

A Fast Approximation of the Bilateral Filter using a Signal Processing Approach

Sylvain Paris and Frédo Durand

Massachusetts Institute of Technology
Computer Science and Artificial Intelligence Laboratory

Abstract. The bilateral filter is a nonlinear filter that smoothes a signal while preserving strong edges. It has demonstrated great effectiveness for a variety of problems in computer vision and computer graphics, and a fast version has been proposed. Unfortunately, little is known about the accuracy of such acceleration. In this paper, we propose a new signal-processing analysis of the bilateral filter, which complements the recent studies that analyzed it as a PDE or as a robust statistics estimator. Importantly, this signal-processing perspective allows us to develop a novel bilateral filtering acceleration using a downsampling in space and intensity. This affords a principled expression of the accuracy in terms of bandwidth and sampling. The key to our analysis is to express the filter in a higher-dimensional space where the signal intensity is added to the original domain dimensions. The bilateral filter can then be expressed as simple linear convolutions in this augmented space followed by two simple nonlinearities. This allows us to derive simple criteria for downsampling the key operations and to achieve important acceleration of the bilateral filter. We show that, for the same running time, our method is significantly more accurate than previous acceleration techniques.

1 Introduction

The bilateral filter is a nonlinear filter proposed by Tomasi and Manduchi to smooth images [1]. It has been adopted for several applications such as texture removal [2], dynamic range compression [3], and photograph enhancement [4, 5]. It has also been adapted to other domains such as mesh fairing [6, 7], volumetric denoising [8] and exposure correction of videos [9]. This large success stems from several origins. First, its formulation and implementation are simple: a pixel is simply replaced by a weighted mean of its neighbors. And it is easy to adapt to a given context as long as a distance can be computed between two pixel values (*e.g.* distance between hair orientations in [10]). The bilateral filter is also non-iterative, *i.e.* it achieves satisfying results with only a single pass. This makes the filter's parameters relatively intuitive since their action does not depend on the cumulated effects of several iterations.

On the other hand, the bilateral filter is nonlinear and its evaluation is computationally expensive since traditional acceleration, such as performing convolution after an FFT, are not applicable. Elad [11] proposes an acceleration method using Gauss-Seidel iterations, but it only applies when multiple iterations of the

filter are required. Durand and Dorsey [3] describe a linearized version of the filter that achieves dramatic speed-ups by downsampling the data, achieving running times under one second. Unfortunately, this technique is not grounded on firm theoretical foundations, and it is difficult to evaluate the accuracy that is sacrificed. In this paper, we build on this work but we interpret the bilateral filter in terms of signal processing in a higher-dimensional space. This allows us to derive an improved acceleration scheme that yields equivalent running times but dramatically improves numerical accuracy.

Contributions This paper introduces the following contributions:

- An interpretation of the bilateral filter in a signal processing framework. Using a higher dimensional space, we formulate the bilateral filter as a convolution followed by simple nonlinearities.
- Using this higher dimensional space, we demonstrate that the convolution computation can be downsampled without significantly impacting the result accuracy. This approximation technique enables a speed-up of several orders of magnitude while controlling the error induced.

2 Related Work

The bilateral filter was first introduced by Smith and Brady under the name “SUSAN” [12]. It was rediscovered later by Tomasi and Manduchi [1] who called it the “bilateral filter” which is now the most commonly used name. The filter replaces each pixel by a weighted average of its neighbors. The weight assigned to each neighbor decreases with both the distance in the image plane (the *spatial domain* \mathcal{S}) and the distance on the intensity axis (the *range domain* \mathcal{R}). Using a Gaussian G_σ as a decreasing function, and considering a grey-level image I , the result I^{bf} of the bilateral filter is defined by:

$$I_{\mathbf{p}}^{\text{bf}} = \frac{1}{W_{\mathbf{p}}^{\text{bf}}} \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|) I_{\mathbf{q}} \quad (1a)$$

$$\text{with } W_{\mathbf{p}}^{\text{bf}} = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|) \quad (1b)$$

The parameter σ_s defines the size of the spatial neighborhood used to filter a pixel, and σ_r controls how much an adjacent pixel is downweighted because of the intensity difference. W^{bf} normalizes the sum of the weights.

Barash [13] shows that the two weight functions are actually equivalent to a single weight function based on a distance defined on $\mathcal{S} \times \mathcal{R}$. Using this approach, he relates the bilateral filter to adaptive smoothing. Our work follows a similar idea and also uses $\mathcal{S} \times \mathcal{R}$ to describe bilateral filtering. Our formulation is nonetheless significantly different because we not only use the higher-dimensional space for the definition of a distance, but we also use convolution in this space. Elad [11] demonstrates that the bilateral filter is similar to the Jacobi algorithm, with the specificity that it accounts for a larger neighborhood instead of the closest adjacent pixels usually considered. Buades *et al.* [14] expose an asymptotic analysis of the Yaroslavsky filter [15] which is a special case of the bilateral filter with a step function as spatial weight. They prove that asymptotically, the

Yaroslavsky filter behaves as the Perona-Malik filter, *i.e.* it alternates between smoothing and shock formation depending on the gradient intensity. Durand and Dorsey [3] cast their study into the robust statistics framework [16, 17]. They show that the bilateral filter is a w -estimator [17] (p.116). This explains the role of the range weight in terms of sensitivity to outliers. They also point out that the bilateral filter can be seen as an extension of the Perona-Malik filter using a larger neighborhood. Mrázek *et al.* [18] relate bilateral filtering to a large family of nonlinear filters. From a single equation, they express filters such as anisotropic diffusion and statistical estimators by varying the neighborhood size and the involved functions. The main difference between our study and existing work is that the previous approaches link bilateral filtering to another nonlinear filter based on PDEs or statistics whereas we cast our study into a signal processing framework. We demonstrate that the bilateral filter can be mainly computed with linear operations, the nonlinearities being grouped in a final step.

Several articles [11, 14, 19] improve the bilateral filter. They share the same idea: By exploiting the local “slope” of the image intensity, it is possible to better represent the local shape of the signal. Thus, they define a modified filter that better preserve the image characteristics *e.g.* they avoid the formation of shocks. We have not explored this direction since the formulation becomes significantly more complex. It is however an interesting avenue for future work.

The work most related to ours are the speed-up techniques proposed by Elad [11] and Durand and Dorsey [3]. Elad [11] uses Gauss-Seidel iterations to accelerate the convergence of iterative filtering. Unfortunately, no results are shown – and this technique is only useful when the filter is iterated to reach the stable point, which is not its standard use of the bilateral filter (one iteration or only a few). Durand and Dorsey [3] linearize the bilateral filter and propose a downsampling scheme to accelerate the computation down to few seconds or less. However, no theoretical study is proposed, and the accuracy of the approximation is unclear. In comparison, we base our technique on signal processing grounds which help us to define a new and meaningful numerical scheme. Our algorithm performs low-pass filtering in a higher-dimensional space than Durand and Dorsey’s [3]. The cost of a higher-dimensional convolution is offset by the accuracy gain, which yields better performance for the same accuracy.

3 Signal Processing Approach

We decompose the bilateral filter into a convolution followed by two nonlinearities. To cast the filter as a convolution, we define a homogeneous intensity that will allow us to obtain the normalization term $W_{\mathbf{p}}^{\text{bf}}$ as an homogeneous component after convolution. We also need to perform this convolution in the product space of the domain and the range of the input signal. Observing Equations (1), the nonlinearity comes from the division by W^{bf} and from the dependency on the pixel intensities through $G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)$. We study each point separately and isolate them in the computation flow.

3.1 Homogeneous Intensity

A direct solution to handle the division is to multiply both sides of Equation (1a) by $W_{\mathbf{p}}^{\text{bf}}$. The two equations are then almost similar. We underline this point by rewriting Equations (1) using two-dimensional vectors:

$$\begin{pmatrix} W_{\mathbf{p}}^{\text{bf}} I_{\mathbf{p}}^{\text{bf}} \\ W_{\mathbf{p}}^{\text{bf}} \end{pmatrix} = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|) \begin{pmatrix} I_{\mathbf{q}} \\ 1 \end{pmatrix} \quad (2)$$

To maintain the property that the bilateral filter is a weighted mean, we introduce a function W whose value is 1 everywhere:

$$\begin{pmatrix} W_{\mathbf{p}}^{\text{bf}} I_{\mathbf{p}}^{\text{bf}} \\ W_{\mathbf{p}}^{\text{bf}} \end{pmatrix} = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|) \begin{pmatrix} W_{\mathbf{q}} I_{\mathbf{q}} \\ W_{\mathbf{q}} \end{pmatrix} \quad (3)$$

By assigning a couple $(W_{\mathbf{q}} I_{\mathbf{q}}, W_{\mathbf{q}})$ to each pixel q , we express the filtered pixels as linear combinations of their adjacent pixels. Of course, we have not “removed” the division since to access the actual value of the intensity, the first coordinate (WI) has still to be divided by the second one (W). This can be compared with homogeneous coordinates used in projective geometry. Adding an extra coordinate to our data makes most of the computation pipeline computable with linear operations; a division is made only at the final stage. Inspired by this parallel, we call the couple (WI, W) the *homogeneous intensity*.

Although Equation (3) is a linear combination, this does not define a linear filter yet since the weights depend on the actual values of the pixels. The next section addresses this issue.

3.2 The Bilateral Filter as a Convolution

If we ignore the term $G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)$, Equation (3) is a classical convolution by a Gaussian kernel: $(W^{\text{bf}} I^{\text{bf}}, W^{\text{bf}}) = G_{\sigma_s} \otimes (WI, W)$. But the range weight depends on $I_{\mathbf{p}} - I_{\mathbf{q}}$ and there is no summation on I . To overcome this point, we introduce an additional dimension ζ and sum over it. With the Kronecker symbol $\delta(\zeta)$ (1 if $\zeta = 0$, 0 otherwise) and \mathcal{R} the interval on which the intensity is defined, we rewrite Equation (3) using $[\delta(\zeta - I_{\mathbf{q}}) = 1] \Leftrightarrow [\zeta = I_{\mathbf{q}}]$:

$$\begin{pmatrix} W_{\mathbf{p}}^{\text{bf}} I_{\mathbf{p}}^{\text{bf}} \\ W_{\mathbf{p}}^{\text{bf}} \end{pmatrix} = \sum_{\mathbf{q} \in \mathcal{S}} \sum_{\zeta \in \mathcal{R}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(|I_{\mathbf{p}} - \zeta|) \delta(\zeta - I_{\mathbf{q}}) \begin{pmatrix} W_{\mathbf{q}} I_{\mathbf{q}} \\ W_{\mathbf{q}} \end{pmatrix} \quad (4)$$

Equation (4) is a sum over the product space $\mathcal{S} \times \mathcal{R}$. We now focus on this space. We use lowercase names for the functions defined on $\mathcal{S} \times \mathcal{R}$. The product $G_{\sigma_s} G_{\sigma_r}$ defines a Gaussian kernel g_{σ_s, σ_r} on $\mathcal{S} \times \mathcal{R}$:

$$g_{\sigma_s, \sigma_r} : (\mathbf{x} \in \mathcal{S}, \zeta \in \mathcal{R}) \mapsto G_{\sigma_s}(\|\mathbf{x}\|) G_{\sigma_r}(|\zeta|) \quad (5)$$

From the remaining part of Equation (4), we build two functions i and w :

$$i : (\mathbf{x} \in \mathcal{S}, \zeta \in \mathcal{R}) \mapsto I_{\mathbf{x}} \quad (6a)$$

$$w : (\mathbf{x} \in \mathcal{S}, \zeta \in \mathcal{R}) \mapsto \delta(\zeta - I_{\mathbf{x}}) W_{\mathbf{x}} \quad (6b)$$

The following relations stem directly from the two previous definitions:

$$I_{\mathbf{x}} = i(\mathbf{x}, I_{\mathbf{x}}) \quad \text{and} \quad W_{\mathbf{x}} = w(\mathbf{x}, I_{\mathbf{x}}) \quad \text{and} \quad \forall \zeta \neq I_{\mathbf{x}}, w(\mathbf{x}, \zeta) = 0 \quad (7)$$

Then Equation (4) is rewritten as:

$$\begin{pmatrix} W_{\mathbf{p}}^{\text{bf}} & I_{\mathbf{p}}^{\text{bf}} \\ W_{\mathbf{p}}^{\text{bf}} & \end{pmatrix} = \sum_{(\mathbf{q}, \zeta) \in \mathcal{S} \times \mathcal{R}} g_{\sigma_s, \sigma_r}(\mathbf{p} - \mathbf{q}, I_p - \zeta) \begin{pmatrix} w(\mathbf{q}, \zeta) & i(\mathbf{q}, \zeta) \\ w(\mathbf{q}, \zeta) & \end{pmatrix} \quad (8)$$

The above formula corresponds to the value at point $(\mathbf{p}, I_{\mathbf{p}})$ of a convolution between g_{σ_s, σ_r} and the two-dimensional function (wi, w) :

$$\begin{pmatrix} W_{\mathbf{p}}^{\text{bf}} & I_{\mathbf{p}}^{\text{bf}} \\ W_{\mathbf{p}}^{\text{bf}} & \end{pmatrix} = \left[g_{\sigma_s, \sigma_r} \otimes \begin{pmatrix} wi \\ w \end{pmatrix} \right] (\mathbf{p}, I_{\mathbf{p}}) \quad (9)$$

According to the above equation, we introduce the functions i^{bf} and w^{bf} :

$$(w^{\text{bf}} i^{\text{bf}}, w^{\text{bf}}) = g_{\sigma_s, \sigma_r} \otimes (wi, w) \quad (10)$$

Thus, we have reached our goal. The bilateral filter is expressed as a convolution followed by nonlinear operations:

linear:	$(w^{\text{bf}} i^{\text{bf}}, w^{\text{bf}}) = g_{\sigma_s, \sigma_r} \otimes (wi, w) \quad (11a)$
nonlinear:	$I_{\mathbf{p}}^{\text{bf}} = \frac{w^{\text{bf}}(\mathbf{p}, I_{\mathbf{p}}) i^{\text{bf}}(\mathbf{p}, I_{\mathbf{p}})}{w^{\text{bf}}(\mathbf{p}, I_{\mathbf{p}})} \quad (11b)$

The nonlinear section is actually composed of two operations. The functions $w^{\text{bf}} i^{\text{bf}}$ and w^{bf} are evaluated at point $(\mathbf{p}, I_{\mathbf{p}})$. We name this operation *slicing*. The second nonlinear operation is the division. In our case, slicing and division commute *i.e.* the result is independent of their order because g_{σ_s, σ_r} is positive and w values are 0 and 1, which ensures that w^{bf} is positive.

3.3 Intuition

To gain more intuition about our formulation of the bilateral filter, we propose an informal description of the process before discussing further its consequences.

The spatial domain \mathcal{S} is a classical xy image plane and the range domain \mathcal{R} is a simple axis labelled ζ . The w function can be interpreted as the “plot in the $xy\zeta$ space of $\zeta = I(x, y)$ ” *i.e.* w is null everywhere except on the points $(x, y, I(x, y))$ where it is equal to 1. The wi product is similar to w . Instead of using binary values 0 or 1 to “plot I ”, we use 0 or $I(x, y)$ *i.e.* it is a plot with a pen whose brightness equals the plotted value.

Then using these two functions wi and w , the bilateral filter is computed as follows. First, we “blur” wi and w *i.e.* we convolve wi and w with a Gaussian defined on $xy\zeta$. This results in the functions $w^{\text{bf}} i^{\text{bf}}$ and w^{bf} . For each point

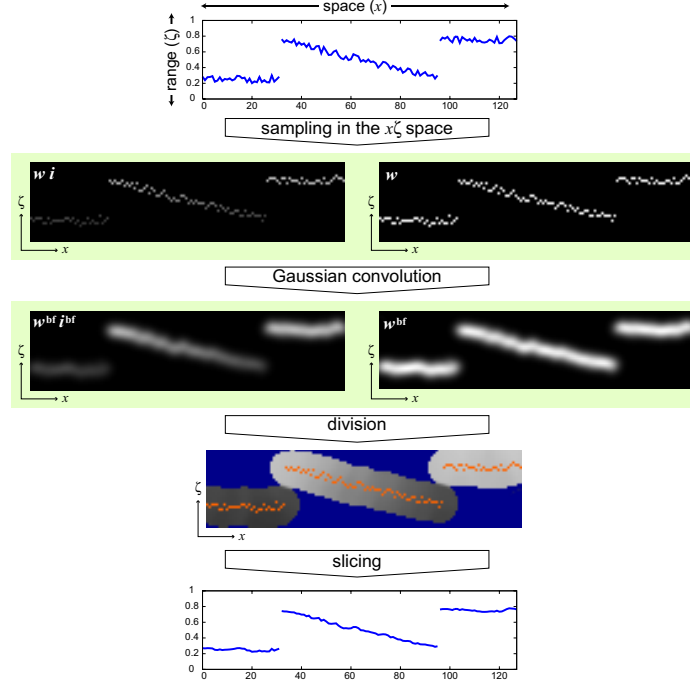


Fig. 1. Our computation pipeline applied to a 1D signal. The original data (top row) are represented by a two-dimensional function (wi, w) (second row). This function is convolved with a Gaussian kernel to form $(w^{\text{bf}} i^{\text{bf}}, w^{\text{bf}})$ (third row). The first component is then divided by the second (fourth row, blue area is undefined because of numerical limitation, $w^{\text{bf}} \approx 0$). Then the final result (last row) is extracted by sampling the former result at the location of the original data (shown in red on the fourth row).

of the $xy\zeta$ space, we compute $i^{\text{bf}}(x, y, \zeta)$ by dividing $w^{\text{bf}}(x, y, \zeta) i^{\text{bf}}(x, y, \zeta)$ by $w^{\text{bf}}(x, y, \zeta)$. The final step is to get the value of the pixel (x, y) of the filtered image I^{bf} . This directly corresponds to the value of i^{bf} at $(x, y, I(x, y))$ which is the point where the input image I was “plotted”. Figure 1 illustrates this process on a simple 1D image.

4 Fast Approximation

We have shown that the bilateral filter can be interpreted as a Gaussian filter in a product space. Our acceleration scheme directly follows from the fact that this operation is a low-pass filter. $(w^{\text{bf}} i^{\text{bf}}, w^{\text{bf}})$ is therefore essentially a band-limited function which is well approximated by its low frequencies.

Using the sampling theorem [20] (p.35), it is sufficient to sample with a rate at least twice shorter than the smallest wavelength considered. In practice, we downsample (wi, w) , perform the convolution, and upsample the result:

$$(w_{\downarrow} i_{\downarrow}, w_{\downarrow}) = \text{downsample}(wi, w) \quad (12a)$$

$$(w_{\downarrow}^{\text{bf}} i_{\downarrow}^{\text{bf}}, w_{\downarrow}^{\text{bf}}) = g_{\sigma_s, \sigma_r} \otimes (w_{\downarrow} i_{\downarrow}, w_{\downarrow}) \quad (12b)$$

$$(w_{\downarrow\uparrow}^{\text{bf}} i_{\downarrow\uparrow}^{\text{bf}}, w_{\downarrow\uparrow}^{\text{bf}}) = \text{upsample}(w_{\downarrow}^{\text{bf}} i_{\downarrow}^{\text{bf}}, w_{\downarrow}^{\text{bf}}) \quad (12c)$$

The rest of the computation remains the same except that we slice and divide $(w_{\downarrow\uparrow}^{\text{bf}}, i_{\downarrow\uparrow}^{\text{bf}}, w_{\downarrow\uparrow}^{\text{bf}})$ instead of $(w^{\text{bf}}, i^{\text{bf}}, w^{\text{bf}})$, using the same $(\mathbf{p}, I_{\mathbf{p}})$ points. Since slicing occurs at points where $w = 1$, it guarantees $w^{\text{bf}} \geq g_{\sigma_s, \sigma_r}(0)$, which ensures that we do not divide by small numbers that would degrade our approximation.

We use box-filtering for the prefilter of the downsampling (*a.k.a.* average downsampling), and linear upsampling. While these filters do not have perfect frequency responses, they offer much better performances than schemes such as tri-cubic filters.

4.1 Evaluation

To evaluate the error induced by our approximation, we compare the result $I_{\downarrow\uparrow}^{\text{bf}}$ from the fast algorithm to I^{bf} obtained from Equations (1). For this purpose, we compute the peak signal-to-noise ratio (PSNR) considering $\mathcal{R} = [0; 1]$:
$$\text{PSNR}(I_{\downarrow\uparrow}^{\text{bf}}) = -10 \log_{10} \left(\frac{1}{|\mathcal{S}|} \sum_{\mathbf{p} \in \mathcal{S}} \left| I_{\downarrow\uparrow}^{\text{bf}}(\mathbf{p}) - I^{\text{bf}}(\mathbf{p}) \right|^2 \right).$$

We have chosen three images as different as possible to cover a broad spectrum of content. We use (see Figure 4):

- An artificial image with various edges and frequencies, and white noise.
- An architectural picture structured along two main directions.
- And a photograph of a natural scene with a more stochastic structure.

The box downsampling and linear upsampling schemes yield very satisfying results while being computationally efficient. We experimented several sampling rates (s_s, s_r) for $\mathcal{S} \times \mathcal{R}$. The meaningful quantities to consider are the ratios $\left(\frac{s_s}{\sigma_s}, \frac{s_r}{\sigma_r} \right)$ that indicate how many high frequencies we ignore compared to the bandwidth of the filter we apply. Small ratios correspond to limited approximations and high ratios to more aggressive downsamplings. A consistent approximation is a sampling rate proportional to the Gaussian bandwidth (*i.e.* $\frac{s_s}{\sigma_s} \approx \frac{s_r}{\sigma_r}$) to achieve similar accuracy on the whole $\mathcal{S} \times \mathcal{R}$ domain. The results plotted in Figure 2 show that this remark is globally valid in practice. A closer look at the plots reveals that \mathcal{S} can be slightly more aggressively downsampled than \mathcal{R} . This is probably due to the nonlinearities and the anisotropy of the signal.

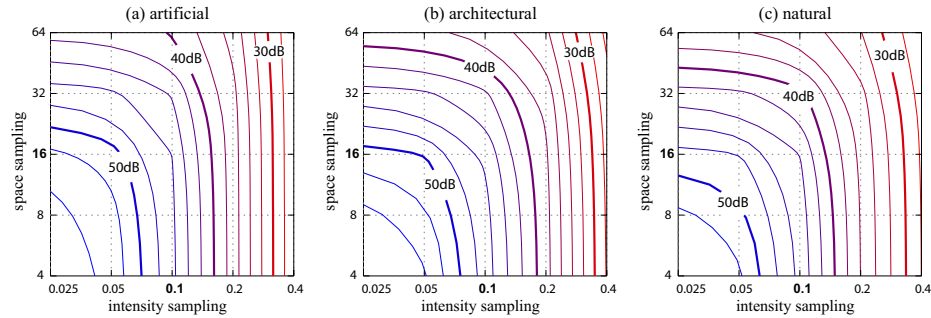


Fig. 2. Accuracy evaluation. All the images are filtered with $(\sigma_s = 16, \sigma_r = 0.1)$. The PSNR in dB is evaluated at various sampling of \mathcal{S} and \mathcal{R} (greater is better). Our approximation scheme is more robust to space downsampling than range downsampling. It is also slightly more accurate on structured scenes (a,b) than stochastic ones (c).

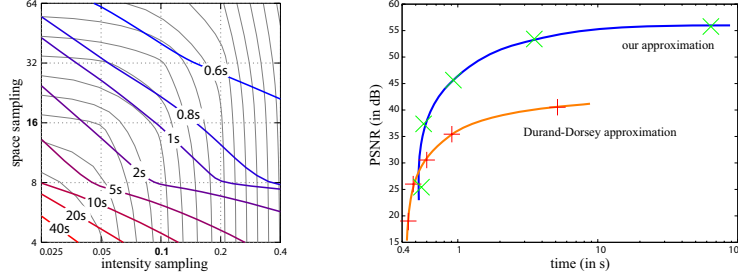


Fig. 3. Left: Running times on the architectural picture with $(\sigma_s = 16, \sigma_r = 0.1)$. The PSNR isolines are plotted in gray. Exact computation takes about 1h. • Right: Accuracy-versus-time comparison. Both methods are tested on the architectural picture (1600×1200) with the same sampling rates of $\mathcal{S} \times \mathcal{R}$ (from left to right): (4;0.025) (8;0.05) (16;0.1) (32;0.2) (64;0.4).

Figure 3-left shows the running times for the architectural picture with the same settings. In theory, the gain from space downsampling should be twice the one from range downsampling since \mathcal{S} is two-dimensional and \mathcal{R} one-dimensional. In practice, the nonlinearities and caching issues induce minor deviations. Combining this plot with the PSNR plot (in gray under the running times) allows for selecting the best sampling parameters for a given error tolerance or a given time budget. As a simple guideline, using sampling steps equal to σ_s and σ_r produce results without visual difference with the exact computation (see Fig. 4). Our scheme achieves a speed-up of two orders of magnitude: Direct computation of Equations (1) lasts about one hour whereas our approximation requires one second. This dramatic improvement opens avenues for interactive applications.

4.2 Comparison with the Durand-Dorsey speed-up

Durand and Dorsey also describe a linear approximation of the bilateral filter [3]. Using evenly spaced intensity values $I_1..I_n$ that cover \mathcal{R} , their scheme can be summarized as (for convenience, we also name G_{σ_s} the 2D Gaussian kernel):

$$\iota_{\downarrow} = \text{downsample}(I) \quad [\text{image downsampling}] \quad (13a)$$

$$\forall \zeta \in \{1..n\} \quad \omega_{\downarrow\zeta}(\mathbf{p}) = G_{\sigma_r}(|\iota_{\downarrow}(\mathbf{p}) - I_{\zeta}|) \quad [\text{range weight evaluation}] \quad (13b)$$

$$\forall \zeta \in \{1..n\} \quad \omega_{\downarrow\zeta}(\mathbf{p}) = \omega_{\downarrow\zeta}(\mathbf{p}) \iota_{\downarrow}(\mathbf{p}) \quad [\text{intensity multiplication}] \quad (13c)$$

$$\forall \zeta \in \{1..n\} \quad (\omega_{\downarrow\zeta}^{\text{bf}}, \omega_{\downarrow\zeta}^{\text{bf}}) = G_{\sigma_s} \otimes (\omega_{\downarrow\zeta}, \omega_{\downarrow\zeta}) \quad [\text{convolution on } \mathcal{S}] \quad (13d)$$

$$\forall \zeta \in \{1..n\} \quad \iota_{\downarrow\zeta}^{\text{bf}} = \omega_{\downarrow\zeta}^{\text{bf}} / \omega_{\downarrow\zeta}^{\text{bf}} \quad [\text{normalization}] \quad (13e)$$

$$\forall \zeta \in \{1..n\} \quad \iota_{\uparrow\zeta}^{\text{bf}} = \text{upsample}(\iota_{\downarrow\zeta}^{\text{bf}}) \quad [\text{layer upsampling}] \quad (13f)$$

$$I^{\text{DD}}(\mathbf{p}) = \text{interpolation}(\iota_{\uparrow\zeta}^{\text{bf}})(\mathbf{p}) \quad [\text{nearest layer interpolation}] \quad (13g)$$

Without downsampling (*i.e.* $\{I_i\} = \mathcal{R}$ and Steps 13a,f ignored), the Durand-Dorsey scheme is equivalent to ours because Steps 13b,c,d correspond to a convolution on $\mathcal{S} \times \mathcal{R}$. Indeed, Step (13b) computes the values of a Gaussian kernel on \mathcal{R} . Step (13c) actually evaluates the convolution on \mathcal{R} , considering that $\iota_{\downarrow}(\mathbf{p})$ is the only nonzero value on the ζ axis. With Step (13d), the convolution on \mathcal{S} , these three steps perform a 3D convolution using a separation between \mathcal{R} and \mathcal{S} .

The differences comes from the downsampling approach. Durand and Dorsey interleave linear and nonlinear operations: The division is done after the convolution 13d but before the upsampling 13f. There is no simple theoretical base to estimate the error. More importantly, the Durand-Dorsey strategy is such that the intensity ι and the weight ω are functions defined on \mathcal{S} only. A given pixel has only one intensity and one weight. After downsampling, both sides of the discontinuity may be represented by the same values of ι and ω . This is a poor representation of the discontinuities since they inherently involve several values. In comparison, we define functions on $\mathcal{S} \times \mathcal{R}$. For a given image point in \mathcal{S} , we can handle several values on the \mathcal{R} domain. The advantage of working in $\mathcal{S} \times \mathcal{R}$ is that this characteristic is not altered by downsampling. It is the major reason why our scheme is more accurate than the Durand-Dorsey technique, especially on discontinuities.

Figure 3-right shows the precision achieved by both approaches relatively to their running time, and Figure 4 illustrates their visual differences. There is no gain in extreme downsampling since nonlinearities are no more negligible. Both approaches also have a plateau in their accuracy *i.e.* beyond a certain point, precision gains increase slowly with sampling refinement but ours reaches a higher accuracy ($\approx 55\text{dB}$ compared to $\approx 40\text{dB}$). In addition, for the same running time, our approach is always more accurate (except for extreme downsampling).

4.3 Implementation

All the experiments have been done on Intel Xeon 2.8GHz using the same code base in C++. We have implemented the Durand-Dorsey technique with the same libraries as our technique. 2D and 3D convolutions are made using FFT. The domains are padded with zeros over 2σ to avoid cross-boundary artefacts. There are no other significant optimizations to avoid bias in the comparisons. A production implementation could therefore be improved with techniques such as separable convolution. Our code is publicly available on our webpage. The software is open-source, under the MIT license.

5 Discussion

Dimensionality Our separation into linear and nonlinear parts comes at the cost of the additional ζ dimension. One has to be careful before increasing the dimensionality of a problem since the incurred performance overhead may exceed the gains, restricting our study to a theoretical discussion. We have however demonstrated that this formalism allows for a computation scheme that is several orders of magnitude faster than a straightforward application of Equation (1). This advocates performing the computation in the $\mathcal{S} \times \mathcal{R}$ space instead of the image plane. In this respect, our approach can be compared to the level sets [21] which also describe a better computation scheme using a higher dimension space.

Note that using the two-dimensional homogeneous intensity does not increase the dimensionality since Equation (1) also computes two functions: W^{bf} and I^{bf} .

Comparison with *Generalized Intensity* Barash also uses points in the $\mathcal{S} \times \mathcal{R}$ space that he names *generalized intensities* [13]. Our two approaches have in common the global meaning of \mathcal{S} and \mathcal{R} : The former is related to pixel positions

sampling (σ_s, σ_r)	(4,0.025)	(8,0.05)	(16,0.1)	(32,0.2)	(64,0.4)
downsampling	1.3s	0.23s	0.09s	0.07s	0.06s
convolution	63s	2.8s	0.38s	0.02s	0.01s
nonlinearity	0.48s	0.47s	0.46s	0.47s	0.46s

Table 1. Time used by each step at different sampling rates of the architectural image. Upsampling is included in the nonlinearity time because our implementation computes $i_{\downarrow\uparrow}^{\text{bf}}$ only at the $(\mathbf{x}, I_{\mathbf{x}})$ points rather than upsampling the whole $\mathcal{S} \times \mathcal{R}$ space.

and the latter to pixel intensities. It is nevertheless important to highlight the differences. Barash handles $\mathcal{S} \times \mathcal{R}$ to compute distances. Thus, he can express the difference between adaptive smoothing and bilateral filtering as a difference of distance definitions. But the actual computation remains the same. Our use of $\mathcal{S} \times \mathcal{R}$ is more involved. We not only manipulate points in this space but also define functions and perform convolutions and slicing. Another difference is our definition of the intensity through a function (i or i^{bf}). Barash associates directly the intensity of a point to its \mathcal{R} component whereas in our framework, the intensity of a point (\mathbf{x}, ζ) is *not* its ζ coordinate *e.g.* in general $i^{\text{bf}}(\mathbf{x}, \zeta) \neq \zeta$. In addition, our approach leads to a more efficient implementation.

Complexity One of the advantage of our separation is that the convolution is the most complex part of the algorithm. Using $|\cdot|$ for the cardinal of a set, the convolution can be done in $\mathcal{O}(|\mathcal{S}| |\mathcal{R}| \log(|\mathcal{S}| |\mathcal{R}|))$ with fast Fourier transform and multiplication in the frequency domain. Then the slicing and the division are done pixel by pixel. Thus they are linear in the image size *i.e.* $\mathcal{O}(|\mathcal{S}|)$. Hence, the algorithm complexity is dominated by the convolution. This result is verified in practice as shown by Table 1. The convolution time rapidly increases as the sampling becomes finer. This validates our choice of focussing on the convolution.

6 Conclusions

We have presented a fast approximation technique of the bilateral filter based on a signal processing interpretation. From a theoretical point of view, we have introduced the notion of homogeneous intensity and demonstrated a new approach of the space-intensity domain. We believe that these concepts can be applied beyond bilateral filtering, and we hope that these contributions will inspire new studies. From a practical point of view, our approximation technique yields results visually similar to the exact computation with interactive running times. This technique paves the way for interactive applications relying on quality image smoothing.

Future Work Our study translates almost directly to higher dimensional data (*e.g.* color images or videos). Analyzing the performance in these cases will provide valuable statistics. Exploring deeper the frequency structure of the $\mathcal{S} \times \mathcal{R}$ domain seems an exciting research direction.

Acknowledgement We thank Soonmin Bae, Samuel Hornus, Thouis Jones, and Sara Su for their help with the paper. This work was supported by a National Science Foundation CAREER award 0447561 “Transient Signal Processing for

Realistic Imagery,” an NSF Grant No. 0429739 “Parametric Analysis and Transfer of Pictorial Style,” a grant from Royal Dutch/Shell Group and the Oxygen consortium. Frédo Durand acknowledges a Microsoft Research New Faculty Fellowship, and Sylvain Paris was partially supported by a Lavoisier Fellowship from the French “Ministère des Affaires Étrangères.”

References

1. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proc. of International Conference on Computer Vision, IEEE (1998) 839–846
2. Oh, B.M., Chen, M., Dorsey, J., Durand, F.: Image-based modeling and photo editing. In: Proc. of SIGGRAPH conference, ACM (2001)
3. Durand, F., Dorsey, J.: Fast bilateral filtering for the display of high-dynamic-range images. ACM Trans. on Graphics **21** (2002) Proc. of SIGGRAPH conference.
4. Eisemann, E., Durand, F.: Flash photography enhancement via intrinsic relighting. ACM Trans. on Graphics **23** (2004) Proc. of SIGGRAPH conference.
5. Petschnigg, G., Agrawala, M., Hoppe, H., Szeliski, R., Cohen, M., Toyama, K.: Digital photography with flash and no-flash image pairs. ACM Trans. on Graphics **23** (2004) Proc. of SIGGRAPH conference.
6. Jones, T.R., Durand, F., Desbrun, M.: Non-iterative, feature-preserving mesh smoothing. ACM Trans. on Graphics **22** (2003) Proc. of SIGGRAPH conference.
7. Fleishman, S., Drori, I., Cohen-Or, D.: Bilateral mesh denoising. ACM Trans. on Graphics **22** (2003) Proc. of SIGGRAPH conference.
8. Wong, W.C.K., Chung, A.C.S., Yu, S.C.H.: Trilateral filtering for biomedical images. In: Proc. of International Symposium on Biomedical Imaging, IEEE (2004)
9. Bennett, E.P., McMillan, L.: Video enhancement using per-pixel virtual exposures. ACM Trans. on Graphics **24** (2005) 845 – 852 Proc. of SIGGRAPH conference.
10. Paris, S., Briceño, H., Sillion, F.: Capture of hair geometry from multiple images. ACM Trans. on Graphics **23** (2004) Proc. of SIGGRAPH conference.
11. Elad, M.: On the bilateral filter and ways to improve it. IEEE Trans. On Image Processing **11** (2002) 1141–1151
12. Smith, S.M., Brady, J.M.: SUSAN – a new approach to low level image processing. International Journal of Computer Vision **23** (1997) 45–78
13. Barash, D.: A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation. IEEE Trans. on Pattern Analysis and Machine Intelligence **24** (2002) 844
14. Buades, A., Coll, B., Morel, J.M.: Neighborhood filters and PDE’s. Technical Report 2005-04, CMLA (2005)
15. Yaroslavsky, L.P.: Digital Picture Processing. Springer Verlag (1985)
16. Huber, P.J.: Robust Statistics. Wiley-Interscience (1981)
17. Hampel, F.R., Ronchetti, E.M., Rousseeuw, P.M., Stahel, W.A.: Robust Statistics – The Approach Based on Influence Functions. Wiley Interscience (1986)
18. Mrázek, P., Weickert, J., Bruhn, A.: On Robust Estimation and Smoothing with Spatial and Tonal Kernels. In: Geometric Properties from Incomplete Data. Springer (to appear)
19. Choudhury, P., Tumblin, J.E.: The trilateral filter for high contrast images and meshes. In: Proc. of Eurographics Symposium on Rendering. (2003)
20. Smith, S.: Digital Signal Processing. Newnes (2002)
21. Osher, S., Sethian, J.A.: Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. J. of Comp. Physics. (1988)

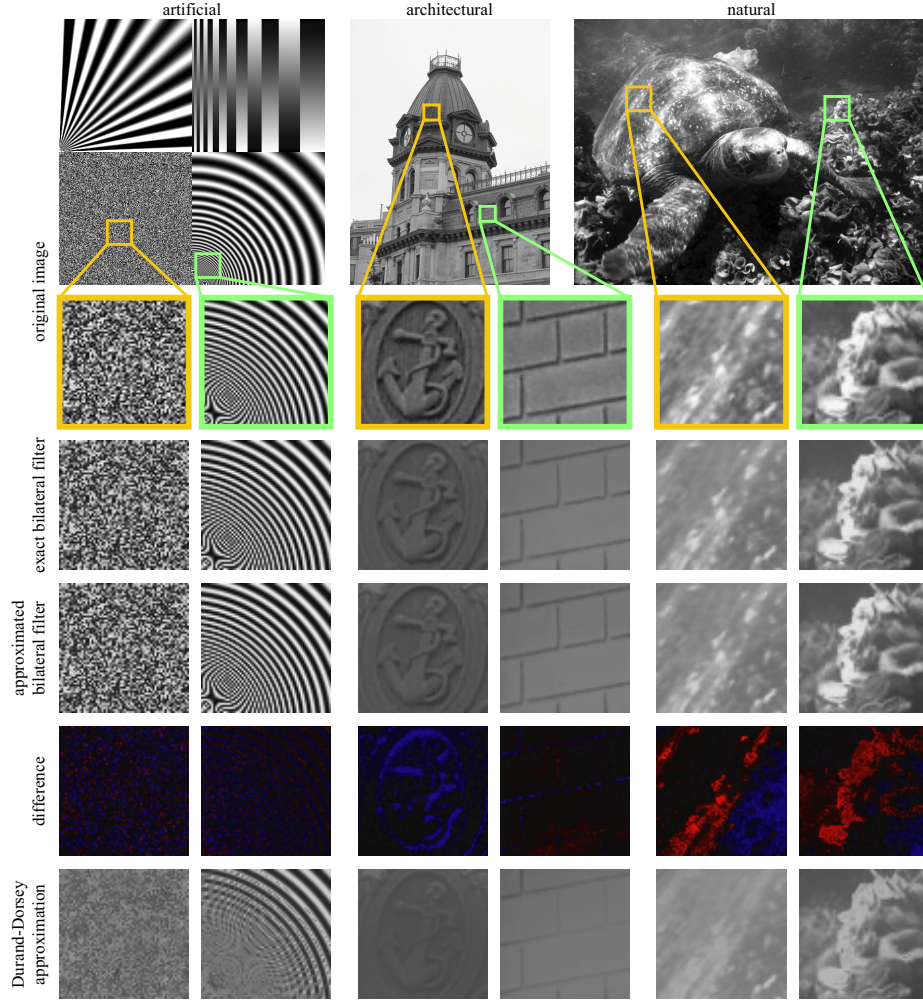


Fig. 4. We have tested our approximated scheme on three images (first row): an artificial image (512×512) with different types of edges and a white noise region, an architectural picture (1600×1200) with strong and oriented features, and a natural photograph (800×600) with more stochastic textures. For clarity, we present representative close-ups (second row). Full resolution images are available on our website. Our approximation produces results (fourth row) visually similar to the exact computation (third row). A color coded subtraction (fifth row) reveals subtle differences at the edges (red: negative, black: 0, and blue: positive). In comparison, the Durand-Dorsey approximation introduces large visual discrepancies: the details are washed out (bottom row). All the filters are computed with $\sigma_s = 16$ and $\sigma_r = 0.1$. Our filter uses a sampling rate of (16,0.1). The sampling rate of the Durand-Dorsey filter is chosen in order to achieve the same (or slightly superior) running time. Thus, the comparison is done fairly, using the same time budget.