36], [37] demonstrated that the nature of the bilateral filter resembles that of anisotropic diffusion [38], [39], and outlined a common framework for bilateral filtering, nonlinear diffusion, adaptive smoothing [40], and a mean shift procedure [41]. A good review on bilateral filter, its properties, and applications can be found in [30].

The bilateral filter is the framework for our proposed algorithm. We will discuss it in more detail in Section II. Here, we would like to point out that the bilateral filter is essentially a smoothing filter, it does not restore the sharpness of a degraded image. Aleksic *et al.* modified the bilateral filter to perform both noise removal and sharpening by adding a high-pass filter to the conventional bilateral filter [42]. This filter essentially performs USM sharpening for pixels that are above a preselected high-pass threshold. Therefore, it produces halo artifacts as does an USM filter.

In contrast to the extensive effort to improve de-noising algorithms, much less has been done for sharpening algorithms. The USM remains the prevalent sharpening tool despite the drawbacks that it has. First, the USM sharpens an image by adding overshoot and undershoot to the edges which produces halo artifacts. Second, when applied to a noisy image, the USM will amplify the noise in smooth regions which significantly impairs the image quality. To address the first problem, slope restoration algorithms have been proposed. Das and Rangayyan developed an edge sharpness enhancement algorithm to improve the slope of edges [43]. In their algorithm, the edge normal direction is first detected and then a 1-D operator is applied to the edge pixels so that the transitions of edges are made steeper. They pointed out that corner artifacts and inferior enhancement for circular regions were limitations of their algorithm as defined in [43]. The testing of their algorithm in [43] was limited

to bi-level, synthesized images. Tegenbosch *et al.* proposed a luminance transient improvement (LTI) algorithm which first sharpens the image with a linear sharpening method such as the USM, then detects the 1-D edge profiles, and finally clips between the start and end level of the edges to get rid of the overshoot and undershoot [44]. The 1-D LTI algorithm is implemented in a 2-D image by three alternative methods. 1) Apply LTI in the direction of the edge normal, which involves estimating the local edge orientation. 2) Apply LTI in the horizontal direction first, then in the vertical direction. 3) Apply LTI in parallel, in the horizontal and vertical directions, then combine the results by a weighted sum depending on the edge orientation. According to the authors, 1) produces the best image quality, but is computationally expensive; 2) is efficient to implement, but has the drawbacks of inhomogeneous enhancement and staircase artifacts; and 3) achieves the best balance between image quality and computational complexity. Results comparing images enhanced by the three proposed methods are given. However, no comparison of the input and enhanced output images is provided, nor are edge profiles shown to demonstrate the effectiveness of the proposed algorithms in restoring edge slopes. We will show in Section V that our method is able to improve image sharpness for both natural images and text images, and the edge slopes are significantly increased.

To address the second problem of the USM filter, locally adaptive sharpening and smoothing algorithms have been proposed. Kotera *et al.* developed an adaptive sharpening algorithm based on the USM and the classification of the pixels [45], [46]. The histogram of the edge strength is used to classify pixels into smooth regions, soft edges, and hard edges, which are subsequently processed with different sharpening strengths. Kim and Allebach developed an adaptive sharpening and de-noising algorithm—the optimal unsharp mask (OUM) [7]. Instead of using a fixed gain λ for the high-pass filter as in the case of the conventional USM filter, the OUM employs a locally adaptive λ , which has been trained for different regions of the image with pairs of high quality original and corresponding degraded images. By allowing the gain λ to be negative, the OUM is capable of both sharpening and de-noising. The degraded images were generated by applying a blur point spread function (PSF) to the original image, and adding noise to the blurred image. The blur PSF was modeled after a hybrid analog and digital imaging system, which involves scanning a silver-halide photograph printed from a negative exposed with a low-cost analog camera. A tone-dependent noise model was used to simulate noise produced in the imaging pipeline. The parameters of our proposed restoration algorithm are optimized in a training-based framework similar to that of the OUM. Another adaptive sharpening and smoothing filter is proposed by Guillon *et al.*, which consists of a lowpass filter and a highpass filter [47]. The high-pass filter is scaled by a factor adaptive to the sharpness of local edges. The coefficients of the filter masks are recursively updated.

The rest of this paper is organized as follows. In Section II, we describe the bilateral filter and how it works. In Section III, we present our proposed adaptive bilateral filter (ABF). In Section IV, we describe the optimization of the ABF parameters. In Section V, we evaluate the performance of the ABF and

compare it with that of the OUM and the bilateral filter. Finally, we provide conclusions in Section VI.

II. BILATERAL FILTER AND ITS PROPERTIES

The bilateral filter proposed by Tomasi and Manduchi in 1998 is a nonlinear filter that smoothes the noise while preserving edge structures [23]. The shift-variant filtering operation of the bilateral filter is given by

$$\hat{f}[m, n] = \sum_k \sum_l h[m, n; k, l] g[k, l] \quad (1)$$

where $\hat{f}[m, n]$ is the restored image, $h[m, n; k, l]$ is the response at $[m, n]$ to an impulse at $[k, l]$, and $g[m, n]$ is the degraded image. For the bilateral filter, see (2), shown at the bottom of the page, where $[m_0, n_0]$ is the center pixel of the window, $\Omega_{m_0, n_0} = \{[m, n] : [m, n] \in [m_0 - N, m_0 + N] \times [n_0 - N, n_0 + N]\}$, σ_d and σ_r are the standard deviations of the domain and range Gaussian filters, respectively, and

$$r_{m_0, n_0} = \sum_{m=m_0-N}^{m_0+N} \sum_{n=n_0-N}^{n_0+N} \exp\left(-\frac{(m-m_0)^2 + (n-n_0)^2}{2\sigma_d^2}\right) \times \exp\left(-\frac{(g[m, n] - g[m_0, n_0])^2}{2\sigma_r^2}\right) \quad (3)$$

is a normalization factor that assures that the filter preserves average gray value in constant areas of the image.

The edge-preserving de-noising bilateral filter adopts a low-pass Gaussian filter for both the domain filter and the range filter. The domain low-pass Gaussian filter gives higher weight to pixels that are spatially close to the center pixel. The range low-pass Gaussian filter gives higher weight to pixels that are similar to the center pixel in gray value. Combining the range filter and the domain filter, a bilateral filter at an edge pixel becomes an elongated Gaussian filter that is oriented along the edge. This ensures that averaging is done mostly along the edge and is greatly reduced in the gradient direction. This is the reason why the bilateral filter can smooth the noise while preserving edge structures.

Fig. 1(b) shows that a bilateral filter with $\sigma_d = 2$ and $\sigma_r = 20$ removes much of the noise that appears in the degraded image shown in Fig. 1(a) and preserves the edge structures. In Fig. 1(c), where the spatial domain Gaussian with $\sigma_d = 2$ is applied alone, the edges are significantly blurred. The range filter with $\sigma_r =$

20 at the edge pixel A is shown in Fig. 1(e). Combining the spatial domain Gaussian filter [Fig. 1(d)] and the range Gaussian filter [Fig. 1(e)] results in the bilateral filter at pixel A shown in Fig. 1(f). The transfer function of the bilateral filter shown in Fig. 1(g) demonstrates that the bilateral filter at pixel A is low pass in one direction, and almost all-pass in the orthogonal direction. This explains, from a frequency domain perspective, why this filter is able to preserve edges while removing noise. On the other hand, the bilateral filter is essentially a smoothing filter. It does not sharpen edges. As shown in Fig. 1(h) and (i), the edge rendered by the bilateral filter has the same level of blurriness as in the original degraded image, although the noise is greatly reduced.

The results of the bilateral filtering are a significant improvement over a conventional linear low-pass filter. However, in order to enhance the sharpness of an image, we need to make some modifications to this filter.

III. ADAPTIVE BILATERAL FILTER (ABF) FOR IMAGE SHARPENING AND DE-NOISING

In this section, we present a new sharpening and smoothing algorithm: the adaptive bilateral filter (ABF). The response at $[m_0, n_0]$ of the proposed shift-variant ABF to an impulse at $[m, n]$ is given by (4), shown at the bottom of the page, where $[m_0, n_0]$ and Ω_{m_0, n_0} are defined as before, and the normalization factor is given by

$$r_{m_0, n_0} = \sum_{m=m_0-N}^{m_0+N} \sum_{n=n_0-N}^{n_0+N} \exp\left(-\frac{(m-m_0)^2 + (n-n_0)^2}{2\sigma_d^2}\right) \times \exp\left(-\frac{(g[m, n] - g[m_0, n_0] - \zeta[m_0, n_0])^2}{2\sigma_r^2}\right). \quad (5)$$

The ABF retains the general form of a bilateral filter, but contains two important modifications. First, an offset ζ is introduced to the range filter in the ABF. Second, both ζ and the width of the range filter σ_r in the ABF are locally adaptive. If $\zeta = 0$ and σ_r is fixed, the ABF will degenerate into a conventional bilateral filter. For the domain filter, a fixed low-pass Gaussian filter with $\sigma_d = 1.0$ is adopted in the ABF. The combination of a locally adaptive ζ and σ_r transforms the bilateral filter into a much more powerful filter that is capable of both smoothing and sharpening. Moreover, it sharpens an image by increasing the slope of the edges. To understand how the ABF works, we need to understand the role of ζ and σ_r in the ABF.

$$h[m_0, n_0; m, n] = \begin{cases} r_{m_0, n_0}^{-1} \exp\left(-\frac{(m-m_0)^2 + (n-n_0)^2}{2\sigma_d^2}\right) \exp\left(-\frac{(g[m, n] - g[m_0, n_0] - \zeta[m_0, n_0])^2}{2\sigma_r^2}\right), & [m, n] \in \Omega_{m_0, n_0} \\ 0, & \text{else} \end{cases} \quad (2)$$

$$h[m_0, n_0; m, n] = \begin{cases} r_{m_0, n_0}^{-1} \exp\left(-\frac{(m-m_0)^2 + (n-n_0)^2}{2\sigma_d^2}\right) \exp\left(-\frac{(g[m, n] - g[m_0, n_0] - \zeta[m_0, n_0])^2}{2\sigma_r^2}\right), & [m, n] \in \Omega_{m_0, n_0} \\ 0, & \text{else} \end{cases} \quad (4)$$

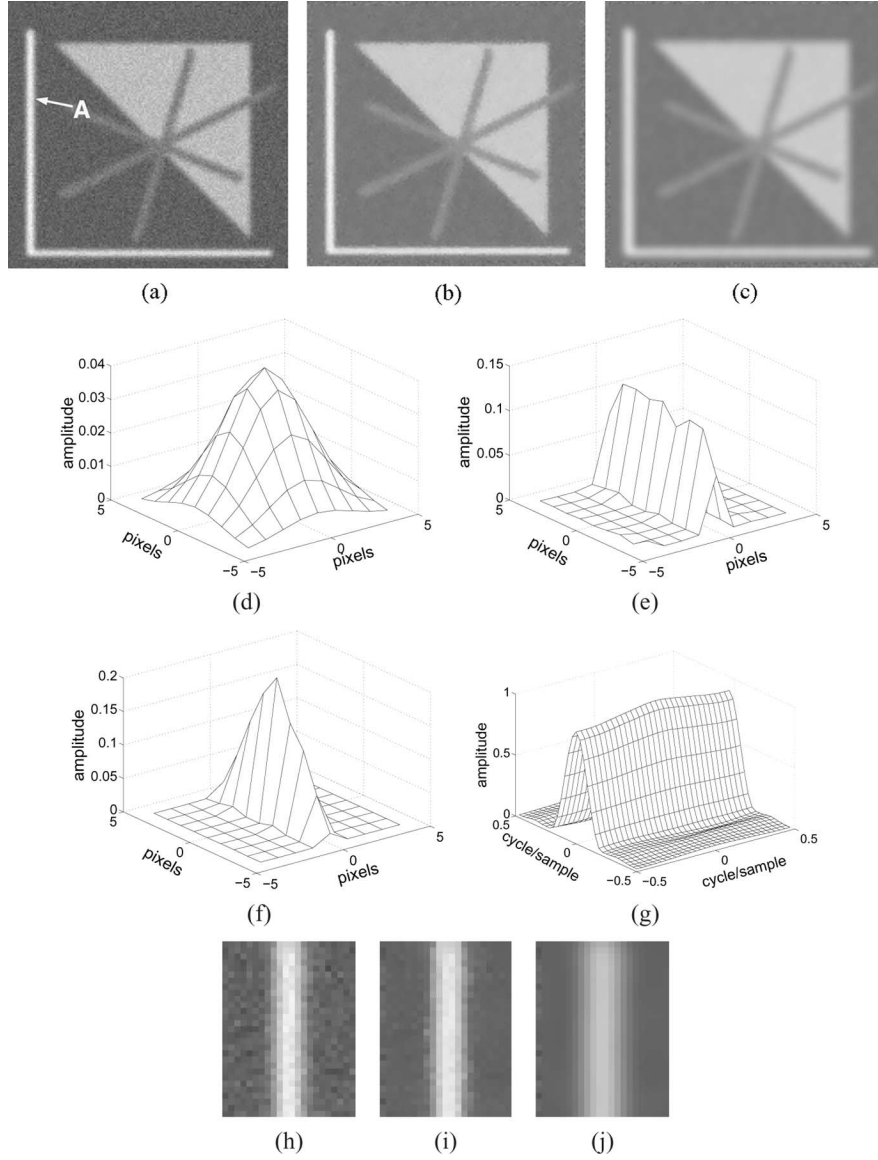


Fig. 1. Illustration of how the bilateral filter works. (a) Blurred and noisy 165×156 image “edge.tif”; (b) bilateral filter restored image ($\sigma_r = 20$; $\sigma_d = 2$); (c) Gaussian filter restored image ($\sigma_d = 2$); (d) the impulse response of the domain Gaussian filter ($\sigma_d = 2$); (e) the impulse response of the range Gaussian filter at the edge pixel A at the tip of the red arrow in (a) ($\sigma_r = 20$); (f) the impulse response of the bilateral filter at pixel A; (g) the transfer function of (f) at pixel A. Zoomed in images of a portion of the edge at pixel A in (a)–(c) are shown in (h)–(j), respectively.

A. Role of ζ in the ABF

The range filter can be interpreted as a 1-D filter that processes the histogram of the image. We will illustrate this viewpoint for the window of data enclosed in the red box in the boy portrait images in Table I. We index the images in the table by their [row, column] coordinates. The original degraded image with the red data box is shown in [1, 2], for which the histogram is shown in [1, 3]. For the conventional bilateral filter, the range filter is located on the histogram at the gray value of the current pixel and rolls off as the pixel values fall farther away from the center pixel value as shown in [2, 1]. By adding an offset ζ to the range filter, we are now able to shift the range filter on the histogram, as shown in [3, 1], [4, 1], and [5, 1]. As before, let Ω_{m_0, n_0} denote the set of pixels in the $(2N + 1) \times (2N + 1)$ window of pixels centered at $[m_0, n_0]$. Let MIN, MAX, and MEAN denote the operations of taking the

minimum, maximum, and average value of the data in Ω_{m_0, n_0} , respectively. Let $\Delta_{m_0, n_0} = g[m_0, n_0] - \text{MEAN}(\Omega_{m_0, n_0})$. We will demonstrate the effect of bilateral filtering with a fixed domain Gaussian filter ($\sigma_d = 1.0$) and a range filter ($\sigma_r = 20$) shifted by the following choices for ζ .

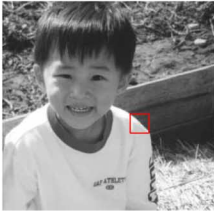
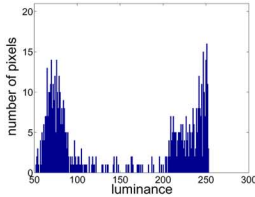
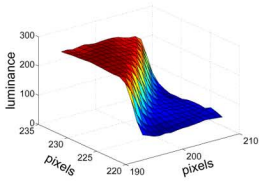
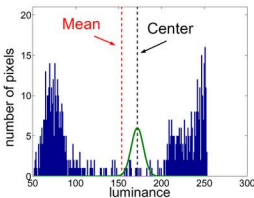

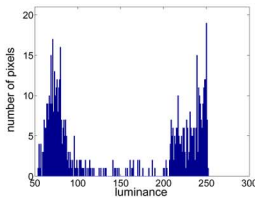
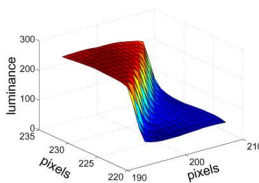
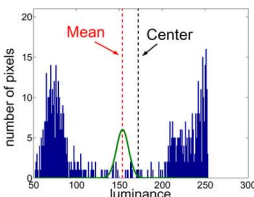
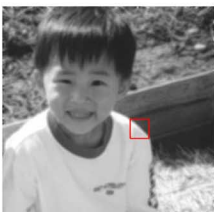
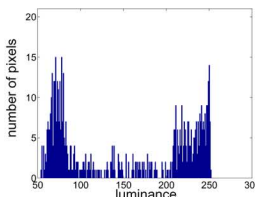
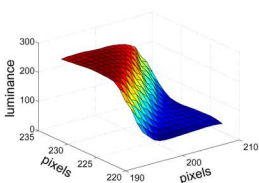
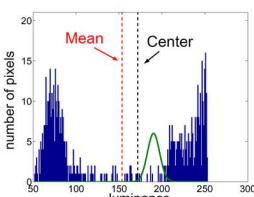
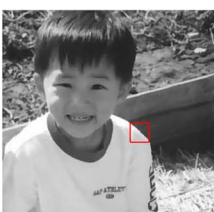
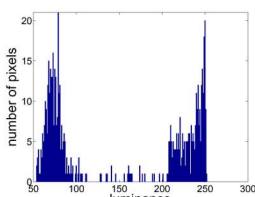
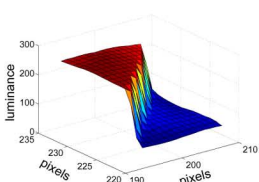
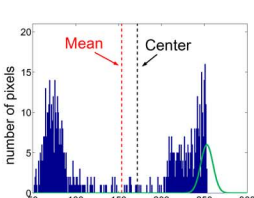
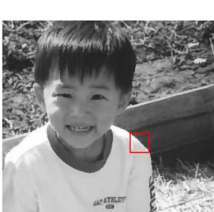
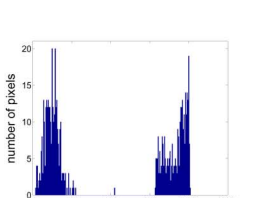
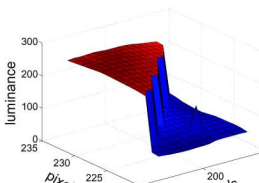
- 1) No offset (conventional bilateral filter): $\zeta[m_0, n_0] = 0$.
- 2) Shifting towards the MEAN: $\zeta[m_0, n_0] = -\Delta_{m_0, n_0}$.
- 3) Shifting away from the MEAN: $\zeta[m_0, n_0] = \Delta_{m_0, n_0}$.
- 4) Shifting away from the MEAN, to the MIN/MAX

$$\zeta[m_0, n_0] = \begin{cases} \text{MAX}(\Omega_{m_0, n_0}) - g[m_0, n_0], & \text{if } \Delta_{m_0, n_0} > 0 \\ \text{MIN}(\Omega_{m_0, n_0}) - g[m_0, n_0], & \text{if } \Delta_{m_0, n_0} < 0 \\ 0, & \text{if } \Delta_{m_0, n_0} = 0. \end{cases} \quad (6)$$

The locations of the resultant range filters with regard to the histogram of the data in Ω_{m_0, n_0} are illustrated in Table I, rows two to five. Here, $N = 12$. As we can see from Table I, shifting

TABLE I

ILLUSTRATION OF THE EFFECT OF BILATERAL FILTERING WITH A FIXED DOMAIN GAUSSIAN FILTER ($\sigma_d = 1.0$) AND A RANGE FILTER ($\sigma_r = 20$) WITH THE FOUR CHOICES OF ζ DESCRIBED BY OPERATIONS NO. 1 (SECOND ROW), NO. 2 (THIRD ROW), NO. 3 (FOURTH ROW), AND NO. 4 (FIFTH ROW). THE DEGRADED IMAGE “BOY” AND THE RESULTANT IMAGES ARE SHOWN IN COLUMN 2, THE RED BOX MARKED ON THE “BOY” DENOTES A 25×25 DATA WINDOW Ω_{m_0, n_0} OF INTEREST

	1. Histograms of the image data in Ω_{m_0, n_0} before filtering, and the position of the range filters at $[m_0, n_0]$ with regard to the local histograms	2. The resultant images	3. The resultant histograms of Ω_{m_0, n_0}	4. The 3D plots of the filtered data in Ω_{m_0, n_0}
1. No Action				
2. Operation No. 1				
3. Operation No. 2				
4. Operation No. 3				
5. Operation No. 4				

the range filter towards $\text{MEAN}(\Omega_{m_0, n_0})$ will blur the image [3, 2]. Shifting the range filter away from $\text{MEAN}(\Omega_{m_0, n_0})$ will sharpen the image [4, 2]). In the extreme case, if for every pixel above $\text{MEAN}(\Omega_{m_0, n_0})$, we shift the range filter to $\text{MAX}(\Omega_{m_0, n_0})$, and for every pixel below $\text{MEAN}(\Omega_{m_0, n_0})$,

we shift the range filter to $\text{MIN}(\Omega_{m_0, n_0})$, we will see a drastic sharpening effect and the image will appear over-sharpened ([5, 2]). The reason behind these observations is the transformation of the histogram of the input image by the range filter. In our case, the data window Ω_{m_0, n_0} marked by the red

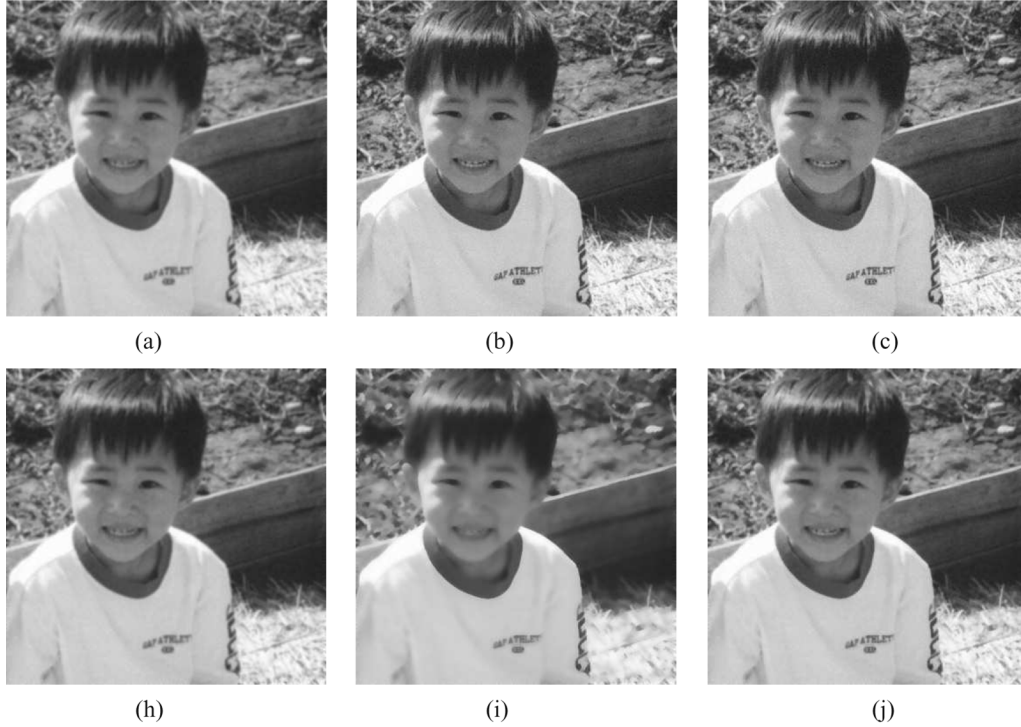


Fig. 2. Impact of σ_r . A fixed $\sigma_d = 1$ is used in the bilateral filter. DF: domain filter only; RF: range filter only; BF: bilateral filter. (a) DF: $\sigma_d = 1$; (b) RF: $\sigma_r = 5$; (c) BF: $\sigma_d = 1, \sigma_r = 5$; (h) DF: $\sigma_d = 1$; (i) RF: $\sigma_r = 50$; (j) BF: $\sigma_d = 1, \sigma_r = 50$.

box in [1, 2] contains an edge. Therefore, the histogram of the data in Ω_{m_0, n_0} has two peaks, which correspond to the darker and brighter sides of the edge, respectively ([1, 3]). Any pixels located between the two peaks appear on the slope of the edge. A 3-D plot of the image data in Ω_{m_0, n_0} is shown in [1, 4]. The conventional bilateral filter (no shift to the range filter) does not significantly alter the histogram of the data ([2, 3]) and, consequently, does not change the slope of the edge ([2, 4]).

Shifting the range filter to $\text{MEAN}(\Omega_{m_0, n_0})$ at each pixel will redistribute the pixels towards the center of the histogram ([3, 3]). Hence, the slope is reduced ([3, 4]). On the other hand, if we shift the range filter further away from $\text{MEAN}(\Omega_{m_0, n_0})$, pixels will be compressed against the two peaks ([4, 3]). The slope will then be increased ([4, 4]). In the case of operation No. 4, the histogram is further compressed around the two peaks, as shown in [5, 3]. The edge almost becomes a step function, as shown in [5, 4]. In this case, one can also observe an outlier pixel in [5, 4]. This is because the range filter can be very sensitive to noise if not applied correctly. In particular, shifting the range filter based on Δ_{m_0, n_0} is not robust to noise. If the range filter is shifted in the wrong direction, large errors will result. In Section IV, we describe a different strategy for choosing ζ that is much more reliable. The domain filter also makes the bilateral filter more robust to noise.

B. Role of σ_r in the ABF

The parameter σ_r of the range filter controls the width of the range filter. It determines how selective the range filter is in choosing the pixels that are similar enough in gray value to be included in the averaging operation. If σ_r is large compared to the range of the data in the window, the range filter will assign similar weight to every pixel in the range. Then, it will not have

much effect on the overall bilateral filter. On the other hand, a small σ_r will make the range filter dominate the bilateral filter. Fig. 2 demonstrates this effect. The bilateral filtered image resembles the range filtered image when $\sigma_r = 5$, and it resembles the domain filtered image when $\sigma_r = 50$.

C. Summary of the Rationale for the ABF

The pixel dependent offset ζ in the ABF is the key to slope restoration. With ζ , we are able to restore the slope by transforming the local histogram of the image, thus circumventing the cumbersome process of locating edge normals and detecting edge profiles. Since at any pixel $[m_0, n_0]$ in the image, the ABF output is bounded between $\text{MIN}(\Omega_{m_0, n_0})$ and $\text{MAX}(\Omega_{m_0, n_0})$, the ABF, in general, does not produce overshoot and undershoot.

By making ζ and σ_r adaptive and jointly optimizing both parameters, we transform the bilateral filter into a much more powerful and versatile filter. To smooth the image at a given pixel, we can shift the range filter towards $\text{MEAN}(\Omega_{m_0, n_0})$, and/or use a large σ_r which enables the spatial Gaussian filter to take charge of the bilateral filtering. To sharpen the image at a given pixel, we can shift the range filter away from the midpoint of the edge slope which will be approximately equal to $\text{MEAN}(\Omega_{m_0, n_0})$, towards $\text{MAX}(\Omega_{m_0, n_0})$ or $\text{MIN}(\Omega_{m_0, n_0})$, depending on the position of the edge pixel on the edge slope. At the same time, we would reduce σ_r accordingly. With a small σ_r , the range filter dominates the bilateral filter and effectively pulls up or pushes down the pixels on the edge slope.

IV. OPTIMIZATION OF THE ABF PARAMETERS

The parameter optimization is formulated as a minimum mean squared error (MMSE) estimation problem. We classify the pixels into T classes, and during the training process

estimate the optimal ζ and σ_r for each class that minimizes the overall MSE between the original and restored images. Let P be the total number of training image sets. The k th set ($k = 1, 2, \dots, P$) consists of an original image $f_k[m, n]$, a degraded image $g_k[m, n]$, the class index image $L_k[m, n]$, and the restored image $\hat{f}_k[m, n]$. All four of these images have dimensions $M_k \times N_k$. Let

$$\mathcal{S}^{(k)} = \{[m, n] : [m, n] \in [0, M_k - 1] \times [0, N_k - 1]\} \quad (7)$$

be the set of indices for the pixels in these images. Also let

$$\mathcal{S}_i^{(k)} = \{[m, n] : L_k[m, n] = i \text{ and } [m, n] \in \mathcal{S}^{(k)}\} \\ i = 1, 2, \dots, T \quad (8)$$

be the set of indices for the pixels belonging to the class i in image k .

Given the P training image sets as described above, the optimal parameters ζ^* and σ_r^* satisfy

$$\{\zeta^*, \sigma_r^*\} = \arg \min_{\{\zeta, \sigma_r\}} \sum_{k=1}^P \|\hat{f}_k[m, n] - f_k[m, n]\|_{\mathcal{S}^{(k)}}^2 \quad (9)$$

where $\|A\|_B^2$ denotes the L-2 norm of the array A over the index set B , $\zeta = \{\zeta_i : i = 1, 2, \dots, T\}$, and $\sigma_r = \{\sigma_{r,i} : i = 1, 2, \dots, T\}$. Since the classes are independent and nonoverlapping, we can separately estimate the optimal ζ^* and σ_r^* for each class

$$\{\zeta_i^*, \sigma_{r,i}^*\} = \arg \min_{\{\zeta_i, \sigma_{r,i}\}} \sum_{k=1}^P \|\hat{f}_k[m, n] - f_k[m, n]\|_{\mathcal{S}_i^{(k)}}^2 \\ i = 1, \dots, T. \quad (10)$$

The restored image $\hat{f}[m, n]$ is a nonlinear function of the parameters ζ_i and $\sigma_{r,i}$. There is no closed-form solution for ζ_i^* and $\sigma_{r,i}^*$. To find the pair of parameters which minimizes the MSE for each class, we perform an exhaustive search in the parameter space $\Phi = \{(\zeta, \sigma_r) : \zeta \in \phi_\zeta \text{ and } \sigma_r \in \phi_{\sigma_r}\}$, where $\phi_\zeta = [-60, 60]$ and $\phi_{\sigma_r} = [5, 45]$. The parameter space is uniformly quantized with step sizes $\Delta_\zeta = 1$ and $\Delta_{\sigma_r} = 2$. The range and the step size of the parameters are chosen empirically such that they can yield adequate sharpening and smoothing for all types of image structures with a balance between accuracy and computational cost. Next, we will discuss the feature design, the training images, and the training results.

A. Feature Design

The feature for pixel classification plays an important role in the success of the training. In Section III-A, we described how shifting the range filter according to Δ , the difference between the center pixel value and the mean of the local data window,

will impact the output image. Despite the high correlation between Δ and the desired sharpening/smoothing effect, Δ is not an appropriate feature for pixel classification because it is very sensitive to noise. Our general guidelines for choosing the feature(s) are: 1) be able to reflect the strength of edges, 2) can distinguish the regions we want to process differently, mainly, the regions for smoothing and sharpening, and 3) have some robustness to noise.

The feature we have chosen to use for pixel classification is the strength of the edges measured by a Laplacian of Gaussian (LoG) operator with a 9×9 kernel and $\sigma_{\text{LoG}} = 1.5$ [see (11), shown at the bottom of the page], where $N = 4$ and

$$C = \frac{1}{(2N+1)^2} \sum_{m=m_0-N}^{m_0+N} \sum_{n=n_0-N}^{n_0+N} -\frac{1}{\pi\sigma_{\text{LoG}}^4} \\ \times \left(1 - \frac{m^2+n^2}{2\sigma_{\text{LoG}}^2}\right) \exp\left(-\frac{m^2+n^2}{2\sigma_{\text{LoG}}^2}\right). \quad (12)$$

The impulse response and frequency response of the LoG operator are shown in Fig. 3(a) and (b), respectively. The LoG operator is a highpass filter. It computes the second derivative of the input image. Therefore, near edges, the magnitude of its response is high; in smooth regions, the magnitude of its response is low; and on the center of an edge, the magnitude of its response is 0. Fig. 4 illustrates these properties with the LoG response to a ramp edge. The LoG strength is also used in the OUM for pixel classification. However, the OUM does not use the sign of the LoG output for classification. For us, this information is very important, as discussed below.

We choose the LoG output as our pixel classification feature for three major reasons. First, the magnitude of the LoG strength reflects the local edge structure. It can distinguish smooth regions from edge regions, where the optimal filter parameters are most likely to be very different. One special case is the edge centers. The magnitude of the LoG response cannot distinguish the center of the edges and the noisy smooth regions very well because both have small LoG response. However, the ABF response at pixel $[m_0, n_0]$ depends on both the spatial domain and range space distribution of the data in the window Ω_{m_0, n_0} . Although, these two region types are not completely separated by the LoG classifier, the very different data structures at the edge centers and in the smooth regions ensure that the ABF with the same ζ and σ_r can satisfy the filtering needs of both types of regions. To illustrate this idea, we consider the three sample pixels shown in Fig. 5(a). The pixels A and B are located in a noisy smooth region and on an edge center, respectively. Even though the optimal filter parameters are the same for both pixels, the impulse responses of the ABF are different for pixels A and B. Pixel C shows how ζ^* and σ_r^* impact the ABF. Without the offset ζ and the locally adaptive σ_r , the ABF at pixel C would

$$\text{LoG}[m, n] = \begin{cases} -\frac{1}{\pi\sigma_{\text{LoG}}^4} \left(1 - \frac{m^2+n^2}{2\sigma_{\text{LoG}}^2}\right) \exp\left(-\frac{m^2+n^2}{2\sigma_{\text{LoG}}^2}\right) - C, & |m|, |n| \leq N \\ 0, & \text{else} \end{cases} \quad (11)$$

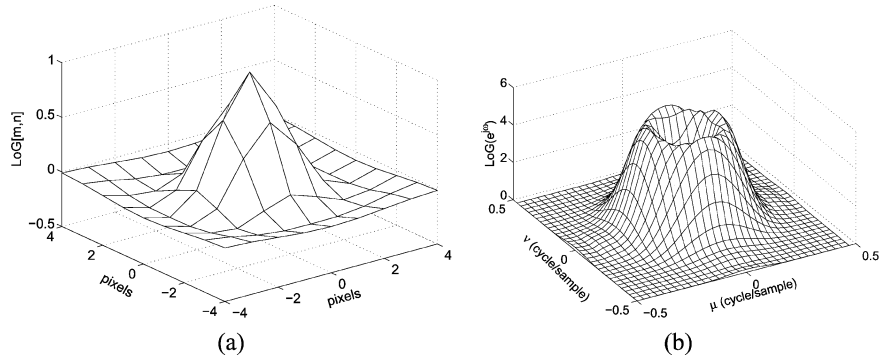


Fig. 3. LoG response. (a) Impulse response; (b) Frequency response.

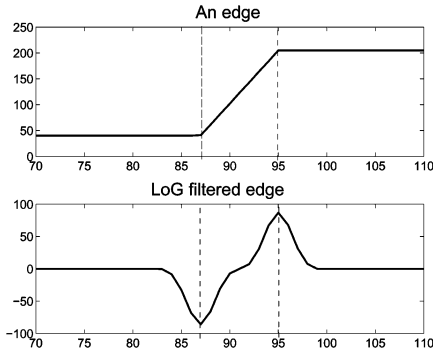


Fig. 4. Ramp edge and its LoG response.

be the same as that at pixel B. In that case, no sharpening effect would be achieved at pixel C.

Second, another advantage of the LoG operator is that it has a Gaussian kernel, which makes it more robust to noise compared to other derivative operators. Since we are restoring blurred and noisy images, this is a very desirable property.

Finally, the sign of the LoG output tells the relative location of the pixel with respect to a nearby edge center. In particular, edge pixels located above the midpoint of the edge slope have a positive LoG response; those below the midpoint of the edge slope have a negative LoG response. This is important because when the edge pixel is below the midpoint of the edge slope, we want to move it further down; hence, ζ should be negative. When the edge pixel is above the midpoint of the edge slope, we want to shift it further up. Therefore, ζ should be positive.

The pixel class index image is computed according to

$$L[m, n] = \begin{cases} [g_{\text{LoG}}[m, n]], & |g_{\text{LoG}}[m, n]| \leq L_{\max} \\ L_{\max}, & g_{\text{LoG}}[m, n] > L_{\max} \\ -L_{\max}, & g_{\text{LoG}}[m, n] < -L_{\max} \end{cases} \quad (13)$$

$[m, n] \in \mathcal{S}^{(k)}$

where $g_{\text{LoG}}[m, n] = \text{LoG}[m, n] * g[m, n]$, $L_{\max} = 60$, and $[x]$ denotes rounding x to the nearest integer.

The choice of the σ_{LoG} of the LoG classifier defines the level at which the edges are detected. Moreover, the choice of the σ_{LoG} and L_{\max} determines the spatial distance between the minimal and maximal classifier output. Once σ_{LoG} is determined, the kernel size of the LoG is chosen to be greater than or equal to $6\sigma_{\text{LoG}}$ to ensure that the window contains most of the LoG response. We chose a 9×9 LoG kernel with $\sigma_{\text{LoG}} =$

1.5 by looking at LoG filtered images from different σ_{LoG} and kernel sizes, and selecting the one that generated reasonably strong responses to both strong edges and textures/details in the image. The quantization threshold of the LoG classifier, L_{\max} in (13), is chosen to be 60 so that the maximal negative and positive responses are separated by 1 to 2 pixels in most cases. Although the choice of σ_{LoG} and the threshold are not unique, as long as the above two requirements are satisfied, we expect the training results to be consistent.

Here, we would like to explain the choice of the standard deviation of the spatial Gaussian filter σ_d and the kernel size of the ABF as well. The choice of σ_d is not critical. As long as the domain Gaussian filter with σ_d can perform reasonable smoothing for the amount of noise introduced by the degradation model on its own, it is sufficient for the ABF. We pointed out in Section VI that jointly optimizing σ_d , σ_r , and ζ would be of interest for future work. The choice of the kernel size of the ABF is based on the following two main considerations. 1) Since the ABF contains a spatial Gaussian filter with $\sigma_d = 1.0$, the kernel size of the ABF should be greater than or equal to $6\sigma_d$. 2) The kernel size of the ABF should also match that of the LoG classifier so that it contains the essential information the feature is trying to capture and represent. If the kernel size of the ABF is too small, it will not be able to cover most of the edge transitions, the ABF will then have limited power to pull up/push down the pixels along the edge slope. On the other hand, if the kernel is too large, it might contain several edges which would confuse the ABF as to which pixels belong to which edge. To minimize the computation, we choose the smallest kernel size that satisfies all these considerations, which is 7×7 in our case.

B. Training Images

We used five high quality, high resolution images for our training set. They are from the six training images used for the OUM [7]. One image was left out of the training set to serve as a test image. The images and their sizes are shown in [7] as Fig. 8(a)–(d) and (f). The content of the training images cover a variety of scenes. To generate the degraded images, we applied a blur PSF to the original image, and added tone-dependent noise to the blurred image.

As stated in the introduction, the scope of this paper is to deal with images that are appropriate for digital photography. These are the images with a reasonable level of quality, from one's digital camera or scanners that one would actually want to print. There is a large market for enhancing the digital photographs

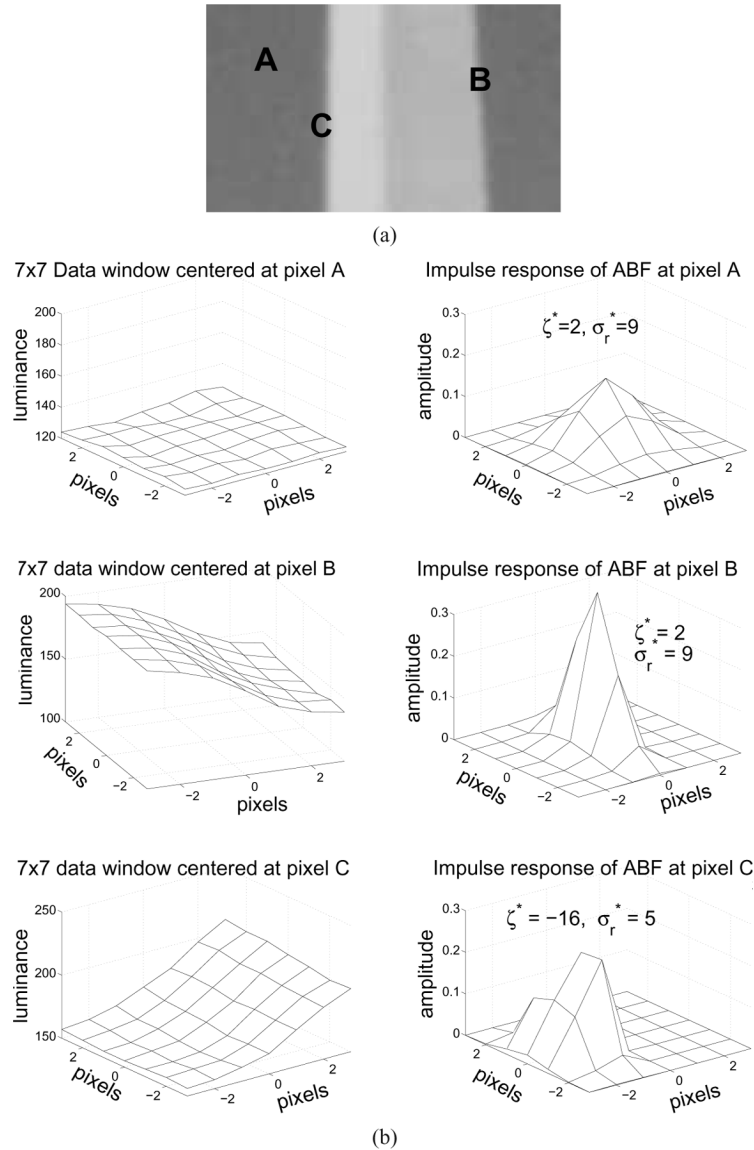


Fig. 5. ABF for various input image structures. (a) Sample image with three selected pixels: “A” is in the noisy smooth region, “B” is located at the center of an edge, and “C” is located on the slope of an edge, but not at the edge center; (b) shows the 7×7 data window centered at pixels A, B, and C, as well as the corresponding impulse responses of the ABF at these positions. The optimal parameters for the ABF, i.e., ζ^* and σ_r^* , are marked on the plots.

from consumers, such as the self-serve photo kiosks in the supermarket, and the on-line printing services. These services all apply image enhancing algorithms to the digital photos before printing them. This is the main application we targeted when designing the ABF. Therefore, we adopted an image degradation model that simulates the blur and noise introduced by a real hybrid imaging pipeline. The blur PSF and the tone-dependent noise are modeled after an imaging pipeline which involves capturing the original image on silver halide film with a low cost point-and-shoot camera, using a common photo processor to develop the film and print a 4×6 in² silver halide print, and finally scanning the print to obtain a digital image.

The size of the blur PSF of this imaging pipeline is approximately 4 pixels. Here, the size of the PSF is defined as twice the distance in pixels from the origin to the point at which the PSF has dropped to 50% of its value at the origin. Moreover, this degradation model is the same one used for the training of the

OUM. Since we are comparing the performance of ABF with that of the OUM, it is important for us to adopt similar training images. Otherwise, we will not be able to factor out the possible impact of the training images from the difference in performance. The blur PSF and tone-dependent noise models are described in detail in [7].

C. Estimated Optimal $\vec{\zeta}^*$ and $\vec{\sigma}_r^*$

Fig. 6(a) shows that ζ^* is approximately proportional to the LoG class. This confirms the validity of using the LoG response as the feature for pixel classification. The desired amount of offset is proportional to the magnitude of the LoG response. In addition, the sign of ζ^* is the same as that of the LoG class. This means that the offset does indeed serve the function described in Section III-A, i.e., it pushes pixel values in edge regions from the edge slope center value either towards the high or low side of the edge based on whether the pixel is originally

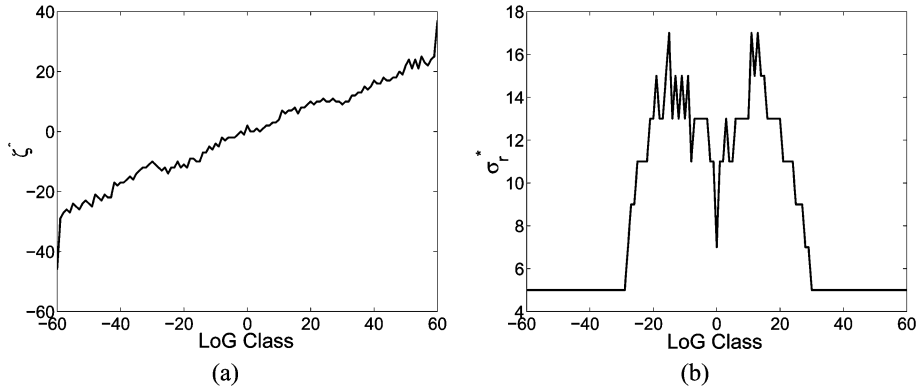


Fig. 6. Estimated optimal parameters for the ABF. (a) Optimal offset ζ^* for each class. (b) Optimal σ_r^* for each class.

above or below the midpoint of the edge slope. Fig. 6(a) and (b) shows that there are three broad classes of pixels in terms of how ζ^* and σ_r^* behave as a function of LoG class. In fact, Fig. 6(a) suggests that ζ^* is chosen entirely to yield optimal sharpening of the edges. The choice of whether ABF sharpens or smooths is controlled by σ_r^* and the local image structure, as illustrated in Figs. 2 and 5.

For the LoG class equal to zero (center of edges and flat areas), ζ^* equals zero and σ_r^* is moderate. Therefore, the edge centers are mostly kept intact, and the flat areas are given some smoothing. As LoG class moves away from zero, becoming either positive or negative, σ_r^* becomes large. Hence, the domain Gaussian filter dominates and the ABF is smoothing the pixels. This occurs in noisy regions of the image that lack details and should be smoothed. As LoG continues to increase in magnitude, we approach the upper or lower side of the edge slope, as illustrated in Fig. 4. Here, we need the offset to have maximal impact. This is assured by the small value assigned to σ_r^* for LoG class less than -29 and greater than 30.

In Section III-A, we demonstrated the rationale of the ABF with a few cases for ζ , i.e., shifting the range filter to the MAX, MIN, and MEAN of the local data window Ω_{m_0, n_0} . As the training results turn out, it is possible that the trained optimal offset ζ^* will shift the location of the range filter to exceed the MIN or MAX of Ω_{m_0, n_0} . For instance, for our test image 2 “kids,” which is discussed in Section V, 20% of the range filters are shifted to be below the MIN, 5% to the MIN, 53% to be somewhere between the MIN and the MAX, 4% to the MAX, and 18% above the MAX. It is also important to note that although the range filter can be shifted below the MIN or above the MAX, the output of the ABF will always be in the range of [MIN MAX] of Ω_{m_0, n_0} , because by definition the ABF is a weighted average of Ω_{m_0, n_0} which maintains the DC component. That is why the ABF in general does not produce overshoot and undershoot when sharpening an edge.

In Sections III-A and B, we also discussed all the ways that ζ and σ_r can control what the ABF does. Fig. 6(a) and (b) shows which of these potential modes actually get used by the optimal ζ^* and σ_r^* . The choice of the optimal parameters, to a great extent, is related to the choice of the feature. For instance, we also tried using both the LoG response and the Prewitt gradient [48] as the features for pixel classification. Adding another dimension to the feature with the Prewitt gradient made the edge centers and noisy smooth regions separable. In this case, for pixels

in the noisy smooth regions, ζ^* had a sign opposite to that of the LoG class, i.e., the de-noising of these regions is achieved by pushing the data distribution towards the MEAN. However, the quality of the images restored with this filter is almost indistinguishable from the images restored using the ABF based on only the LoG response. Therefore, we decided to use the simpler feature.

V. RESULTS AND DISCUSSIONS

The performance of the ABF is evaluated with three images. The first test image is degraded exactly according to the degradation model described in Section IV-B. The second test image is obtained from a real imaging pipeline on which the aforementioned degradation model is based. The third test image is outside the context of our degradation model. The results obtained with this image will suggest how robust the ABF is for images that are significantly different from our training images. At the end of this section, we will also discuss briefly the application, implementation, and training of the ABF.

We begin by comparing the performance of the ABF with a simple two-step approach. For this approach, we adopted a Gaussian filter with standard deviation of 1.0, which is the same as σ_d in ABF for the denoising step. We then used an USM filter with standard deviation of 1.0, and two different weights: 2.0 and 4.0, to sharpen the denoised image. When applied to the second test image, we found that for both values of the sharpening weight, the ABF restored images appeared sharper on the edges/textures, and less noisy in the smooth regions. Due to space limitations, we do not show these results here. In the rest of this section, we compare the performance of the ABF with the conventional bilateral filter and the OUM.

The first test image is “Sculpture,” as shown in Fig. 7. The original image is a digital image taken by a Canon S30 digital camera. The degraded image is generated in the same way that the degraded training images are generated, which is described in Section IV-B. In Fig. 7, we compare the restoration results from the ABF, the OUM, and the bilateral filter ($\sigma_d = 1$ and $\sigma_r = 5$). The value for σ_d is the same as that for the ABF, and $\sigma_r = 5$ is the value that the ABF adopted for edge regions. The bilateral filter removes the noise in the degraded image; but the edges are as blurred as those in the degraded image. The OUM produces sharp edges, but with strong halos. The ABF removes the noise a little better than the OUM and the bilateral filter do.

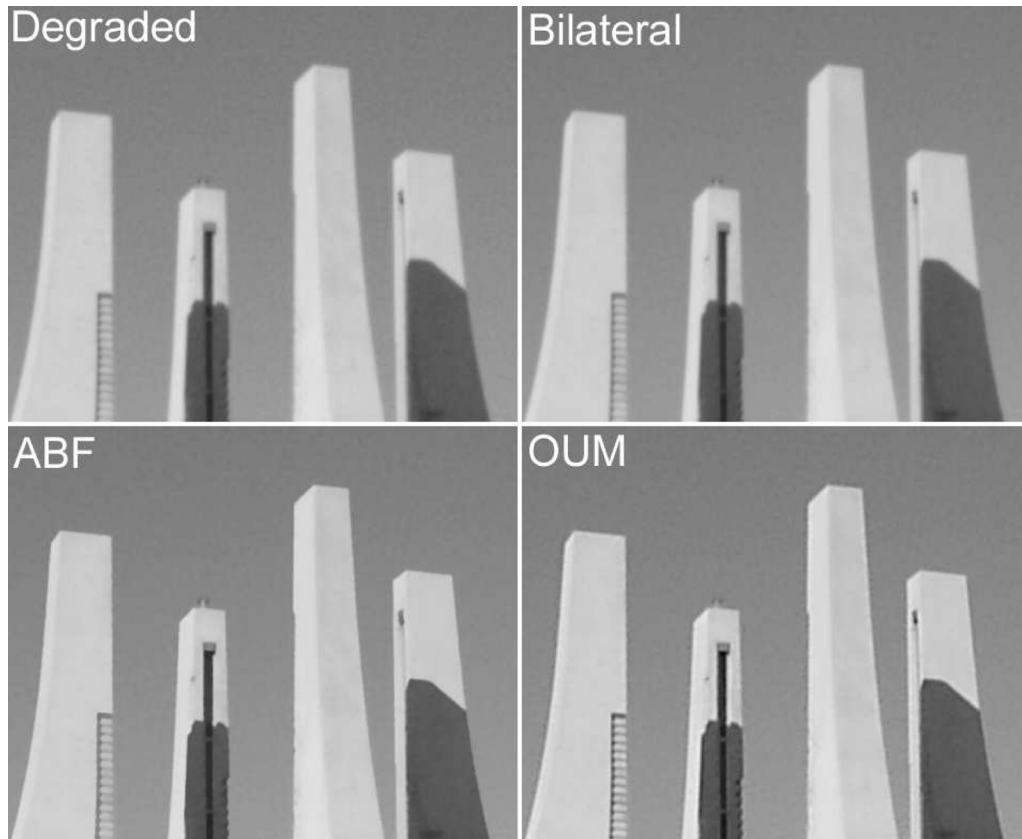


Fig. 7. Test image 1: Sculpture. “Sculpture” is degraded synthetically according to the model described in Section IV-B. Image size is 476×388 . Comparing the restoration results from the conventional bilateral filter ($\sigma_r = 5$, $\sigma_d = 1$), the ABF, and the OUM: the bilateral filter removes the noise in the degraded image, but does not make the edges sharper; the OUM produces halo around the edges; the ABF has the best noise suppression among the three algorithms. At the same time, it renders clean and crisp edges.

At the same time, it renders clean and sharp edges without the halo artifacts.

The second example is another natural image “Kids,” which is degraded by a real imaging pipeline. Kids was taken by a Canon Autoboy-A automatic analog camera, printed as a 4×6 in² photo, and scanned at 600 dpi [7]. Fig. 8(b) shows the edge profiles at edge pixel A inside the right marked region in Fig. 8(a). The bilateral filter does not change the slope of the edge. The OUM increases the slope but results in large overshoot and undershoot. The ABF enhances the slope the most without generating overshoot and undershoot around the edges. In Fig. 8(c)–(f), we zoom in to compare the rendering of the smooth regions, the edges, and the textured areas. Fig. 8(c) shows that both the ABF and the OUM significantly sharpen the edges of the text on the boy’s shirt. However, the text rendered by the ABF does not have the halo that appears in the OUM restored image. Both the ABF and the OUM are able to sharpen the grass in the background. The OUM rendered grass appears more contrasty; but the contrast of the ABF rendered grass is closer to that of the original image. Fig. 8(d) shows that the ABF is the most effective one of the three filters in removing the noise on the girl’s shirt. Fig. 8(e) shows that the ABF and the OUM both sharpen the contour of the eyes, but ABF does not have the halo artifacts. Fig. 8(f) shows that ABF renders the teeth in a more uniform way than does the OUM. The conventional bilateral filter does not restore the sharpness of the image. The reader may notice that in the grass region shown in Fig. 8(c), the ABF restored grass leaves exhibit some

posterization effect, which is caused by the fact that ABF sharpens edges by steepening the slope. Buades *et al.* [49] proposed a linear regression correction method that specifically addresses this problem, and which can reduce the posterization or staircasing effect commonly found in neighborhood filter denoised images.

The last test image is an example of a text image that was obtained by scanning a page of text with an HP Color LaserJet 4730 Multi-functional Printer at 600 dpi. This image was not generated according to the degradation model on which training of the ABF or OUM is based. The scanned text image, as shown in Fig. 9(a), appears blurry. In Fig. 9(b) and (c), we can see that both ABF and OUM significantly improve the sharpness of the degraded text image. The OUM rendered text looks more contrasty; but halos appear around the edges. The ABF restored text does not have any halos; and the contrast is closer to that of the original image.

As discussed in Section I, the bilateral filter has been widely applied since it was first proposed. The ABF is an extension of the traditional bilateral filter. With its capability of adaptive sharpening and smoothing, it offers the potential for wider application and better image quality. Although the ABF is developed in the framework of image restoration, it can be used as an image enhancing algorithm, as we have demonstrated with the test images. In particular, our main motivation for developing the ABF is to enhance the quality of digital images that consumers want to view and/or print. Therefore, we chose a noise model and a blur PSF that mimics a real world hybrid imaging

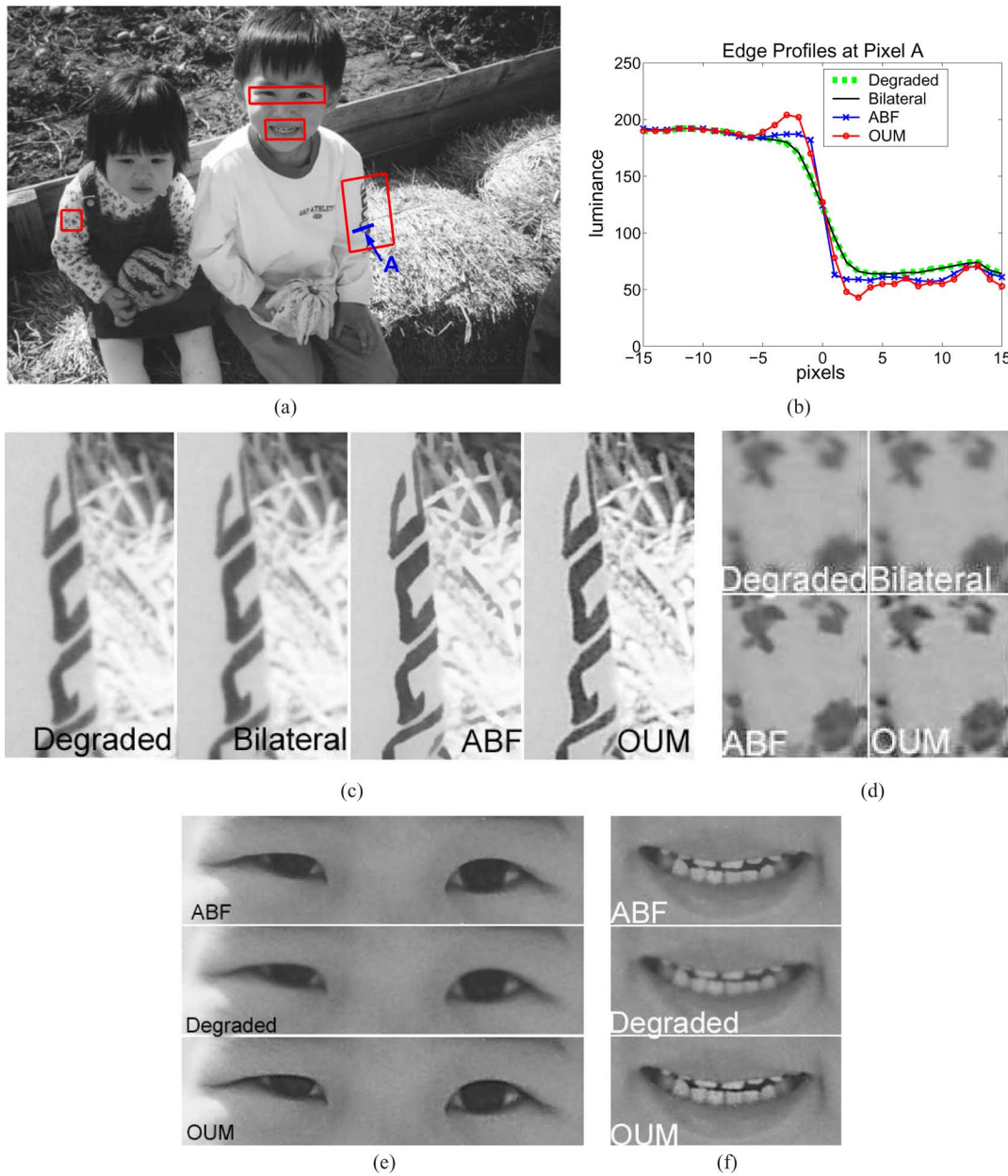


Fig. 8. Test image 2. (a) Test image “Kids” (3439 × 2344). (b) Edge normal profiles [52] taken at an edge pixel A at the tip of the blue arrow in the degraded image, the bilateral restored image, the ABF restored image, and the OUM restored image. The blue line at pixel A shows the edge normal direction along which the edge profiles are extracted. The regions inside the four boxes in (a) are zoomed in to compare results from the bilateral filter ($\sigma_r = 5$ and $\sigma_d = 1$), the OUM, and the ABF, as shown in (c)–(f). Since the bilateral filter does not restore the sharpness of the degraded image, we focus on the comparison between the ABF and the OUM in (e) and (f).

system to generate the training images. To understand the impact of the training images on the optimal filter parameters, we experimented with training the ABF with different sizes of the training set (from 1 image to 5 images), different image contents (two entirely different sets of training images, degraded by the same degradation model), and training with two different noise models (the same five training images, degraded by the tone-dependent silver halide noise model described in Section IV-B versus additive Gaussian noise with zero mean and standard deviation 6). We found that the training results are very consistent and converge as the size of the training set increases. Using training sets with different contents also leads to consistent training results. The degradation model used to gen-

erate the training images, however, does impact the training results. In particular, across our training set, the ratio of the RMSE (root mean square error) before restoration to the RMSE after restoration was 1.14 for the tone-dependent silver halide noise model and 1.36 for the additive Gaussian noise model, when the restoring algorithm was trained for the noise to which it was applied. When the restoring algorithm was applied to the noise model for which it was not trained, we obtained ratios of 1.12 and 1.24, respectively. As long as the ABF is optimized with training images that resemble the target images, it can effectively restore different images with different types of degradations. The ABF also demonstrates good robustness to the degradation model for the training set. In terms of RMSE, the im-



Fig. 9. Test image 3: Scanned Text (232×337). (a) Scanned text image. (b) ABF restored scanned text. (c) OUM restored scanned text. Please note that any JPEG artifacts that appear around the text edges are introduced by the PDF converter.

provement brought by the learning part seems rather modest. However, MSE or RMSE are known to not correlate well with perceived image quality. We feel that visual inspection of the test results presented in Section V provides a better indication of the quality of ABF restored images.

As an alternative to the training procedure, a user interface could be employed to obtain ζ and σ_r . For instance, the user could adjust two sets of slider bars to modify ζ and σ_r , respectively, for different classes of pixels and, hence, control the strength of the ABF for different regions of the image. For ζ , the mid-point of the sliding bar would be zero offset. Shifting the bar to the left/right would correspond to offsetting the range filter to the left/right on the histogram. Shifting the range filter away from the MEAN would sharpen the image. For σ_r , the left end of the slider bar would correspond to the minimal σ_r , while the right side would be maximal σ_r . The greater σ_r is, the more smoothing the ABF would do. In the case that an unusually large degree of smoothing is desired, the standard deviation of the domain Gaussian filter σ_d could also be made adjustable. In this way, the ABF could be incorporated into a photo editing tool such as Adobe Photoshop.

The ABF is efficient to implement. The time-consuming optimization of the ABF is done off-line. To restore an image with ABF, for each pixel, we first compute its LoG class; then determine the optimal ζ^* and σ_r^* for that pixel from a look-up table generated during the training procedure; and finally evaluate the ABF coefficients and apply them to the local data window. The offset is equivalent to an adjustment of the center pixel value. Therefore, once ζ^* and σ_r^* have been determined, the coefficients for the ABF are evaluated in the same way as they are for the classical bilateral filter. Some of the acceleration schemes proposed for the bilateral filter may also be able to speed up the ABF. Paris and Durand provide a good review [50] of many of these schemes.

In their own method, Paris and Durand extend the input image from 2-D (spatial) to 3-D (spatial and range), perform 3-D convolution in this 3-D space, and then “slice” the result to extract the desired bilateral filtered image values as a function of

two spatial dimensions. The gain in computational efficiency is achieved by approximating the 3-D convolution via subsampling. Using an Intel Xeon 2.8-GHz processor with a 1-MB cache, they reported processing a 2 mega-pixel image in less than a second to achieve image quality visually similar to the exact computation, which took several tens of minutes (varying with the implementation). Chen *et al.* further improved the performance by mapping the algorithm onto modern graphics hardware, obtaining running times on the order of a few milliseconds [30], [51]. For a rough comparison, the current ABF with a kernel size of 7×7 without optimization of the code nor any acceleration schemes, takes 59 s to process a 3 mega-pixel image with an Intel Xeon 3.2-GHz processor.

The parameter ζ from the ABF can be included in Paris and Durand’s framework by simply offsetting the slice by ζ . However, adaptation of the parameter σ_r from the ABF will destroy the 3-D convolution form. Thus, the full ABF cannot be accelerated by Paris and Durand’s approach. Since the adaptation provided by the offset ζ is the most important part of the ABF, their approach may still have merit for accelerating a restricted version of the ABF. How much is lost in terms of image quality by fixing σ_r could be a topic for future investigation.

VI. CONCLUSION

In this paper, we presented an adaptive bilateral filter (ABF). The ABF retains the general form of the bilateral filter [23], but contains two important modifications. First, an offset ζ is introduced to the range filter in the ABF. Second, both ζ and the width of the range filter σ_r in the ABF are locally adaptive. They depend on the local image structure, classified according to the output of the Laplacian of Gaussian (LoG) operator applied to the degraded image. The optimal ζ and σ_r are estimated in advance during the development of the algorithm via a off-line training procedure using a set of pairs of original and degraded images. During the training procedure, the MSE between the original and reconstructed images is minimized for each class of pixels.

The ABF outperforms the bilateral filter in noise removal. At the same time, it renders much sharper images than the bilateral filter does. As a result, the overall quality of the restored image is significantly improved. Compared with the OUM [7], an adaptive sharpening/denoising method based on the unsharp mask, the ABF restored edges are as sharp as the OUM restored edges, but without the halo artifacts that the OUM produces. The ABF also achieves better noise suppression than the OUM. The ABF sharpens an image by increasing the slope of the edges. Previous slope restoration algorithms involve complex algorithms to determine edge orientation and edge profiles, or compromise the performance by using 1-D horizontal and vertical projections of the edge profiles. In these approaches, the adjustment of the edge profiles tends to produce artifacts. In the ABF, we present a new approach to slope restoration: restoring edge slope by transforming the histogram of the edges. The ABF is efficient to implement, and provides a more reliable and more robust solution to slope restoration. We have demonstrated that the ABF works well for both natural images and text images.

For future development of the ABF, we would suggest that the following issues be addressed. First, the ABF tends to posterize the image, due to its fundamental mechanism of sharpening an image by pulling up or pushing down pixels along the edge slope. Second, the ABF does not perform as well at corners as it does on lines and spatially slow-varying curves, since the ABF is primarily based on transforming the histogram of the local data, which cannot effectively represent 2-D structures. Finally, in the current design of the ABF, a fixed domain Gaussian filter is used. It would be interesting to see what can be gained by jointly optimizing both the domain and the range filters.

ACKNOWLEDGMENT

The authors would like to thank the Hewlett-Packard Company for sponsoring this research. They would also like to thank Dr. S. Kim for providing the training images, the code of the OUM algorithm, and his valuable assistance with this research. They would also like to thank the anonymous reviewers of their paper for the many helpful suggestions.

REFERENCES

- [1] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. New York: Academic, 1982, vol. 1.
- [2] A. C. Bovik, Ed., "Regularization in image restoration and reconstruction," in *Handbook of Image & Video Processing*. San Diego, CA: Academic, 2000, ch. 3.6 (W. C. Karl), pp. 141–160.
- [3] C. B. Atkins, C. A. Bouman, J. P. Allebach, J. S. Gondek, M. T. Schramm, and F. W. Sliz, "Computerized Method for Improving Data Resolution," U.S. Patent 058248, 2000.
- [4] C. B. Atkins, C. A. Bouman, and J. P. Allebach, "Optimal image scaling using pixel classification," in *Proc. ICIP*, 2001, vol. 3, pp. 864–867.
- [5] H. Hu and G. de Haan, "Classification-based hybrid filters for image processing," in *Proc. SPIE Int. Soc. Opt. Eng.*, 2006, vol. 6077, pp. 607711–607711.
- [6] B. Zhang, J. Gondek, M. Schramm, and J. P. Allebach, "Improved resolution synthesis for image interpolation," in *Proc. IS&T's NIP22*, 2006, pp. 343–345.
- [7] S. Kim and J. P. Allebach, "Optimal unsharp mask for image sharpening and noise removal," *J. Electron. Imag.*, vol. 14, no. 2, pp. 023007–1, 2005.
- [8] Z. He and C. A. Bouman, "AM/FM halftoning: Digital halftoning through simultaneous modulation of dot size and dot density," *J. Electron. Imag.*, vol. 13, no. 2, pp. 286–302, 2004.
- [9] P. Li and J. P. Allebach, "Tone-dependent error diffusion," *IEEE Trans. Image Process.*, vol. 13, no. 2, pp. 201–215, Feb. 2004.
- [10] J. H. Lee and J. P. Allebach, "Inkjet printer model-based halftoning," *IEEE Trans. Image Process.*, vol. 14, no. 5, pp. 674–689, May 2005.
- [11] T. C. Chang and J. P. Allebach, "Quantization of accumulated diffused errors in error diffusion," *IEEE Trans. Image Process.*, vol. 14, no. 12, pp. 1960–1976, Dec. 2005.
- [12] V. Monga, N. Damera-Venkata, and B. L. Evans, "Design of tone-dependent color-error diffusion halftoning systems," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 198–211, Jan. 2007.
- [13] H. Siddiqui and C. A. Bouman, "Training-based descreeing," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 789–802, Mar. 2007.
- [14] H. Siddiqui and C. A. Bouman, "Training-based color correction for cellphone camera images," *Proc. ICASSP*, vol. 1, pp. 1733–1736, 2007.
- [15] A. C. Bovik, Ed., "Nonlinear filtering for image analysis and enhancement," in *Handbook of Image & Video Processing*. San Diego, CA: Academic, 2000, ch. 3.2 (G. R. Arce), pp. 95–95.
- [16] G. R. Arce and R. E. Foster, "Detail-preserving ranked-order based filters for image processing," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 1, pp. 83–98, Jan. 1989.
- [17] H. G. Longbotham and A. C. Bovik, "Theory of order statistic filters and their relationship to linear fir filters," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 37, no. 2, pp. 275–287, Feb. 1989.
- [18] M. Gabbouj, E. J. Coyle, and N. C. Gallagher, "An overview of median and stack filtering," *Circuits Syst. Signal Process.*, vol. 11, no. 1, pp. 7–45, 1992.
- [19] I. Pitas and A. N. Venetsanopoulos, "Order statistics in digital image processing," *Proc. IEEE*, vol. 80, no. 12, pp. 1893–1921, Dec. 1992.
- [20] E. A. Thompson, R. C. Hardie, and K. E. Barner, "Hybrid order statistic filter and its application to image restoration," *Appl. Opt.*, vol. 40, no. 5, pp. 656–661, 2001.
- [21] Y. H. Lee and S. A. Kassam, "Generalized median filtering and related nonlinear filtering techniques," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-33, no. 3, pp. 672–683, Jun. 1985.
- [22] S. M. Smith and J. M. Brady, "Susan—A new approach to low level image processing," *Int. J. Comput. Vis.*, vol. 23, no. 1, pp. 45–78, 1997.
- [23] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. Int. Conf. Comput. Vis.*, 1998, pp. 839–846.
- [24] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 950–953, 2003.
- [25] T. R. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 943–949, 2003.
- [26] S. Kim and K. Hong, "Composite video artifact removal by nonlinear bilateral filtering," in *Proc. SPIE Int. Soc. Opt. Eng.*, 2005, vol. 5960, pp. 306–315.
- [27] T. Q. Pham and L. J. V. Vliet, "Separable bilateral filtering for fast video preprocessing," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2005, pp. 454–457.
- [28] S. Yang and K. Hong, "Bilateral interpolation filters for image size conversion," in *Proc. ICIP*, 2005, vol. 2, pp. 986–989.
- [29] M. Elad, "Retinex by two bilateral filters," in *Proc. 5th Int. Conf. Scale-Space*, 2005, pp. 217–229.
- [30] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, "A gentle introduction to bilateral filtering and its applications," presented at the ACM SIGGRAPH, 2007.
- [31] M. Elad, "On the origin of the bilateral filter and the ways to improve it," *IEEE Trans. Image Process.*, vol. 11, no. 10, pp. 1141–1151, Oct. 2002.
- [32] A. K. Jain, *Fundamentals of Digital Image Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1989.
- [33] R. Lagendijk and J. Biemond, *Iterative Identification and Restoration of Images*. Norwell, MA: Kluwer, 1991.
- [34] N. Sochen, R. Kimmel, and R. Malladi, "A geometrical framework for low level vision," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 310–318, Mar. 1998.
- [35] N. Sochen, R. Kimmel, and A. M. Bruckstein, "Diffusions and confusions in signal and image processing," *J. Math. Imag. Vis.*, vol. 14, no. 3, pp. 195–209, 2001.
- [36] D. Barash, "A fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 6, pp. 844–847, Jun. 2002.
- [37] D. Barash and D. Comaniciu, "A common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift," *Image Vis. Comput.*, vol. 22, no. 1, pp. 73–81, 2004.
- [38] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 7, pp. 629–639, Jul. 1990.

- [39] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Trans. Image Process.*, vol. 7, no. 3, pp. 421–432, Mar. 1998.
- [40] P. Saint-Marc, J. Chen, and G. Medioni, "Adaptive smoothing: a general tool for early vision," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 6, pp. 514–529, Jun. 1991.
- [41] D. Comaniciu and P. Meer, "Mean shift: a robust approach towards feature space," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002.
- [42] M. Aleksic, M. Smirnov, and S. Goma, "Novel bilateral filter approach: Image noise reduction with sharpening," in *Proc. SPIE Int. Soc. Opt. Eng.*, 2006, vol. 6069, pp. 60690F1–60690F7.
- [43] A. Das and R. M. Rangayyan, "Enhancement of image edge sharpness and acutance," in *Proc. SPIE Int. Soc. Opt. Eng.*, 1997, vol. 3026, pp. 133–142.
- [44] J. Tegenbosch, P. Hofman, and M. Bosma, "Improving nonlinear up-scaling by adapting to the local edge orientation," in *Proc. SPIE Int. Soc. Opt. Eng.*, 2004, vol. 5308, pp. 1181–1190.
- [45] H. Kotera, Y. Yamada, and K. Shimo, "Sharpness improvement adaptive to edge strength of color image," in *Proc. IS&T/SID Eighth Color Imaging Conf.*, 2000, pp. 149–154.
- [46] H. Kotera and W. Hui, "Multi-scale image sharpening with noise reduction," in *Proc. IS&T's NIP18*, 2002, pp. 590–594.
- [47] S. Guillon, P. Baylou, M. Najim, and N. Keskes, "Adaptive nonlinear filters for 2D and 3D image enhancement," *Signal Process.*, vol. 67, pp. 237–254, 1998.
- [48] R. Gonzalez and R. Woods, *Digital Image Processing*. Norwell, MA: Addison Wesley, 1992.
- [49] A. Buades, B. Coll, and J.-M. Morel, "The staircasing effect in neighborhood filters and its solution," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1499–1505, Jun. 2006.
- [50] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *Int. J. Comput. Vis.*
- [51] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Trans. Graph.*, vol. 26, no. 3, 2007.
- [52] B. Zhang, J. P. Allebach, and Z. Pizlo, "An investigation of perceived sharpness and sharpness metrics," in *Proc. SPIE Int. Soc. Opt. Eng.*, 2005, vol. 5668, pp. 98–110.



color science.

Dr. Zhang was the recipient of the 2005 Raymond Davis Scholarship from the Society for Imaging Science and Technology (IS&T).



Buyue Zhang (S'98–M'06) received the B.S. and M.S. degrees from Tsinghua University, China, in 1999 and 2001, respectively, majoring in control theory and control engineering, and the Ph.D. degree from School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, in 2007.

She is currently with the DSPS R&D center, Texas Instruments, Inc. Her research interests include signal and image processing, image quality and psychophysics, electronic imaging systems, and

Jan P. Allebach (F'91) received the B.S.E.E. from the University of Delaware, Newark, and the Ph.D. degree from Princeton University, Princeton, NJ, in 1972 and 1976, respectively.

He was on the faculty at the University of Delaware from 1976 to 1983. Since 1983, he has been at Purdue University, West Lafayette, IN, where he is the Michael J. and Katherine R. Birck Professor of Electrical and Computer Engineering. His current research interests include image rendering, image quality, color imaging and color measurement,

printer and sensor forensics, and digital publishing.

Prof. Allebach is a Fellow the Society for Imaging Science and Technology (IS&T) and SPIE. He has been especially active with the IEEE Signal Processing Society and IS&T. He has served as Distinguished/Visiting Lecturer for both societies, and has served as an officer and on the Board of Directors of both societies. He is a past Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and the IEEE TRANSACTIONS ON IMAGE PROCESSING. He is presently Editor for the IS&T/SPIE *Journal of Electronic Imaging*. He received the Senior (best paper) Award from the IEEE Signal Processing Society and the Bowman Award from IS&T. In 2004, he was named Electronic Imaging Scientist of the Year by IS&T and SPIE. He is the recipient of four teaching awards from Purdue University and co-recipient, with five other Purdue faculty members, of the 2006 Purdue College of Engineering Team Award in recognition of his long-term work with HP. In 2007, he was named Honorary Member of IS&T the highest award that IS&T bestows.