[26] D. L. Donoho and I. M. Johnstone, "Adapting to unknown smoothness via wavelet shrinkage," *J. Amer. Statist. Assoc.*, vol. 90, no. 432, pp. 1200–1224, 1995.
[27] A. Benazza-Benyahia and J.-C. Pesquet, "Building robust wavelet estimators for multicomponent images using Stein^{\prime}s principle," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1814–1830, Nov. 2005.
[28] F. Luisier, T. Blu, and M. Unser, "A new SURE approach to image denoising: Interscale orthonormal wavelet thresholding," *IEEE Trans. Image Process.*, vol. 16, no. 3, pp. 593–606, Mar. 2007.
[29] S. Ramani, T. Blu, and M. Unser, "Monte-Carlo SURE: A black-box optimization of regularization parameters for general denoising algorithms," *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1540–1554, Sep. 2008.
[30] D. Van De Ville and M. Kocher, "SURE-based non-local means," *IEEE Signal Process. Lett.*, vol. 16, no. 11, pp. 973–976, Nov. 2009.
[31] T. Blu and F. Luisier, "The SURE-LET approach to image denoising," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2778–2786, Nov. 2007.
[32] J. Salmon, "On two parameters for denoising with non-local means," *IEEE Signal Process. Lett.*, vol. 17, no. 3, pp. 269–272, Mar. 2010.
[33] J. Salmon and Y. Strozecki, "From patches to pixels in non-local methods: Weighted-average reprojection," in *Proc. 2010 IEEE 17th Int. Conf. Image Process.*, 2010, pp. 1929–1932.

Fast Bilateral Filter With Arbitrary Range and Domain Kernels

Bahadir K. Gunturk, *Senior Member, IEEE*

***Abstract*—In this paper, we present a fast implementation of the bilateral filter with arbitrary range and domain kernels. It is based on the histogram-based fast bilateral filter approximation that uses uniform box as the domain kernel. Instead of using a single box kernel, multiple box kernels are used and optimally combined to approximate an arbitrary domain kernel. The method achieves better approximation of the bilateral filter compared to the single box kernel version with little increase in computational complexity. We also derive the optimal kernel size when a single box kernel is used.**

***Index Terms*—Image enhancement, nonlinear filtering.**

I. Introduction

The bilateral filter is a nonlinear weighted averaging filter, where the weights depend on both the spatial distance and the intensity distance with respect to the center pixel. The main feature of the bilateral filter is its ability to preserve edges while doing spatial smoothing. The term "bilateral filter" was first used by Tomasi and Manduchi in [1]; the same filter was earlier called the SUSAN (Smallest Univalue Segment Assimilating Nucleus) filter by Smith and Brady in [2]. The variants of the bilateral filter have been published even earlier as the sigma filter [3] and the neighborhood filter [4].

Manuscript received April 14, 2010; revised November 02, 2010; accepted February 18, 2011. Date of publication March 14, 2011; date of current version August 19, 2011. This work was supported in part by the National Science Foundation under Grant 0528785 and the National Institutes of Health under Grant 1R21AG032231-01. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Luminita Aura Vese.

The author is with the Department of Electrical and Computer Engineering, Louisiana State University, Baton Rouge, LA 70803 USA (e-mail: bahadir@ece.lsu.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.2011.2126585

At a pixel location $\mathbf{x} = (x_1, x_2)$, the output of the bilateral filter is calculated as follows:

$$\hat{I}(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} K_d(\|\mathbf{y} - \mathbf{x}\|) K_r(|I(\mathbf{y}) - I(\mathbf{x})|) I(\mathbf{y}) \quad (1)$$

where $K_d(\cdot)$ is the spatial domain kernel, $K_r(\cdot)$ is the intensity range kernel, $\mathcal{N}(\mathbf{x})$ is the set of pixels within a spatial neighborhood of $\mathbf{x}$, and $C(\mathbf{x})$ is the normalization term

$$C(\mathbf{x}) = \sum_{\mathbf{y} \in \mathcal{N}(\mathbf{x})} K_d(\|\mathbf{y} - \mathbf{x}\|) K_r(|I(\mathbf{y}) - I(\mathbf{x})|). \quad (2)$$

The kernels $K_d(\cdot)$ and $K_r(\cdot)$ determine how the spatial and intensity differences are treated. The contribution (weight) of a pixel $I(\mathbf{y})$ is determined by the product of $K_d(\cdot)$ and $K_r(\cdot)$. The bilateral filter in [1] uses the Gaussian kernel, $G_\sigma(z) = \exp(-z^2/2\sigma^2)$, for both the domain and range kernels:

$$K_d(\|\mathbf{y} - \mathbf{x}\|) = G_{\sigma_d}(\|\mathbf{y} - \mathbf{x}\|) \quad (3)$$

and

$$K_r(|I(\mathbf{y}) - I(\mathbf{x})|) = G_{\sigma_r}(|I(\mathbf{y}) - I(\mathbf{x})|). \quad (4)$$

On the other hand, the sigma filter [3] and the neighborhood filter [4] use different kernels. The sigma filter [3] first calculates the local standard deviation around $I(\mathbf{x})$; the standard deviation is then used to determine a threshold value for pixel intensities, and pixels that are within the threshold of the center pixel $I(\mathbf{x})$ are averaged (with equal weights) to calculate the filter output at that pixel. In case of the neighborhood filter [4], the range kernel is a Gaussian as in (3), and the spatial kernel is a uniform box kernel. Among different kernel options, the Gaussian kernel is the most popular choice for both the range and spatial kernels, as it gives an intuitive and simple control of the behavior of the filter with two parameters, σ_d and σ_r.

The bilateral filter has found a wide range of applications in image processing and computer vision. The immediate application of the bilateral filter is image denoising as it can do spatial averaging without blurring edges. [5] presents a multiresolution extension of the bilateral filter for image denoising and an empirical study on optimal parameter selection. It is shown that the optimal value of σ_d is relatively insensitive to noise power, while the optimal σ_r value is linearly proportional to the noise standard deviation. Other applications of bilateral filter include tone mapping in high-dynamic range imaging [6], contrast enhancement [7], [8], fusion of flash and no-flash images [9], [10], fusion of visible spectrum and infrared spectrum images [11], compression artifact reduction [12], 3-D mesh denoising [13], [14], depth map estimation [15], video stylization [16], video enhancement [17], texture and illumination separation [18], orientation smoothing [19], and optical flow estimation [20].

This paper presents a fast approximation of the bilateral filter with arbitrary range and domain kernels. It is based on a method presented by Porikli in [21]. The method in [21] (which uses a box domain kernel) is extended by optimally combining multiple box kernels to approximate an arbitrary domain kernel. As there is no restriction on the range kernel either, any range and domain kernels can be used with this fast bilateral filter implementation. Section II reviews the fast bilateral filter techniques in the literature. The proposed method is explained in Section III. In Section IV, the question of optimal kernel size in case of a single box kernel is addressed. Section V provides some experimental results, and Section VI concludes the paper.

II. Fast Bilateral Filter Methods

The direct implementation of the bilateral filter is given here.

- For each pixel $\mathbf{x} \in \mathcal{S}$, where $\mathcal{S}$ is the set of all pixels in the image
- Initialize: $\hat{I}(\mathbf{x}) = 0$ and $C(\mathbf{x}) = 0$.
- For each $\mathbf{y}$ in the local neighborhood $\mathcal{N}$ of $\mathbf{x}$:
  - Calculate the weight: $w = K_d(\|\mathbf{y} - \mathbf{x}\|)K_r(|I(\mathbf{y}) - I(\mathbf{x})|)$.
  - Update: $\hat{I}(\mathbf{x}) = \hat{I}(\mathbf{x}) + wI(\mathbf{y})$.
  - Update: $C(\mathbf{x}) = C(\mathbf{x}) + w$.
- Normalize: $\hat{I}(\mathbf{x}) = \hat{I}(\mathbf{x})/C(\mathbf{x})$.

The computational complexity of this implementation is $\mathcal{O}(|\mathcal{S}||\mathcal{N}|)$, where $|\mathcal{S}|$ is the number of pixels in the entire image and $|\mathcal{N}|$ is the number of pixels in the neighborhood $\mathcal{N}$. The local neighborhood is typically chosen such that $\|\mathbf{y}-\mathbf{x}\| \leq 3\sigma_d$; therefore, the neighborhood size $|\mathcal{N}|$ is proportional to σ_d^2. While the overall complexity $\mathcal{O}\left(|\mathcal{S}|\sigma_d^2\right)$ is acceptable for small σ_d, it quickly becomes limiting with increasing σ_d. To address this issue, a number of fast implementation/approximation methods have been proposed.

A. Kernel Separation

One method of speeding up the bilateral filter is to separate the 2-D filter kernel into two 1-D kernels. First, the rows of an image are filtered; the result is then filtered along the columns [22], [23]. This reduces the complexity to $\mathcal{O}(|\mathcal{S}|\sigma_d)$. Although its performance is good in smooth regions and horizontal/vertical edges, the algorithm may not perform satisfactorily on texture regions because of 1-D handling of spatial domain.

B. Bilateral Grid

Another fast bilateral filter algorithm is obtained through representing an image in a 3-D space, where the signal intensity is added to the spatial domain as the third dimension [24]. This vector representation can be used to interpret the bilateral filter as linear filtering the entries of a vector-valued image separately, followed by point-by-point division. Because the linear filtering involved is low-pass filtering, the results are bandlimited and can be represented well with their low-frequency components. Therefore, the 3-D grids can be downsampled without losing much performance to speed up the algorithm. The work in [24] proposes downsampling of the spatial domain $\mathcal{S}$ by σ_d and the intensity range $\mathcal{R}$ by σ_r. The complexity of the algorithm then becomes $\mathcal{O}(|\mathcal{S}| + (|\mathcal{S}|/\sigma_d)(|\mathcal{R}|/\sigma_r))$.

C. Polynomial Representation of Range Filter

In [21], the author presents two approaches for a fast bilateral filter. The first approach does not have any restriction on the domain filter, but the range kernel is approximated with a polynomial. Doing a Taylor series expansion on the Gaussian range filter, it turns out that the bilateral filter could be approximated through linear filtering images I, I^2, I^3, etc., and point-by-point multiplication/division of the results. The performance of this algorithm is good for small σ_r, but it degrades quickly for large σ_r since the polynomial representation does not approximate the Gaussian well. The use of higher order polynomials should improve the results.

D. Local-Histogram-Based Bilateral Filter With Box Spatial Filter

The second approach presented in [21] is based on the use of a uniform box kernel for the domain kernel. With the box kernel $B_N(\cdot)$, the bilateral filter can be written in terms of local histograms as

$$\hat{I}_N(\mathbf{x}) = \frac{1}{C_N(\mathbf{x})} \sum_{\mathbf{y}} B_N(\mathbf{y} - \mathbf{x}) K_r(|I(\mathbf{y}) - I(\mathbf{x})|) I(\mathbf{y})$$

$$= \frac{1}{C_N(\mathbf{x})} \sum_{i=0}^{255} i H_{\mathbf{x},N}(i) K_r(|i - I(\mathbf{x})|) \tag{5}$$

where $i \in \mathbb{Z}$ in $[0, 255]$ for an 8-bit image, $H_{\mathbf{x},N}$ is the local histogram in the $2N + 1 \times 2N + 1$ neighborhood around $\mathbf{x}$, $B_N(\cdot)$ is the box kernel, given by

$$B_N(\mathbf{z}) = \begin{cases} 1, |z_j| \leq N, & j = 1, 2 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

and $C_N(\mathbf{x})$ is the normalization term, given by

$$C_N(\mathbf{x}) = \sum_{i=0}^{255} H_{\mathbf{x},N}(i) K_r(|i - I(\mathbf{x})|). \tag{7}$$

There are several advantages of this formulation [21], including those given here.

1) $K_r(|i - I(\mathbf{x})|)$ can be calculated for all values of i and at all locations $\mathbf{x}$ independently and therefore in parallel. Similarly, $H_{\mathbf{x},N}(i)$ and $iH_{\mathbf{x},N}(i)$, which are also required in the bilateral filter calculation (5), can be obtained independent of $K_r(|i - I(\mathbf{x})|)$.
2) $H_{\mathbf{x},N}(i)$ can be calculated efficiently using the integral histogram technique [25].
3) The algorithm can be further speeded up by quantizing the histogram.

As a result, the *box* bilateral filter can be implemented independent of the kernel size. The method can be applied to 16-bit images (e.g., in medical, satellite, and microscopy, applications) as well and extended to color images through either processing each color channel separately [21] or using 3-D integral histograms [25]. A similar histogram-based method was proposed in [26], although the histogram computation is not as efficient as the one in [21], which also reports that the box bilateral filter outperforms other fast bilateral filters available at the time in terms of speed-versus-quality tradeoff characteristics.

In addition to the above-mentioned methods, there are some other fast bilateral filter implementations. In [6], the signal intensities are quantized such that the final output is obtained through linear interpolation of a set of linear filter outputs. [27] shows that this method can be used with constant-time bilateral filters and further improve the speed. In [28], the block size is adjusted to gain speed improvements, while [29] considers the memory usage as well as the accuracy of implementation. In [30], compared with the regular grid representation of the bilateral grid [24], a more efficient Gaussian kd-tree representation is used to improve over [24]. An alternative high-dimensional representation is presented in [31], where the permutohedral lattice representation is shown to be more efficient than the Gaussian kd-tree for large filter size.

III. Proposed Method

Our method is based on the histogram-based method of [21]. Instead of a single box bilateral filter, we would like to approximate the bilateral filter as a weighted sum of multiple box bilateral filters

$$\hat{I}(\mathbf{x}) \approx \sum_{m=0}^{M} k_m \hat{I}_m(\mathbf{x}). \tag{8}$$

Note that, in this equation, $\hat{I}_0(\mathbf{x})$ is the input image $I(\mathbf{x})$ itself. This is not inconsistent with our previous definition of $\hat{I}_m(\mathbf{x})$ because $B_0(\cdot)$ is a Kronecker delta function according to (6), and therefore $\hat{I}_0(\mathbf{x})$ becomes $I(\mathbf{x})$ in (5).

We will show that $\sum_{m=0}^{M} k_m \hat{I}_m(\mathbf{x})$ can approximate $\hat{I}(\mathbf{x})$ better than any box bilateral filter $\hat{I}_m(\mathbf{x})$, and the computational cost is not increased significantly. Now, the question that needs to be answered is the optimal values of k_m. First, we write (8) as

$$\begin{aligned}
&\frac{1}{C(\mathbf{x})}\sum_{\mathbf{y}} G_{\sigma_d}(\|\mathbf{y}-\mathbf{x}\|)K_r(|I(\mathbf{y})-I(\mathbf{x})|)I(\mathbf{y}) \\
&\quad\approx \sum_{m=0}^{M} k_m \frac{1}{C_m(\mathbf{x})}\sum_{\mathbf{y}} B_m(\mathbf{y}-\mathbf{x}) \\
&\qquad \times K_r(|I(\mathbf{y})-I(\mathbf{x})|)I(\mathbf{y}) \\
&\quad= \sum_{\mathbf{y}}\left(\sum_{m=0}^{M}\frac{k_m}{C_m(\mathbf{x})}B_m(\mathbf{y}-\mathbf{x})\right. \\
&\qquad\left. \times K_r(|I(\mathbf{y})-I(\mathbf{x})|)\right)I(\mathbf{y}) \qquad (9)
\end{aligned}$$

where we switched the order of summations in the last line. To minimize the approximation error at an arbitrary point, the coefficients of $I(\mathbf{y})$ on both sides of (9) should be close to each other. To keep a simple notation, we absorb the normalization coefficients into k_m; in other words, we redefine k_m as $k_m C(\mathbf{x})/C_m(\mathbf{x})$. Then, we want the following coefficients to be close:

$$G_{\sigma_d}(\|\mathbf{y}-\mathbf{x}\|)K_r(|I(\mathbf{y})-I(\mathbf{x})|) \qquad (10)$$

and

$$\sum_{m=0}^{M} k_m B_m(\mathbf{y}-\mathbf{x})K_r(|I(\mathbf{y})-I(\mathbf{x})|). \qquad (11)$$

Therefore, we define the error function to be minimized as

$$E(k_0,\ldots,k_M)=\sum_{\mathbf{z}\in\mathcal{N}}\left(G_{\sigma_d}(\|\mathbf{z}\|)-\sum_{m=0}^{M}k_m B_m(\mathbf{z})\right)^2 \qquad (12)$$

where $\mathcal{N}=\{\mathbf{z}_1,\mathbf{z}_2,\ldots,\mathbf{z}_{|\mathcal{N}|}\}$ is the set of pixels within the support of the domain kernels.

Minimization of this error function is equivalent to solving the following linear set of equations:

$$\underbrace{\begin{bmatrix} G_{\sigma_d}(\|\mathbf{z}_1\|) \\ G_{\sigma_d}(\|\mathbf{z}_2\|) \\ \vdots \\ G_{\sigma_d}(\|\mathbf{z}_{|N|}\|)\end{bmatrix}}_{\mathbf{g}} = \underbrace{\begin{bmatrix} B_0(\mathbf{z}_1) & B_1(\mathbf{z}_1) & \cdots & B_M(\mathbf{z}_1) \\ B_0(\mathbf{z}_2) & B_1(\mathbf{z}_2) & \cdots & B_M(\mathbf{z}_2) \\ \vdots & \vdots & \ddots & \vdots \\ B_0(\mathbf{z}_{|N|}) & B_1(\mathbf{z}_{|N|}) & \cdots & B_M(\mathbf{z}_{|N|})\end{bmatrix}}_{\mathbf{B}}\underbrace{\begin{bmatrix} k_0 \\ k_1 \\ \vdots \\ k_M\end{bmatrix}}_{\mathbf{k}} \qquad (13)$$

whose solution is $\mathbf{k}=(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{g}$. This is independent of the input image and only requires the parameters σ_d, the size of the neighborhood $\mathcal{N}$, and the number of box filters parameter M.

For a given image, $\hat{I}_1(\mathbf{x}), \hat{I}_2(\mathbf{x}),\ldots,\hat{I}_M(\mathbf{x})$ are calculated, and a weighted sum of these images and the input image produces the filter output. The box bilateral filter implementation involves calculation of integral histogram, calculation of local histogram $H_{\mathbf{x},N}$, and evaluation of (5). For the proposed bilateral filter, the local histogram calculation and the evaluation of (5) are repeated M times; the integral histogram is calculated once and not repeated. The repetitions do not add much computational cost: the local histograms can be obtained by linear filtering the integral histogram, and the evaluation of (5) requires point-by-point multiplications of arrays that can be obtained in parallel. Calculation of $\mathbf{k}$ is not an issue either as it can be done offline and saved in a lookup table. In our unoptimized MATLAB implementation (on a 2.4-GHz machine), the box bilateral filter takes 0.61 s for a 256×256 image with histogram quantization level of 15. The proposed bilateral filter adds 0.07 s for each additional M; that is, it takes 0.68 s for $M=2$, 0.75 s for $M=3$, and so on. Through optimization of the codes, it is possible to reduce these numbers further.

The memory usage of the histogram-based method [21] is about $Q\times|\mathcal{N}|$, where Q is the number of histogram bins, and $|\mathcal{N}|$ is the number of pixels in the image, and it is basically allocated to store the integral histogram. We should note that this method is not the most memory-efficient fast bilateral filter; for example, [27] has a memory usage of $4\times|\mathcal{N}|$, and [29] has a memory usage of $Q\times|\mathcal{N}|^{1/2}+Q$. The proposed method is based on [21], and has similar memory usage. The memory space $Q\times|\mathcal{N}|$ allocated for the integral histogram does not change; if the output image of each box filter is stored separately as it is done in this paper, then the additional memory requirement is $M\times|\mathcal{N}|$; that is, the total memory usage of the proposed method is $(Q+M)\times|\mathcal{N}|$. The algorithm could be implemented in a more memory-efficient way by not saving the output of each filter separately but updating single output as different size box filters are applied.

IV. What is the Best Single Box Bilateral Filter?

In the previous section, we showed how to optimally combine multiple box bilateral filters. If we are supposed to use a single box bilateral filter, then we would like to know what box size we should choose. The optimal box size can be derived using the error function (12). In the Appendix, we write the continuous version of (12) and derive the formula for the optimal box size, and it turns out that N should be $1.4\sigma_d$, which needs to be rounded to the nearest integer in practice.

V. Experiments and Analysis

We compared the outputs of the box bilateral filter and the proposed bilateral filter with respect to the output of the standard bilateral filter for different values of σ_r, σ_d, and histogram quantization levels. Fig. 1 shows the PSNR values of the proposed bilateral filter ($M=5$) and five box bilateral filters ($N=1,\ldots,5$) for $\sigma_r=15$, 25, 75, and $\sigma_d=0.8,1.0,\ldots,3.0$. For each σ_d value, the k_m values are calculated as discussed in the previous section. It is seen that the proposed bilateral filter outperforms the box bilateral filter regardless of the size of the box. The best box bilateral filter depends on the σ_d value. In Tables I and II, we compared the PSNR of the proposed bilateral with the best possible PSNR that can be achieved with a box bilateral filter for $\sigma_r=5$, 15, 25, 50, and 75 and $\sigma_d=0.8,1.0,\ldots,3.0$. It is observed that, for small σ_r, the proposed bilateral filter does not improve much over the box bilateral filter. The reason is that the defining factor in the bilateral filter kernel becomes the range parameter σ_r if its value is small; therefore, it does not make much difference whether the domain kernel is approximated well or not. For large values of σ_r, σ_d becomes more important, and the improvement of the proposed bilateral filter is more pronounced. Comparing Tables I and II, it is also observed that with higher number of quantization levels (finer quantization of intensities), the improvement of the proposed bilateral filter increases on average.

Fig. 2 shows a visual comparison of the standard, proposed, and box bilateral filters. While the outputs of the proposed bilateral filter and

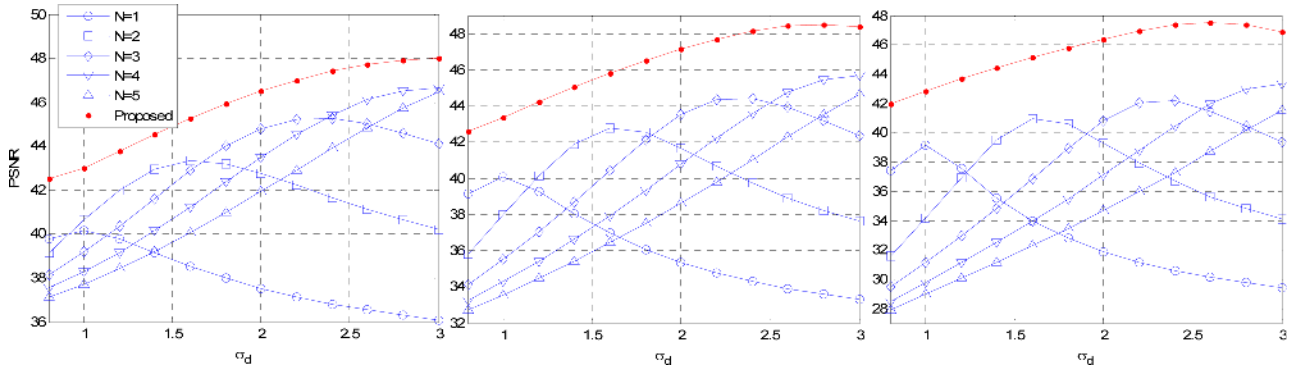

Fig. 1. Approximation of the standard bilateral filter with the box bilateral filters and the proposed method. The test image is the standard *Mandrill* image, which is of size 512 × 512. To speed up the implementation, the intensity values are quantized to 15 bins. Left: $\sigma_r = 15$; middle: $\sigma_r = 25$; right: $\sigma_r = 75$. In each subfigure, blue lines are box bilateral filters for $N = 1, \ldots, 5$; red lines are proposed bilateral filter with $M = 5$.

TABLE I
APPROXIMATION OF THE STANDARD BILATERAL FILTER. BEST BOX: MAXIMUM PSNR THAT CAN BE ACHIEVED WITH A BOX FILTER. PROPOSED: PSNR OF THE PROPOSED FILTER. THE QUANTIZATION LEVELS IS 15. THE RESULTS ARE AVERAGE VALUES FOR THE BARBARA, LENA, BOAT, GOLDHILL, AND MANDRILL TEST IMAGES

		σ_d												
σ_r		0.8	1	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8	3	Average
5	Best Box	37.01	37.00	37.01	37.03	37.05	37.08	37.10	37.12	37.14	37.15	37.17	37.18	37.09
5	Proposed	37.82	37.16	36.96	36.91	36.92	36.95	36.99	37.02	37.05	37.07	37.09	37.11	37.09
15	Best Box	39.72	40.43	41.19	41.72	42.03	42.45	42.85	43.13	43.30	43.62	43.86	43.98	42.36
15	Proposed	41.80	41.73	42.03	42.42	42.80	43.15	43.46	43.73	43.94	44.11	44.23	44.28	43.14
25	Best Box	39.46	39.97	40.59	41.57	41.96	42.06	42.69	43.08	43.22	43.45	43.82	43.96	42.15
25	Proposed	42.08	42.14	42.52	42.96	43.38	43.77	44.12	44.43	44.68	44.86	44.94	44.92	43.73
50	Best Box	38.78	39.56	39.05	40.56	41.31	41.10	41.58	42.28	42.26	42.19	42.84	42.98	41.21
50	Proposed	41.91	42.04	42.44	42.87	43.29	43.67	44.03	44.36	44.62	44.77	44.75	44.54	43.61
75	Best Box	38.43	39.30	38.30	40.01	40.88	40.45	40.97	41.80	41.70	41.46	42.24	42.36	40.66
75	Proposed	41.79	41.93	42.31	42.72	43.11	43.48	43.84	44.16	44.41	44.52	44.41	44.06	43.39

TABLE II
APPROXIMATION OF THE STANDARD BILATERAL FILTER. BEST BOX: MAXIMUM PSNR THAT CAN BE ACHIEVED WITH A BOX FILTER. PROPOSED: PSNR OF THE PROPOSED FILTER. THE QUANTIZATION LEVELS IS 25. THE RESULTS ARE AVERAGE VALUES FOR THE BARBARA, LENA, BOAT, GOLDHILL, AND MANDRILL TEST IMAGES

		σ_d												
σ_r		0.8	1	1.2	1.4	1.6	1.8	2	2.2	2.4	2.6	2.8	3	Average
5	Best Box	45.23	45.49	45.80	46.13	46.45	46.75	47.02	47.28	47.51	47.71	47.89	48.04	46.78
5	Proposed	45.16	45.12	45.43	45.83	46.22	46.59	46.91	47.18	47.41	47.59	47.74	47.85	46.58
15	Best Box	43.84	44.49	45.19	46.55	47.13	47.11	47.92	48.52	48.57	48.80	49.37	49.52	47.25
15	Proposed	46.80	47.34	48.04	48.73	49.37	49.94	50.45	50.89	51.26	51.50	51.60	51.52	49.79
25	Best Box	43.15	44.10	43.41	45.38	46.32	45.93	46.40	47.30	47.31	47.12	47.84	48.06	46.03
25	Proposed	46.67	47.28	47.97	48.62	49.21	49.74	50.22	50.66	51.01	51.20	51.14	50.81	49.54
50	Best Box	41.93	43.33	41.46	43.44	44.77	44.00	44.24	45.47	45.33	44.68	45.69	45.89	44.19
50	Proposed	46.15	46.75	47.35	47.90	48.38	48.82	49.26	49.68	49.99	50.03	49.67	48.91	48.58
75	Best Box	41.39	42.83	40.41	42.57	44.01	42.90	43.30	44.66	44.30	43.54	44.70	44.80	43.28
75	Proposed	45.94	46.51	47.06	47.55	47.98	48.38	48.79	49.20	49.47	49.37	48.75	47.70	48.06

the standard bilateral filter are very similar, the box bilateral filter loses some texture details. The best box bilateral filter is with $N = 2$, which is also predicted by the formula derived in the Appendix.

In Fig. 3, we investigate the effect of the number of box filters on PSNR performance. It is seen that M should be sufficiently large to achieve the best possible performance. (A rule of thumb deducted from the experiments is that M should be at least $2\sigma_d$ to have the best performance.) If M is not sufficiently large compared with σ_d, the proposed filter cannot approximate the standard bilateral filter well. This is the reason we are seeing drops in the PSNR curves for large values of σ_d in both Figs. 1 and 3. On the other hand, using more than a sufficient number of boxes does not degrade the performance, as we would expect.

As a final experiment, we compared the standard, box, and proposed bilateral filters in a denoising example. As seen in Fig. 4, the proposed bilateral filter works very similar to the standard bilateral filter, while box bilateral filter may over-blur or under-work.

VI. CONCLUSION

In this paper, we presented an extension of the box bilateral filter to approximate the bilateral filter with arbitrary range and domain kernels. Although we demonstrated the performance improvement for the Gaussian kernel only, the proposed approximation method could be applied to other symmetric kernels as well. The method would be important especially in applications where the shape of the domain kernel

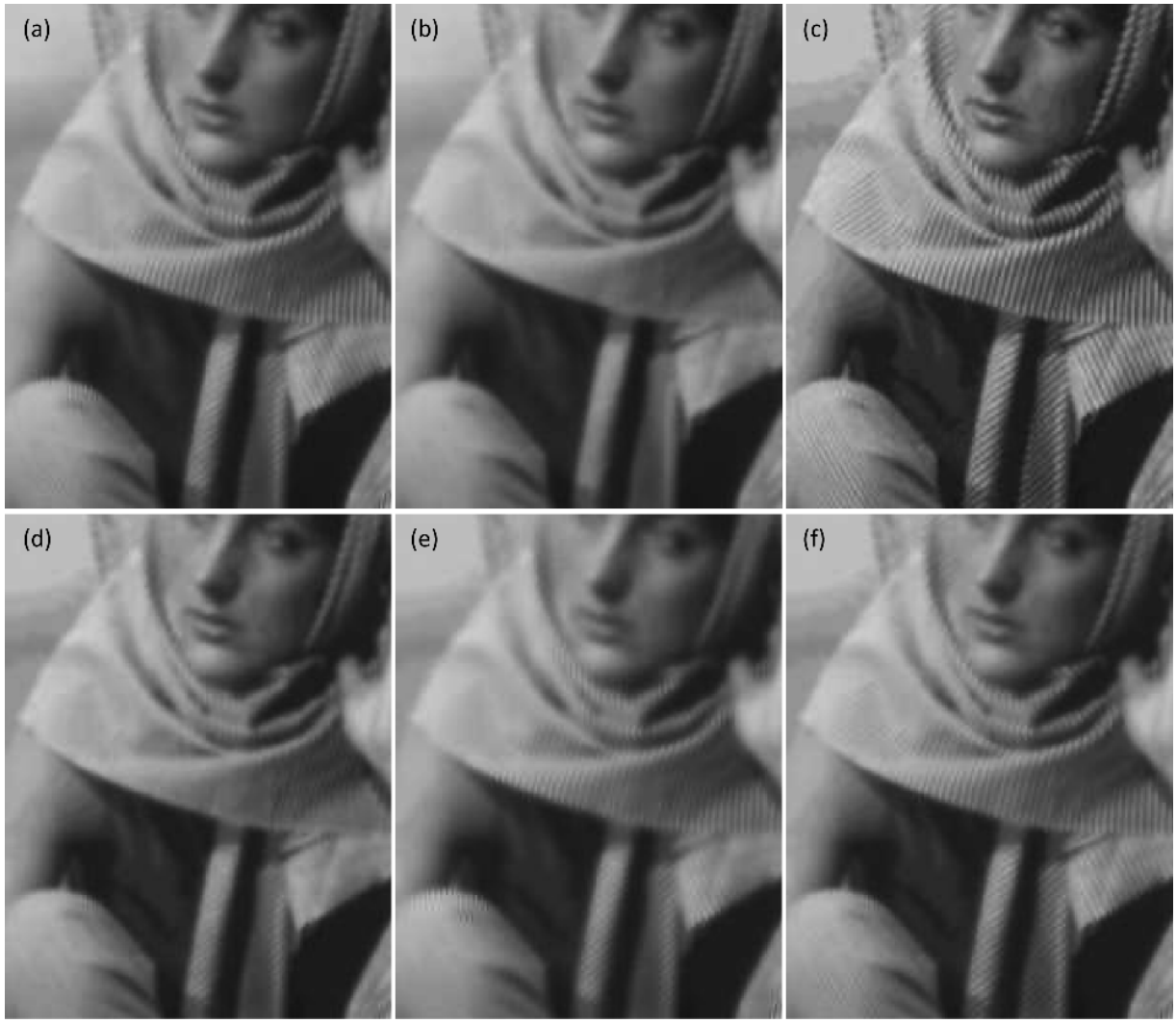

Fig. 2. Visual and quantitative comparison of the box and proposed bilateral filters with respect to the standard bilateral filter. (a) Standard bilateral filter. (b) Gaussian low-pass filter ($\mathrm{PSNR} = 33.82$ dB). (c) Box bilateral filter with $N = 1$ ($\mathrm{PSNR} = 34.44$ dB). (d) Box bilateral filter with $N = 2$ ($\mathrm{PSNR} = 39.68$ dB). (e) Box bilateral filter with $N = 3$ ($\mathrm{PSNR} = 37.94$ dB). (f) Proposed bilateral filter with $M = 5$ ($\mathrm{PSNR} = 42.28$ dB). The parameters are as follows: the quantization level is 15, $\sigma_r = 100$, and $\sigma_d = 1.6$.

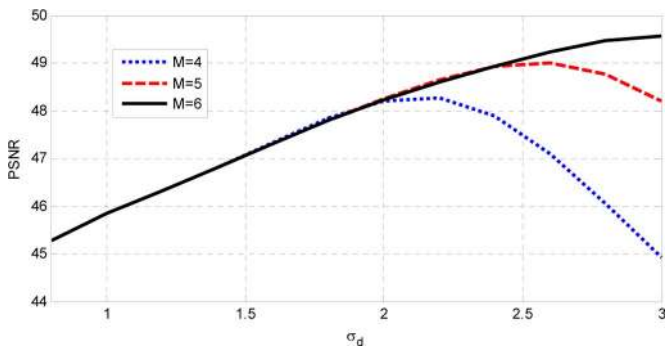

Fig. 3. Effect of the number of box filters on the approximation of the standard bilateral filter. Quantization level is 25. $\sigma_r = 50$. (The test image is the standard *Barbara* image of size 512×512.)

is critical. One such example is the image denoising method of [32], where the domain kernel is spatially adapted.

We included a set of experiments to demonstrate and analyze certain features of the method. The additional computational cost over the single box bilateral filter is not much and would be bearable in applications where the domain kernel accuracy is crucial. Our implementation was done in MATLAB without any optimization; this is not a concern in this paper since the goal is to present the theoretical aspect.

A side information that can be deducted from the experiments is the optimal box size when single box bilateral filter is used: Fig. 1 shows the best possible box size for various σ_d and σ_r values. A quick look into the plots reveals that the best box size depends on σ_s, and someone can come up with a rule of thumb for choosing the box size based on σ_d. It is indeed possible (as shown in the Appendix) to derive optimal box size using the error function defined to obtain the optimal combination when multiple box kernels are used. We have observed that the experimental data is consistent with the theoretically predicted box sizes.

Finally, we should note that the derivations are based on the domain kernel of the bilateral filter. We may expect deviations from the theoretical predictions based on the content of an image since the range kernel is multiplied with the domain kernel to form the overall kernel.

Fig. 4. Denoising example. (a) Input image. (b) Noisy image with noise standard deviation of 15 ($\mathrm{PSNR} = 24.64$ dB). (c) Box bilateral filter with $N = 1$ ($\mathrm{PSNR} = 28.44$). (d) Box bilateral filter with $N = 2$ ($\mathrm{PSNR} = 28.56$). (e) Box bilateral filter with $N = 3$ ($\mathrm{PSNR} = 28.55$). (f) Box bilateral filter with $N = 4$ ($\mathrm{PSNR} = 28.42$). (g) Box bilateral filter with $N = 5$ ($\mathrm{PSNR} = 28.22$). (h) Standard bilateral filter ($\mathrm{PSNR} = 28.84$). (i) Proposed bilateral filter with $M = 5$ ($\mathrm{PSNR} = 28.73$). The quantization level is 15, $\sigma_r = 30$, and $\sigma_d = 1.8$.

APPENDIX

To derive the best single box kernel to approximate a Gaussian kernel with spatial parameter σ_d, we write the 1-D continuous version of the error function (12) for a single box kernel as

$$\begin{aligned} E(k_m, m) &= \int_{-\infty}^{\infty} (G_{\sigma_d}(z) - k_m B_m(z))^2 dz \\ &= 2\int_{0}^{\infty} (G_{\sigma_d}(z) - k_m B_m(z))^2 dz \\ &= 2\int_{0}^{m} (G_{\sigma_d}(z) - k_m)^2 dz + 2\int_{m}^{\infty} (G_{\sigma_d}(z))^2 dz \end{aligned} \tag{14}$$

where m is now an unknown nonnegative real number and $B_m(z)$ is equal to 1 for $|z| \leq m$ and 0 otherwise. We would like to find what value of m minimizes this error function. At an extrema, the gradient of the error function must be equal to zero; therefore, taking the derivative of $E(k_m, m)$ with respect to k_m and m, we get the following equations:

$$\begin{aligned} \frac{\partial E(k_m, m)}{\partial k_m} &= -4\int_{0}^{m} (G_{\sigma_d}(z) - k_m) dz \\ &= -4\int_{0}^{m} G_{\sigma_d}(z) dz + 4\int_{0}^{m} k_m dz \\ &= -4\int_{0}^{m} G_{\sigma_d}(z) dz + 4mk_m, \\ \Rightarrow k_m &= \frac{1}{m}\int_{0}^{m} G_{\sigma_d}(z) dz \end{aligned} \tag{15}$$

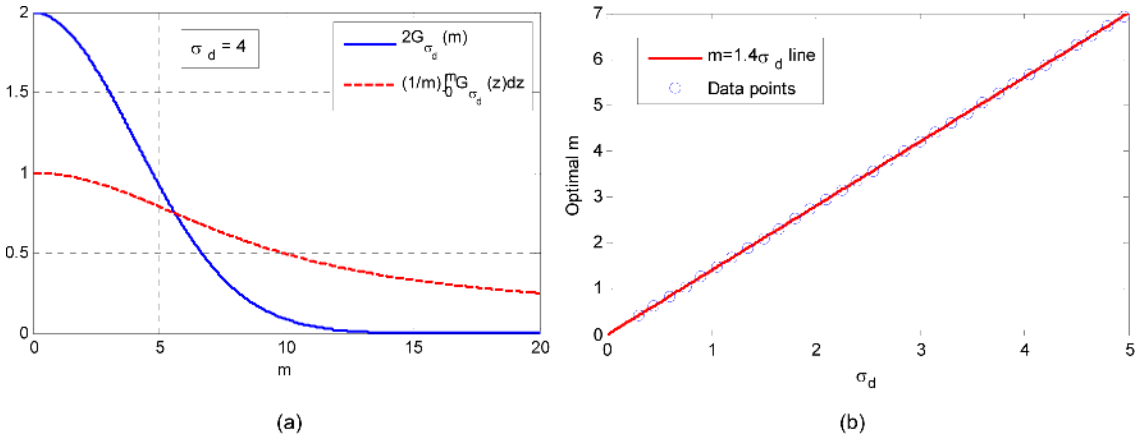

Fig. 5. (a) Left and right sides of (17) for $\sigma_d = 4$. The intersection gives the optimal m value. (b) By repeating this process, we can obtain the optimal m values for different σ_d values. The data points show the optimal m values as a function σ_d values.

and

$$\begin{aligned}\frac{\partial E(k_m, m)}{\partial m} &= 2(G_{\sigma_d}(m) - k_m)^2 - 2(G_{\sigma_d}(m))^2 \\ &= -4G_{\sigma_d}(m)k_m + 2k_m^2 \\ \Rightarrow\ k_m &= 2G_{\sigma_d}(m). \end{aligned} \tag{16}$$

In other words, the optimal m should satisfy the following condition:

$$2G_{\sigma_d}(m) = \frac{1}{m}\int_0^m G_{\sigma_d}(z)dz. \tag{17}$$

We have numerically found the solution to this integral equation. Fig. 5(a) plots the left and right sides of (17) as a function of m for a particular σ_d value; the intersection of these curves gives the solution for m. Repeating this procedure for different values of σ_d, we obtain a plot for the optimal m as a function of σ_d. Fig. 5(b) verifies that the $m = 1.4\sigma_d$ line describes the relation very well. In practice, we need an integer-sized box kernel; therefore, $1.4\sigma_d$ should be rounded to the nearest integer to determine the best box kernel size.

REFERENCES

[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE Int. Conf. Comput. Vis.*, Bombay, India, Jan. 1998, pp. 839–846.

[2] 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, May 1997.

[3] J. S. Lee, "Digital image smoothing and the sigma filter," *CVGIP: Graphical Models and Image Process.*, vol. 24, no. 2, pp. 255–269, Nov. 1983.

[4] L. Yaroslavsky, *Digital Picture Processing—An Introduction*. Berlin, Germany: Springer-Verlag, 1985.

[5] M. Zhang and B. K. Gunturk, "Multiresolution bilateral filtering for image denoising," *IEEE Trans. Image Process.*, vol. 17, no. 12, pp. 2324–2333, Dec. 2008.

[6] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *Proc. ACM SIGGRAPH*, San Antonio, TX, Jul. 2002, pp. 257–266.

[7] M. Elad, "Retinex by two bilateral filters," in *Proc. Scale-Space and PDE Methods Comput. Vis.*, Hofgeismar, Germany, Apr. 2005, vol. 3459, pp. 217–229.

[8] S. Bae, S. Paris, and F. Durand, "Two-scale tone management for photographic look," in *Proc. ACM SIGGRAPH*, Boston, MA, Jul. 2006, pp. 637–645.

[9] E. Eisemann and F. Durand, "Flash photography enhancement via intrinsic relighting," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, Aug. 2004, pp. 673–678.

[10] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama, "Digital photography with flash and no-flash image pairs," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, Aug. 2004, pp. 664–672.

[11] E. P. Bennett, J. L. Mason, and L. McMillan, "Multispectral video fusion," in *Proc. ACM SIGGRAPH*, Boston, MA, Jul. 2006, pp. 123–123.

[12] M. Zhang and B. K. Gunturk, "Compression artifact reduction with adaptive bilateral filtering," in *Proc. SPIE Electron. Imaging*, San Jose, CA, 2009, vol. 7257, pp. 72571A–72571A.

[13] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," in *Proc. ACM SIGGRAPH*, San Diego, CA, Jul. 2003, pp. 950–953.

[14] T. R. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing," in *Proc. ACM SIGGRAPH*, San Diego, CA, Jul. 2003, pp. 943–949.

[15] E. A. Khan, E. Reinhard, R. Fleming, and H. Buelthoff, "Image-based material editing," in *Proc. ACM SIGGRAPH*, Boston, MA, Jul. 2006, pp. 654–663.

[16] H. Winnemoller, S. C. Olsen, and B. Gooch, "Real-time video abstraction," in *Proc. ACM SIGGRAPH*, Boston, MA, Jul. 2006, pp. 1221–1226.

[17] E. P. Bennett and L. McMillan, "Video enhancement using per-pixel virtual exposures," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, Jul. 2005, pp. 845–852.

[18] B. M. Oh, M. Chen, J. Dorsey, and F. Durand, "Image-based modeling and photo editing," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, Aug. 2001, pp. 433–442.

[19] S. Paris, H. Briceno, and F. Sillion, "Capture of hair geometry from multiple images," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, Jul. 2004, pp. 712–719.

[20] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi, "Bilateral filtering-based optical flow estimation with occlusion detection," in *Proc. Eur. Conf. Comput. Vis.*, Graz, Austria, May 2006, vol. 1, pp. 211–224.

[21] F. Porikli, "Constant time O(1) bilateral filtering," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, Anchorage, AK, Jun. 2008, pp. 1–8.

[22] T. Q. Pham and L. J. Vliet, "Separable bilateral filtering for fast video preprocessing," in *Proc. IEEE Int. Conf. Multimedia and Expo.*, Amsterdam, The Netherlands, Jul. 2005, pp. 1–4.

[23] T. Q. Pham, *Spatiotonal Adaptivity in Super-Resolution of Under-Sampled Image Sequences*. Delft, The Netherlands: Delft Univ. of Technol., 2006.

[24] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," in *Proc. Eur. Conf. Comput. Vis.*, Graz, Austria, May 2006, pp. 568–580.

[25] F. Porikli, "Integral histogram: A fast way to extract histograms in Cartesian spaces," in *Proc. Int. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, Jun. 2005, pp. 829–836.

[26] B. Weiss, "Fast median and bilateral filtering," in *Proc. ACM SIGGRAPH*, Boston, MA, Jul. 2006, pp. 519–526.

[27] Q. Yang, K.-H. Tan, and N. Ahuja, "Real-time $O(1)$ bilateral filtering," in *Proc. Comput. Vis. Pattern Recognit.*, 2009, pp. 557–564.

[28] W. Yu, F. Franchetti, J. C. Hoe, Y.-J. Chang, and T. Chen, "Fast bilateral filtering by adapting block size," in *Proc. IEEE Int. Conf. Image Process.*, 2010, pp. 3281–3284.

[29] M. Igarashi, M. Ikebe, S. Shimoyama, K. Yamano, and J. Motohisa, "$O(1)$ bilateral filtering with low memory usage," in *Proc. IEEE Int. Conf. Image Process.*, 2010, pp. 3301–3304.

[30] A. Adams, N. Gelfand, J. Dolson, and M. Levoy, "Gaussian kd-trees for fast high-dimensional filtering," in *Proc. ACM SIGGRAPH*, New Orleans, LA, Aug. 2009, vol. Article 21, pp. 1–12.

[31] A. Adams, J. Baek, and A. Davis, "Fast high-dimensional filtering using the permutohedral lattice," in *Proc. Eurographics*, May 2010, vol. 29, pp. 753–762.

[32] C. Kervrann and J. Boulanger, "Optimal spatial adaptation for patch-based image denoising," *IEEE Trans. Image Process.*, vol. 15, no. 10, pp. 2866–2878, Oct. 2006.