Fast $O(1)$ Bilateral Filtering Using Trigonometric Range Kernels

Kunal Narayan Chaudhury, *Student Member, IEEE*, Daniel Sage, and Michael Unser, *Fellow, IEEE*

***Abstract*—It is well known that spatial averaging can be realized (in space or frequency domain) using algorithms whose complexity does not scale with the size or shape of the filter. These fast algorithms are generally referred to as constant-time or $O(1)$ algorithms in the image-processing literature. Along with the spatial filter, the edge-preserving bilateral filter involves an additional range kernel. This is used to restrict the averaging to those neighborhood pixels whose intensity are similar or close to that of the pixel of interest. The range kernel operates by acting on the pixel intensities. This makes the averaging process nonlinear and computationally intensive, particularly when the spatial filter is large. In this paper, we show how the $O(1)$ averaging algorithms can be leveraged for realizing the bilateral filter in constant time, by using trigonometric range kernels. This is done by generalizing the idea presented by Porikli, i.e., using polynomial kernels. The class of trigonometric kernels turns out to be sufficiently rich, allowing for the approximation of the standard Gaussian bilateral filter. The attractive feature of our approach is that, for a fixed number of terms, the quality of approximation achieved using trigonometric kernels is much superior to that obtained by Porikli using polynomials.**

***Index Terms*—Bilateral filter, constant-time algorithm, edge-preserving smoothing, $O(1)$ complexity, raised cosines.**

I. Introduction

THE bilateral filtering of an image $f(\boldsymbol{x})$ in the general setting is given by

$$\tilde{f}(\boldsymbol{x}) = \eta^{-1} \int w(\boldsymbol{x},\boldsymbol{y})\, \phi\left(f(\boldsymbol{x}), f(\boldsymbol{y})\right)\ f(\boldsymbol{y})\, d\boldsymbol{y}$$

where

$$\eta = \int w(\boldsymbol{x},\boldsymbol{y})\, \phi\left(f(\boldsymbol{x}), f(\boldsymbol{y})\right)\ d\boldsymbol{y}.$$

In this formula, $w(\boldsymbol{x},\boldsymbol{y})$ measures the geometric proximity between the pixel of interest $\boldsymbol{x}$, and nearby pixel $\boldsymbol{y}$. Its role is to localize the averaging to a neighborhood of $\boldsymbol{x}$. On the other hand, function $\phi(u,v)$ measures the similarity between the intensity of the pixel of interest $f(\boldsymbol{x})$ and its neighbor $f(\boldsymbol{y})$. Normalizing factor η is used to preserve constants and, in particular, the local mean.

In this paper, we consider the so-called *unbiased form* of the bilateral filter [1], where $w(\boldsymbol{x},\boldsymbol{y})$ is translation invariant, that is, $w(\boldsymbol{x},\boldsymbol{y}) = w(\boldsymbol{x}-\boldsymbol{y})$, and where the range filter is symmetric and depends on the difference of intensity, $\phi(f(\boldsymbol{x}), f(\boldsymbol{y})) = \phi(f(\boldsymbol{x}) - f(\boldsymbol{y}))$. In this case, the filter is given by

$$\tilde{f}(\boldsymbol{x}) = \eta^{-1} \int_{\Omega} w(\boldsymbol{y})\phi\left(f(\boldsymbol{x}-\boldsymbol{y}) - f(\boldsymbol{x})\right) f(\boldsymbol{x}-\boldsymbol{y})\, d\boldsymbol{y} \quad (1)$$

where

$$\eta = \int_{\Omega} w(\boldsymbol{y})\phi\left(f(\boldsymbol{x}-\boldsymbol{y}) - f(\boldsymbol{x})\right)\ d\boldsymbol{y}. \quad (2)$$

We call $w(\boldsymbol{x})$ the *spatial kernel*, and $\phi(s)$ the *range kernel*. The local support Ω of the spatial kernel specifies the neighborhood over which the averaging takes place. A popular form of the bilateral filter is one where both $w(\boldsymbol{x})$ and $\phi(s)$ are Gaussian [1]–[4].

The edge-preserving bilateral filter was originally introduced by Tomasi and Manduchi in [1] as a simple noniterative alternative to anisotropic diffusion [5]. This was motivated by the observation that while standard spatial averaging performs well in regions with homogenous intensities, it tends to poorly perform in the vicinity of sharp transitions, such as edges. For the bilateral filter in (1), the difference $f(\boldsymbol{x}-\boldsymbol{y})-f(\boldsymbol{x})$ is close to zero in homogenous regions, and hence, $\phi(f(\boldsymbol{x}-\boldsymbol{y})-f(\boldsymbol{x})) \approx 1$. In this case, (1) simply results in the averaging of pixels in the neighborhood of the pixel of interest. On the other hand, if the pixel of interest $\boldsymbol{x}$ is in the vicinity of an edge, $\phi(f(\boldsymbol{x}-\boldsymbol{y})-f(\boldsymbol{x}))$ is large when $\boldsymbol{x}-\boldsymbol{y}$ belongs to the same side of the edge as $\boldsymbol{x}$ and is small when $\boldsymbol{x}-\boldsymbol{y}$ is on the other side of the edge. As a result, the averaging is restricted to neighborhood pixels that are on the same side of the edge as the pixel of interest. This is the basic idea that allows one to perform smoothing while preserving edges at the same time. Since its inception, the bilateral filter has found widespread use in several image processing, computer graphics, and computer vision applications. This includes denoising [6], video abstraction [7], demosaicing [8], optical-flow estimation [9], and stereo matching [10], to name a few. More recently, the bilateral filter has been extended by Baudes *et al.* [3] to realize the popular nonlocal neighborhood filter, where the similarity between pixels is measured using patches centered around the pixels.

Manuscript received December 10, 2010; revised May 25, 2011; accepted June 01, 2011. Date of publication June 09, 2011; date of current version November 18, 2011. This work was supported by the Swiss National Science Foundation under Grant 200020-109415. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Oscar. C. Au.

K. N. Chaudhury is with the Program in Applied and Computational Mathematics, Princeton University, Princeton, NJ 08544-1000 USA (e-mail: kchaudhu@princeton.edu).

D. Sage and M. Unser are with the Biomedical Imaging Group, Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland.

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

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.2011.2159234

The direct implementation of (1) turns out to be rather computationally intensive for real-time applications. Several efficient numerical schemes have been proposed in the past for implementing the filter in real time, even at video rates [11]–[14]. These algorithms (with the exception of [11]), however, do not scale well with the size of the spatial kernel, and this limits their usage in high-resolution applications. A significant advance was obtained when Porikli [2] proposed a constant-time implementation of the bilateral filter (for arbitrary spatial kernels) using polynomial range kernels. The $O(1)$ algorithm was also extended to include Gaussian $\phi(s)$ by locally approximating it using polynomials. More recently, Yang *et al.* [4] have proposed a $O(1)$ algorithm for arbitrary range and spatial kernels by extending the bilateral filtering method of Durand and Dorsey [11]. Their algorithm is based on a piecewise-linear approximation of the bilateral filter obtained by quantizing $\phi(s)$.

In this paper, we extend the $O(1)$ algorithm of Porikli to provide an exact implementation of the bilateral filter, using trigonometric range kernels. Our main observation is that trigonometric functions share a common property of polynomials, which allows one to "linearize" the otherwise nonlinear bilateral filter. The common property is that the translation of a polynomial (resp. trigonometric function) is again a polynomial (resp. trigonometric function) and, importantly, of the same degree. By fixing $\phi(s)$ to be a trigonometric function, we show how this self-shiftable property can be used to (locally) linearize the bilateral filter. This is the crux of the idea that was used for deriving the $O(1)$ algorithm for polynomial $\phi(s)$ in [2].

II. Constant-Time Bilateral Filter

A. Main Idea

It is the presence of term $\phi(f(\boldsymbol{x}-\boldsymbol{y})-f(\boldsymbol{x}))$ in (1) that makes the filter nonlinear. In the absence of this term, that is, when $\phi(s)$ is constant, the filter is simply given by averaging

$$\overline{f}(\boldsymbol{x}) = \int_{\Omega} w(\boldsymbol{y}) f(\boldsymbol{x}-\boldsymbol{y})\, d\boldsymbol{y} \tag{3}$$

where we assume $w(\boldsymbol{x})$ to have a total mass of unity. It is well known that (3) can be implemented in constant time, irrespective of the size and shape of the filter, using the convolution-multiplication property of the (fast) Fourier transform. The number of computations required per pixel, however, depends on the size of the image in this case [15]. On the other hand, it is known that (3) can be realized at the cost of a constant number of operations per pixel (independent of the size of the image and the filter) using recursive algorithms. These $O(1)$ recursive algorithms are based on specialized kernels, such as the box and the hat function [16]–[18], and the more general class of Gaussian-like box splines [19].

Our present idea is to leverage these fast averaging algorithms by expressing (1) in terms of (3), where the averaging is performed on the image and its simple pointwise transforms. Our observation is that we can do so if the range kernel is of the form

$$\phi(s) = \cos(\gamma s) \qquad (-T \le s \le T). \tag{4}$$

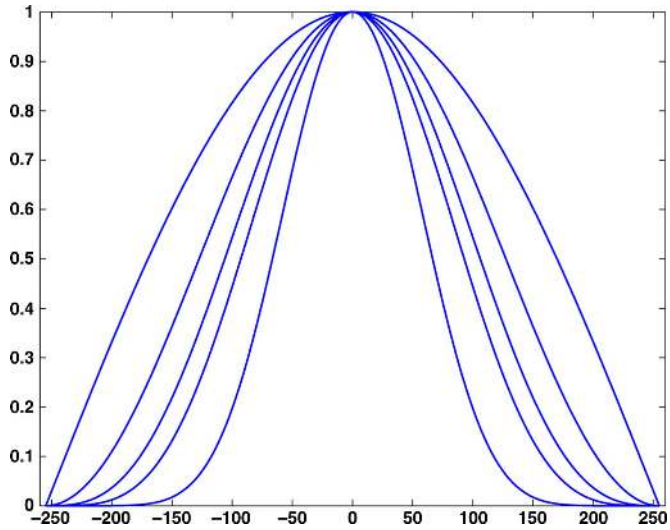

Fig. 1. Family of raised cosines $g(s) = [\cos(\gamma s)]^N$ over the dynamic range $-T \le s \le T$ as N goes from 1 to 5 (outer to inner curves). We set $T = 255$ corresponding to the maximum dynamic range of a grayscale image, and $\gamma = \pi/2T$. They satisfy the two essential properties required to qualify as a valid range kernel of the bilateral filter—nonnegativity and monotonicity (decay). Moreover, they have the remarkable property that they converge to a Gaussian (after appropriate normalization) as N gets large; see (7).

By plugging (4) into (1), we can write the integral as

$$\cos(\gamma f(\boldsymbol{x})) \int_{\Omega} w(\boldsymbol{y}) \cos(\gamma f(\boldsymbol{x}-\boldsymbol{y}))\, f(\boldsymbol{x}-\boldsymbol{y})\, d\boldsymbol{y} + \sin(\gamma f(\boldsymbol{x})) \int_{\Omega} w(\boldsymbol{y}) \sin(\gamma f(\boldsymbol{x}-\boldsymbol{y}))\, f(\boldsymbol{x}-\boldsymbol{y})\, d\boldsymbol{y}.$$

This is clearly shown to be the linear combination of two spatial averages, performed on images $\cos(\gamma f(\boldsymbol{x}))f(\boldsymbol{x})$ and $\sin(\gamma f(\boldsymbol{x}))f(\boldsymbol{x})$. Similarly, we can write the integral in (2) as

$$\cos(\gamma f(\boldsymbol{x})) \int_{\Omega} w(\boldsymbol{y}) \cos(\gamma f(\boldsymbol{x}-\boldsymbol{y}))\, d\boldsymbol{y} + \sin(\gamma f(\boldsymbol{x})) \int_{\Omega} w(\boldsymbol{y}) \sin(\gamma f(\boldsymbol{x}-\boldsymbol{y}))\, d\boldsymbol{y}.$$

In this case, the averaging is on images $\cos(\gamma f(\boldsymbol{x}))$ and $\sin(\gamma f(\boldsymbol{x}))$. This is the trick that allows us to express (1) in terms of linear convolution filters applied to pointwise transforms of the image.

Note that the domain of $\phi(s)$ is $[-T, T]$ in (4). We assume here (without loss of generality) that the dynamic range of the image is within $[0, T]$. The maximum of $|f(\boldsymbol{x}) - f(\boldsymbol{y})|$ over all $\boldsymbol{x}$ and $\boldsymbol{y}$ such that $\boldsymbol{x}-\boldsymbol{y} \in \Omega$ is within T in this case. Therefore, by letting $\gamma = \pi/2T$, we can guarantee the argument γs of the cosine function to be within a range of $[-\pi/2, \pi/2]$. The crucial point here is that the cosine function is oscillating and can assume negative values over $(-\infty, \infty)$. However, its restriction over the half-period $[-\pi/2, \pi/2]$ has two essential properties of a range kernel—it is nonnegative and has a bump shape (cf. the outermost curve in Fig. 1). Note that, in practice, the bound on the local variations of intensity could be much lower than T.

B. General Trigonometric Kernels

The aforementioned idea can be easily extended to more general trigonometric functions of form $\phi(s) = a_0 + a_1\cos(\gamma s) + \cdots + a_N\cos(N\gamma s)$. This is most conveniently done by writing $\phi(s)$ in terms of complex exponentials, namely

$$\phi(s) = \sum_{|n|\leq N} c_n \exp(jn\gamma s). \tag{5}$$

Coefficients c_n must be real and symmetric since $\phi(s)$ is real and symmetric. Now, using the addition–multiplication property of exponentials, we can write the following:

$$\phi\left(f(\boldsymbol{x}-\boldsymbol{y}) - f(\boldsymbol{x})\right) = \sum_{|n|\leq N} d_n(x)\ \exp\left(jn\gamma f(\boldsymbol{x}-\boldsymbol{y})\right)$$

where $d_n(\boldsymbol{x}) = c_n\exp(-jn\gamma f(\boldsymbol{x}))$. Plugging this into (1), we immediately see that

$$\tilde{f}(\boldsymbol{x}) = \frac{\sum_{|n|\leq N}\ d_n(\boldsymbol{x})\overline{g_n}(\boldsymbol{x})}{\sum_{|n|\leq N}\ d_n(\boldsymbol{x})\overline{h_n}(\boldsymbol{x})} \tag{6}$$

where $h_n(\boldsymbol{x}) = \exp(jn\gamma f(\boldsymbol{x}))$ and $g_n(\boldsymbol{x}) = f(\boldsymbol{x})h_n(\boldsymbol{x})$. We refer to $h_n(\boldsymbol{x})$ and $g_n(x)$ as the *auxiliary images*, and N as the *degree* of the kernel.

The above analysis gives us the following $O(1)$ algorithm for the bilateral filter: We first set up the auxiliary images and coefficients $d_n(\boldsymbol{x})$ from the input image. We then average each of the auxiliary images using a $O(1)$ algorithm (this can be done in parallel). The samples of the filtered image is then given by the simple sum and division in (6). In particular, for an image of size $M \times M$, we can compute the spatial averages for any arbitrary $w(\boldsymbol{x})$ at the cost of $O(M^2\log_2 M)$ operations using the Fourier transform. As mentioned earlier, this can be further reduced to a total of $O(M^2)$ operations using specialized spatial kernels [16], [17], [19].

C. Raised Cosines

We now address the fact that $\phi(s)$ must have some additional properties to qualify as a valid range kernel (aside from being symmetric). Namely, $\phi(s)$ must be nonnegative and must be monotonic in that $\phi(s_1) \leq \phi(s_2)$ whenever $|s_1| > |s_2|$. In particular, it must have a peak at the origin. This ensures that large differences in intensity gets more penalized than small differences and that (1) purely behaves as a spatial filter in a region having uniform intensity. Moreover, one must also have some control on the variance (effective width) of $\phi(s)$. We now address these design problems in order.

The properties of symmetry, nonnegativity, and monotonicity are simultaneously enjoyed by the family of *raised cosines* of the form

$$\phi(s) = [\cos(\gamma s)]^N \qquad (-T \leq s \leq T).$$

Writing $\cos\theta = (e^{j\theta} + e^{-j\theta})/2$ and applying the binomial theorem, we see that

$$\phi(s) = \sum_{n=0}^{N} 2^{-N}\binom{N}{n}\exp\left(j(2n-N)\gamma s\right).$$

This expresses the raised cosines, as in (5), although we have used a slightly different summation. Since $\phi(s)$ has a total of $(N+1)$ terms, this gives a total of $2(N+1)$ auxiliary images in (6). The central term $n = N/2$ is constant when N is even, and we have one less auxiliary image to process in this case.

D. Approximation of Gaussian Kernels

Fig. 1 shows the raised cosines of degree $N = 1$ to $N = 5$. It is shown that $\phi(s)$ becomes more Gaussian-like over the half-period $[-\pi, \pi]$ with the increase in N. The fact, however, is that $\phi(s)$ converges pointwise to zero at all points as N gets large, except for node points $0, \pm\pi, \pm 2\pi, \ldots$. This problem can be nevertheless addressed by suitably scaling the raised cosine. The precise result is given by the following pointwise convergence:

$$\lim_{N\longrightarrow\infty}\left[\cos\left(\frac{\gamma s}{\sqrt{N}}\right)\right]^N = \exp\left(-\frac{\gamma^2 s^2}{2}\right). \tag{7}$$

Proof: Note that Taylor's theorem with remainder tells us that if $f(x)$ is sufficiently smooth, then $f(x) = \sum_{k=0}^{m-1} x^k f^{(k)}(0)/k! + x^m f^{(m)}(\theta)/m!$, where θ is some number between 0 and x. Applied to the cosine function, we have $\cos(x) = 1 - x^2/2 + x^4\cos\theta/24$. In other words, $\cos(x) = 1 - x^2/2 + r(x)$, where $|r(x)| \lesssim x^4$ (we write $f(x) \lesssim g(x)$ to signify that $f(x) \leq Cg(x)$ for some absolute constant C, where C is independent of x). Using this estimate, along with the binomial theorem, we can write

$$\left[\cos\left(\frac{\gamma s}{\sqrt{N}}\right)\right]^N = \left(1 - \frac{\gamma^2 s^2}{2N}\right)^N + \sum_{k=1}^{N}\binom{N}{k} r(s,N)^k\left(1 - \frac{\gamma^2 s^2}{2N}\right)^{N-k}$$

where $|r(s,N)| \lesssim s^4/N^2$. We are almost done since it is well known that $(1 + x/N)^N$ approaches $\exp(x)$ as N gets large. To establish (7), all we need to show is that, for any fixed s, the residual terms can be made negligibly small by simply setting N large.

Now note that if $|s| \lesssim N^{1/2}$, then the magnitude of $(1 - \gamma^2 s^2/2N)$ is within unity, and on the other hand, $s^4/N < 1$ when $|s| < N^{1/4}$. Thus, given any fixed s, we set N to be large enough so that s satisfies the above bounds. Then, following the trivial inequality $\binom{N}{k} < N^k$, we see that the modulus of the residual is

$$\lesssim \sum_{k=1}^{N} N^k\left(\frac{s^4}{N^2}\right)^k \lesssim N\left(\frac{s^4}{N}\right)^N \lesssim \frac{1}{N}$$

provided that $|s| < L_N = (N^{1-2/N})^{1/4}$. This can be clearly achieved by increasing N since L_N is monotonic in N. ■

We have seen that raised cosines of sufficiently large order provide arbitrarily close approximations of the Gaussian. The crucial feature about (7) is that the rate of convergence is much faster than that of Taylor polynomials, which were used to approximate the Gaussian range kernel in [2]. In particular, we can obtain an approximation comparable to that achieved using

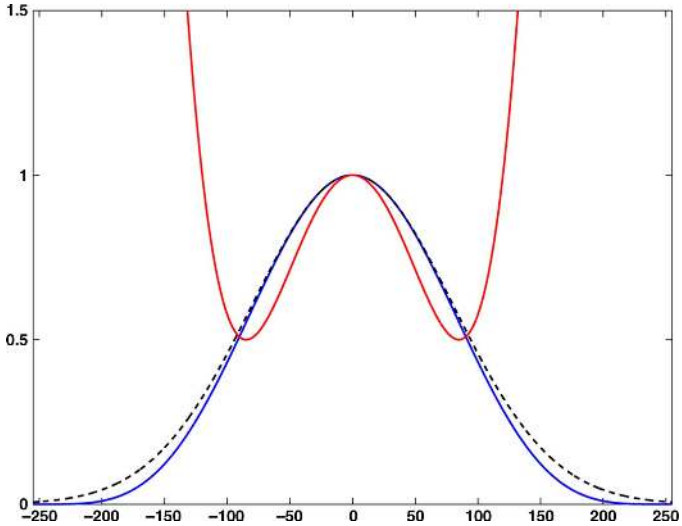

Fig. 2. (Dashed black curve) Approximation of the Gaussian $\exp(-x^2/2\sigma^2)$ over the interval $[-255, 255]$ using (solid red curve) the Taylor polynomial and (solid blue curve) the raised cosine. We set $\sigma = 80$ and use $N = 4$ for the raised cosine in (7). The raised cosine is of the form $a_0 + a_1\cos(2\theta) + a_2\cos(4\theta)$ in this case. We use a three-term Taylor polynomial of the form $b_0 + b_1x^2 + b_2x^4$. It is clear that the raised cosine offers a much better approximation than its polynomial counterpart. In particular, note how the polynomial blows up beyond $|x| > 100$.

polynomials using a fewer number of terms. This is important from the practical standpoint. In Fig. 2, we consider the target Gaussian kernel $\exp(-s^2/2\sigma^2)$, where $\sigma = 80$. We approximate this using the raised cosine of degree 4, which has three terms. We also plot the polynomial corresponding to the three-term Taylor expansion of the Gaussian, which is used for approximating the Gaussian in [2]. It is clear that the approximation quality of the raised cosine is superior to that offered by a Taylor polynomial having equal number of terms. In particular, note that the Taylor approximation does not automatically offer the crucial monotonic property.

E. Control of the Width of Range Kernel

The approximation in (7) also suggests a means of controlling the variance of the raised cosine, namely, by controlling the variance of the target Gaussian. The target Gaussian (with normalization) has a fixed variance of γ^{-2}. This can be increased by simply rescaling the argument of the cosine in (7) by some $\rho > 1$. In particular, for sufficiently large N

$$\left[\cos\left(\frac{\gamma s}{\rho\sqrt{N}}\right)\right]^N \approx \exp\left(-\frac{s^2}{2\rho^2\gamma^{-2}}\right). \tag{8}$$

The variance of the target Gaussian (again with normalization) has now increased to $\rho^2\gamma^{-2}$. A fairly accurate estimate of the variance of the raised cosine is therefore $\sigma^2 \approx \rho^2\gamma^{-2}$. In particular, we can increase the variance by simply setting $\rho = \gamma\sigma$ for all $\sigma > \gamma^{-2}$, provided that N is large enough.

Bringing down the variance below γ^{-2}, on the other hand, is more subtle. This cannot be achieved by simply rescaling with $\rho < 1$ on account of the oscillatory nature of the cosine. For instance, setting $\rho < 1$ can cause $\phi(s)$ to become nonnegative or lose its monotonicity. The only way of doing so is by increasing the degree of the cosine (cf. Fig. 1). In particular, N must be large enough so that the argument of $\cos(\cdot)$ is within $[-\pi/2, \pi/2]$ for all $-T \le s \le T$. This is the case if

$$N \ge \rho^{-2} \approx (\gamma\sigma)^{-2}.$$

In other words, to approximate a Gaussian having small variance σ, N must be at least as large as $N_0 \approx (\gamma\sigma)^{-2}$. The bound is quite tight for large σ but is loose when σ is small. We empirically determined N_0 for certain values of σ for the case when $T = 255$, some of which are given in Table I. It turned out to be much lower than the estimate $(\gamma\sigma)^{-2}$ when σ is small. For a fixed setting of T (e.g., for grayscale images), this suggests the use of a lookup table for determining N_0 for small-σ on the fly.

The above analysis leads us to an $O(1)$ algorithm for approximating the Gaussian bilateral filtering, where both the spatial and range filters are Gaussians. The steps are summarized in Algorithm 1.

TABLE I
N_0 IS THE MINIMUM DEGREE OF THE RAISED COSINE REQUIRED TO APPROXIMATE A GAUSSIAN OF STANDARD DEVIATION σ ON INTERVAL $[-255, 255]$. THE ESTIMATE $\lceil(\gamma\sigma)^{-2}\rceil$ IS ALSO SHOWN

σ	200	150	100	80	60	50	40
N_0	1	2	3	4	5	7	9
$\lceil(\gamma\sigma)^{-2}\rceil$	1	2	3	5	8	11	17

TABLE II
TIME IN MILLISECONDS REQUIRED FOR PROCESSING A GRAYSCALE IMAGE OF SIZE 720 × 540 PIXELS USING OUR ALGORITHM. THE PROCESSING WAS DONE ON A MAC OS X 2 × QUAD CORE 2.6-GHZ MACHINE USING MULTITHREADING

$\sigma_r \rightarrow$	10	20	30	40	50	60	70	80	90	100
$\sigma_s = 10$	3604	452	195	120	74	61	49	34	32	27
$\sigma_s = 100$	3755	482	217	127	89	69	54	43	37	28

Algorithm 1 Fast $O(1)$ bilateral filtering for the Gaussian kernel

Input: Image $f(\boldsymbol{x})$, dynamic range $[-T, T]$, σ_s^2, and σ_r^2 for the spatial and range filters.

1. Set $\gamma = \pi/2T$ and $\rho = \gamma\sigma_r$.

2. If $\sigma_r > \gamma^{-2}$, pick any large N. Else, set $N = (\gamma\sigma_r)^{-2}$, or use a look-up table to fix N.

3. For $0 \le n \le N$, set up images $h_n(\boldsymbol{x}) = \exp(j\gamma(2n-N)f(\boldsymbol{x})/\rho\sqrt{N})$ and $g_n(\boldsymbol{x}) = f(\boldsymbol{x})h_n(\boldsymbol{x})$ and coefficients $d_n(\boldsymbol{x}) = 2^{-N}\binom{N}{n}\exp(-j\gamma(2n-N)f(\boldsymbol{x})/\rho\sqrt{N})$.

4. Use an $O(1)$ algorithm to filter $h_n(\boldsymbol{x})$ and $g_n(\boldsymbol{x})$ with a Gaussian of variance σ_s^2 to get $\overline{h_n}(\boldsymbol{x})$ and $\overline{g_n}(\boldsymbol{x})$.

5. Set $\tilde{f}(\boldsymbol{x})$ as the ratio of $\sum_{n=0}^{N} d_n(\boldsymbol{x})\overline{g_n}(\boldsymbol{x})$ and $\sum_{n=0}^{N} d_n(\boldsymbol{x})\overline{h_n}(\boldsymbol{x})$.

Return: Filtered image $\tilde{f}(\boldsymbol{x})$.

III. EXPERIMENTS

We implemented the proposed algorithm for Gaussian bilateral filtering in Java on a Mac Operating System (OS) X 2× Quad core 2.66-GHz machine as an ImageJ plug-in. We used multithreading for computing the spatial averages of the auxiliary images in parallel. A recursive $O(1)$ algorithm was used for

Fig. 3. Comparison of various implementations of the Gaussian bilateral filter on the grayscale image *Isha* of size 600 × 512. The filter settings are $\sigma_s = 15$ and $\sigma_r = 80$. (a) Original image. (b) Direct implementation of the bilateral filter. (c) Output obtained using polynomial kernel [2]. (d) Output of our algorithm. Note the strange artifacts in (c), particularly around the right eye (see zoomed insets). This is on account of the distortion caused by the polynomial approximation shown in Fig. 2. The standard deviation of the error between (b) and (c) is 6.5, whereas that between (b) and (d) is 1.2.

implementing the Gaussian filter in space domain [15]. The average time required for processing a 720 × 540 grayscale image using our algorithm are shown in Table II. We repeated the experiment for different variances of the Gaussian range kernel and at different spatial variances. As shown in the table, the processing time is quite fast, compared with a direct implementation of the bilateral filter, which requires considerable time depending on the size of the spatial filter. For instance, a direct implementation of the filter on a 512 × 512 image required 4 s for σ_s as low as 3 on our machine (using discretized Gaussians supported on $[-3\sigma, 3\sigma]^2$), and this climbed up to almost 10 s for $\sigma_s = 10$. As shown in Table II, the processing time of our algorithm, however, suddenly shoots up for narrow Gaussians with $\sigma_r < 15$. This is due to the large N required to approximate the Gaussian in this regime (cf. Table I). We have figured out an approximation scheme for further accelerating the processing for very small σ_r, without appreciably degrading the final output. Discussion of this method is, however, beyond the present scope of this paper.

We next tried a visual comparison of the output of our algorithm with the algorithm in [2]. In Fig. 3, we compare the outputs of the two algorithms with the direct implementation, on a natural grayscale image. As is clearly shown in the processed images, our result very closely resembles the exact output. The result obtained using the polynomial kernel, on the other hand, shows strange artifacts. The difference is also clear from the standard deviation of the error between the exact output and the approximations. We note, however, that the execution time of the polynomial method is slightly lower than that of our method since it requires half the number of auxiliary images for a given degree.

We also tested our implementation of the Gaussian bilateral filter on color (red–green–blue) images. We tried a naive processing, where each of the three color channels were independently processed. The results on a couple of images are shown in Fig. 4. The Java source code can be downloaded from the web at http://bigwww.epfl.ch/algorithms/bilateral-filter.

IV. Discussion

We have presented a general method of computing the bilateral filter in constant time using trigonometric range kernels. Within this framework, we have shown how feasible range kernels could be realized using the family of raised cosines. The highlights of our approach are the following.

Accuracy. Our method is exact, at least for the family of raised cosines. It does not require the quantization of the range kernel, as is the case in [4] and [11]. Moreover, note that the auxiliary images in (6) have the same dynamic range as the input image irrespective of the degree N. This is unlike the situation in [2], where the dynamic range of the auxiliary images exponentially grow with N. This makes the computations susceptible to numerical errors for large N.

Fig. 4. Results on the color images *Greekdome* and *Tulip*, using our implementation of the Gaussian bilateral filter. The original image is on the left, and the processed image is on the right. In either case, the red, green, and blue channels were independently processed. We used $\sigma_s = 10$ and $\sigma_r = 20$ for *Greekdome*, and $\sigma_s = 20$ and $\sigma_r = 60$ for *Tulip*. (Images courtesy of S. Paris and F. Durand).

Speed. Besides having $O(1)$ complexity, our algorithm can be also implemented in parallel. This allows us to further accelerate its speed.

Approximation Property. Trigonometric functions yield better (local) approximation of Gaussians than polynomials. In particular, we showed that by using a particular class of raised cosines, we can obtain much better approximations of the Gaussian range kernel than that offered by the Taylor polynomials in [2]. The final output is artifact-free and very closely resembles the true output. The only flip side of our approach (this is also the case with [2], as noted in [4]) is that a large number of terms are required to approximate very narrow Gaussians over large intervals.

Space-Variant Extension. The spatial kernel in (1) can be changed from point to point within the image to control the amount of smoothing (particularly in homogenous regions), while the range kernel is kept fixed. Due to (6), this can be done by simply computing the space-variant averages of each auxiliary image. The good news is that this can be also realized for an $M \times M$ image at the cost of $O(M^2)$ operations, using particular spatial kernels. This includes the 2-D box and hat filter [16], [17] and the more general class of Gaussian-like box splines in [19].

ACKNOWLEDGMENT

The authors would like to thank A. Bhandari for his help with the insets in Fig. 3 and also S. Sanyal for providing the image used in the same figure.

REFERENCES

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

[2] F. Porikli, "Constant time $O(1)$ bilateral filtering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.

[3] A. Buades, B. Coll, and J. Morel, "A review of image denoising algorithms, with a new one," *Multiscale Model. Simul.*, vol. 4, no. 2, pp. 490–530, 2005.

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

[5] 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.

[6] E. Bennett, J. Mason, and L. McMillan, "Multispectral bilateral video fusion," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1185–1194, May 2007.

[7] H. Winnemöller, S. C. Olsen, and B. Gooch, "Real-time video abstraction," in *Proc. ACM SIGGRAPH*, 2006, pp. 1221–1226.

[8] R. Ramanath and W. E. Snyder, "Adaptive demosaicking," *J. Electron. Imag.*, vol. 12, no. 4, pp. 633–642, 2003.

[9] 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.*, 2006, pp. 211–224.

[10] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister, "Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 3, pp. 492–504, Mar. 2009.
[11] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," in *Proc. ACM SIGGRAPH*, 2002, vol. 21, pp. 257–266.
[12] B. Weiss, "Fast median and bilateral filtering," in *Proc. ACM SIGGRAPH*, 2006, vol. 25, pp. 519–526.
[13] T. Pham and L. van Vliet, "Separable bilateral filtering for fast video preprocessing," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2005, pp. 1–4.
[14] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," in *Proc. Eur. Conf. Comput. Vis.*, 2006, pp. 568–580.
[15] I. Young, J. Gerbrands, and L. van Vliet, *Fundamentals of Image Processing*. Delft, The Netherlands: PH Publications, 1995.
[16] P. Heckbert, "Filtering by repeated integration," *ACM SIGGRAPH Comput. Graph.*, vol. 20, no. 4, pp. 315–321, Aug. 1986.
[17] F. C. Crow, "Summed-area tables for texture mapping," in *Proc. ACM SIGGRAPH*, 1984, vol. 18, pp. 207–212.
[18] I. Young and L. van Vliet, "Recursive implementation of the Gaussian filter," *Signal Process.*, vol. 44, no. 2, pp. 139–151, Jun. 1995.
[19] K. Chaudhury, A. Muñoz-Barrutia, and M. Unser, "Fast space-variant elliptical filtering using box splines," *IEEE Trans. Image Process.*, vol. 19, no. 9, pp. 2290–2306, Sep. 2010.

Kunal Narayan Chaudhury (S'08) received the B.E. degree in electrical engineering from Jadavpur University, Kolkata, India, the M.E. degree in system science and automation from the Indian Institute of Science, Bangalore, India, and the Ph.D. degree from the Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland.

He is currently a Research Fellow in the Department of Applied Mathematics, Princeton University, Princeton, NJ. His research interests broadly include time-frequency analysis and wavelet theory and the application of spline approximations in image processing.

Daniel Sage received the M.S. and Ph.D. degrees in control and signal processing from the Institut National Polytechnique de Grenoble, Grenoble, France, in 1986 and 1989, respectively.

From 1989 to 1998, he was a Consulting Engineer with the Industrial Vision Department, Attexor S.A. During his career, he has developed vision systems oriented to the quality control in the industrial sector. In 1998, he joined the Biomedical Imaging Group, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, as the Head of software development. He is in charge of both the coordination of software development and of setting down the computing infrastructure of the group. He is also involved in the development of bioimaging software and methods for computer-assisted teaching.

Michael Unser (M'89–SM'94–F'99) received the M.S. (*summa cum laude*) and Ph.D. degrees in electrical engineering from the Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland, in 1981 and 1984, respectively.

From 1985 to 1997, he worked as a Scientist with the National Institutes of Health, Bethesda, MD. He is currently a Full Professor and the Director of the Biomedical Imaging Group, EPFL, Lausanne, Switzerland. His main research area is biomedical image processing. He has a strong interest in sampling theories, multiresolution algorithms, wavelets, and the use of splines for image processing. He has published over 150 journal papers on those topics and is one of the Institute for Scientific Information Highly Cited Authors in Engineering (http://isihighlycited.com).

Dr. Unser is a Fellow of the European Association for Signal Processing and a member of the Swiss Academy of Engineering Sciences. He coorganized the first IEEE International Symposium on Biomedical Imaging in 2002 and was the Founding Chair of the technical committee of the IEEE Signal Processing Society on bioimaging and signal processing. He is currently a member of the editorial boards of the Foundations and Trends in Signal Processing and the Society for Industrial and Applied Mathematics *Journals on Imaging Sciences* and *Sampling Theory in Signal and Image Processing*. He has held the position of an Associate Editor-in-Chief of the IEEE TRANSACTIONS ON MEDICAL IMAGING from 2003 to 2005 and has served as an Associate Editor of the same journal from 1999 to 2002 and from 2006 to 2007, the IEEE TRANSACTIONS ON IMAGE PROCESSING from 1992 to 1995, and the IEEE SIGNAL PROCESSING LETTERS from 1994 to 1998. He was the recipient of the 1995 and 2003 Best Paper Awards, the 2000 Magazine Award, and the 2008 Technical Achievement Award from the IEEE Signal Processing Society.