Foundations and Trends® in
Computer Graphics and Vision
Vol. 4, No. 1 (2008) 1–73
© 2009 S. Paris, P. Kornprobst, J. Tumblin and
F. Durand
DOI: 10.1561/0600000020

Bilateral Filtering: Theory and Applications

Sylvain Paris[1], Pierre Kornprobst[2], Jack Tumblin[3] and Frédo Durand[4]

[1] *Adobe Systems, Inc., CA 95110-2704, USA, sparis@adobe.com*
[2] *NeuroMathComp Project Team INRIA, ENS Paris, UNSA LJAD, France, Pierre.Kornprobst@inria.fr*
[3] *Department of Electrical Engineering and Computer Science, Northwestern University, IL 60208, USA, jet@cs.northwestern.edu*
[4] *Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, MA 02139, USA, fredo@mit.edu*

Abstract

The bilateral filter is a non-linear technique that can blur an image while respecting strong edges. Its ability to decompose an image into different scales without causing haloes after modification has made it ubiquitous in computational photography applications such as tone mapping, style transfer, relighting, and denoising. This text provides a graphical, intuitive introduction to bilateral filtering, a practical guide for efficient implementation and an overview of its numerous applications, as well as mathematical analysis.

1

Introduction

Bilateral filtering is a technique to smooth images while preserving edges. It can be traced back to 1995 with the work of Aurich and Weule [4] on nonlinear Gaussian filters. It was later rediscovered by Smith and Brady [59] as part of their SUSAN framework, and Tomasi and Manduchi [63] who gave it its current name. Since then, the use of bilateral filtering has grown rapidly and is now ubiquitous in image-processing applications Figure 1.1. It has been used in various contexts such as denoising [1, 10, 41], texture editing and relighting [48], tone management [5, 10, 21, 22, 24, 53], demosaicking [56], stylization [72], and optical-flow estimation [57, 74]. The bilateral filter has several qualities that explain its success:

- Its formulation is simple: each pixel is replaced by a weighted average of its neighbors. This aspect is important because it makes it easy to acquire intuition about its behavior, to adapt it to application-specific requirements, and to implement it.
- It depends only on two parameters that indicate the size and contrast of the features to preserve.
- It can be used in a non-iterative manner. This makes the parameters easy to set since their effect is not cumulative over several iterations.

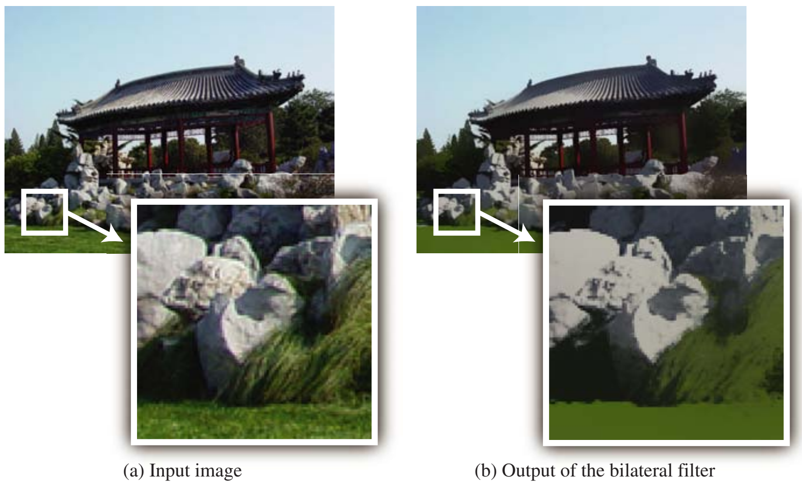

Fig. 1.1 The bilateral filter converts any input image (a)to a smoothed version (b). It removes most texture, noise, and fine details, but preserves large sharp edges without blurring.

- It can be computed at interactive speed even on large images, thanks to efficient numerical schemes [21, 23, 55, 54, 50, 71], and even in real time if graphics hardware is available [16].

In parallel to applications, a wealth of theoretical studies [6, 7, 13, 21, 23, 46, 50, 60, 65, 66] explain and characterize the bilateral filter's behavior. The strengths and limitations of bilateral filtering are now fairly well understood. As a consequence, several extensions have been proposed [14, 19, 23].

This paper is organized as follows. Section 2 presents linear Gaussian filtering and the nonlinear extension to the bilateral filter. Section 3 revisits several recent, novel and challenging applications of bilateral filtering. Section 4 compares different ways to implement the bilateral filter efficiently. Section 5 presents several links of bilateral filtering with other frameworks and also different ways to interpret it. Section 6 exposes extensions and variants of the bilateral filter. We also provide a website with code and relevant pointers (http://people.csail.mit.edu/sparis/bf_survey/).

2

From Gaussian Convolution to Bilateral Filtering

To introduce bilateral filtering, we begin with a description of Gaussian convolution in Section 2.2. This filter is simpler, introduces the notion of local averaging, and is closely related to the bilateral filter but does not preserve edges. Section 2.3 then underscores the specific features of the bilateral filter that combine smoothing with edge preservation. First, we introduce the notation used throughout this paper.

2.1 Terminology and Notation

For simplicity, most of the exposition describes filtering for a gray-level image I although every filtering operation can be duplicated for each component of a color image unless otherwise specified. We use the notation $I_{\mathbf{p}}$ for the image value at pixel position $\mathbf{p}$. Pixel size is assumed to be 1. $F[I]$ designates the output of a filter F applied to the image I. We will consider the set $\mathcal{S}$ of all possible image locations that we name the *spatial domain*, and the set $\mathcal{R}$ of all possible pixel values that we name the *range domain*. For instance, the notation $\sum_{\mathbf{p}\in\mathcal{S}}$ denotes a sum over all image pixels indexed by $\mathbf{p}$. We use $|\cdot|$ for the absolute value and $\|\cdot\|$ for the L_2 norm, e.g., $\|\mathbf{p}-\mathbf{q}\|$ is the Euclidean distance between pixel locations $\mathbf{p}$ and $\mathbf{q}$.

2.2 Image Smoothing with Gaussian Convolution

Blurring is perhaps the simplest way to smooth an image; each output image pixel value is a weighted sum of its neighbors in the input image. The core component is the convolution by a kernel which is the basic operation in linear shift-invariant image filtering. At each output pixel position it estimates the local average of intensities, and corresponds to low-pass filtering. An image filtered by Gaussian Convolution is given by:

$$GC[I]_{\mathbf{p}} = \sum_{\mathbf{q}\in\mathcal{S}} G_\sigma(\|\mathbf{p}-\mathbf{q}\|)\, I_{\mathbf{q}}, \tag{1}$$

where $G_\sigma(x)$ denotes the 2D Gaussian kernel (see Figure 2.1):

$$G_\sigma(x) = \frac{1}{2\pi\,\sigma^2}\,\exp\left(-\frac{x^2}{2\,\sigma^2}\right). \tag{2}$$

Gaussian filtering is a weighted average of the intensity of the adjacent positions with a weight decreasing with the spatial distance to the center position $\mathbf{p}$. The weight for pixel $\mathbf{q}$ is defined by the Gaussian $G_\sigma(\|\mathbf{p}-\mathbf{q}\|)$, where σ is a parameter defining the neighborhood size. The strength of this influence depends only on the spatial distance between the pixels and not their values. For instance, a bright pixel has a strong influence over an adjacent dark pixel although these two pixel values are quite different. As a result, image edges are blurred because pixels across discontinuities are averaged together (see Figure 2.1).

The action of the Gaussian convolution is independent of the image content. The influence that a pixel has on another one depends only their distance in the image, not on the actual image values.

Remark. Linear shift-invariant filters such as Gaussian convolution (Equation (1)) can be implemented efficiently even for very large σ using the Fast Fourier Transform (FFT) and other methods, but these acceleration techniques do not apply to the bilateral filter or other nonlinear or shift-variant filters. Fortunately, several fast numerical schemes were recently developed specifically for the bilateral filter (see Section 4).

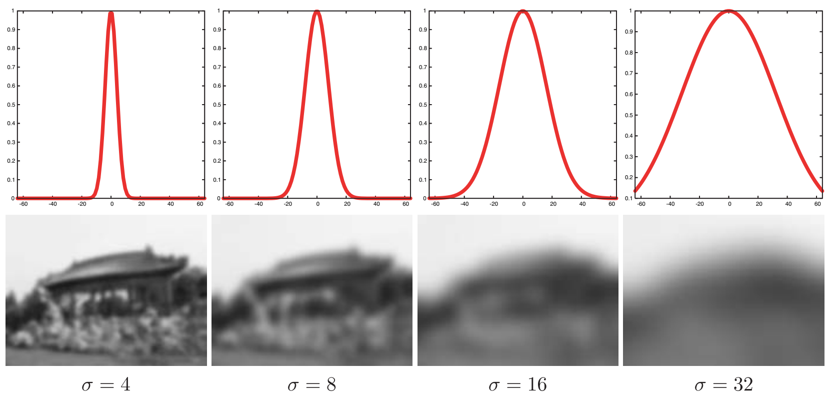

Fig. 2.1 Example of Gaussian linear filtering with different σ. Top row shows the profile of a 1D Gaussian kernel and bottom row the result obtained by the corresponding 2D Gaussian convolution filtering. Edges are lost with high values of σ because averaging is performed over a much larger area.

2.3 Edge-preserving Filtering with the Bilateral Filter

The bilateral filter is also defined as a weighted average of nearby pixels, in a manner very similar to Gaussian convolution. The difference is that the bilateral filter takes into account the difference in value with the neighbors to preserve edges while smoothing. The key idea of the bilateral filter is that for a pixel to influence another pixel, it should not only occupy a nearby location but also have a similar value.

The formalization of this idea goes back in the literature to Yaroslavsky [77], Aurich and Weule [4], Smith and Brady [59] and Tomasi and Manduchi [63]. The bilateral filter, denoted by $BF[\,\cdot\,]$, is defined by:

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}\in\mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)\, I_{\mathbf{q}}, \tag{3}$$

where normalization factor $W_{\mathbf{p}}$ ensures pixel weights sum to 1.0:

$$W_{\mathbf{p}} = \sum_{\mathbf{q}\in\mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(|I_{\mathbf{p}} - I_{\mathbf{q}}|). \tag{4}$$

Parameters σ_{s} and σ_{r} will specify the amount of filtering for the image I. Equation (3) is a normalized weighted average where $G_{\sigma_{\mathrm{s}}}$ is a spatial

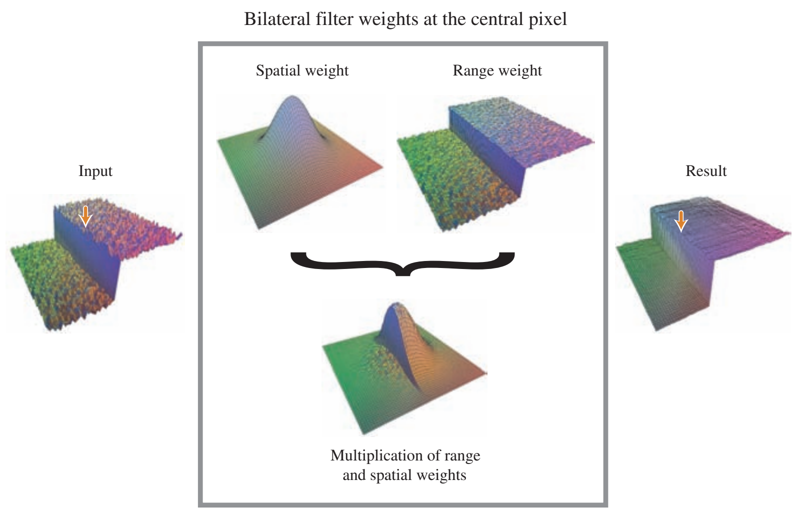
Bilateral filter weights at the central pixel
Spatial weight
Range weight
Input
Result
Multiplication of range and spatial weights

Fig. 2.2 The bilateral filter smooths an input image while preserving its edges. Each pixel is replaced by a weighted average of its neighbors. Each neighbor is weighted by a spatial component that penalizes distant pixels and range component that penalizes pixels with a different intensity. The combination of both components ensures that only nearby similar pixels contribute to the final result. The weights shown apply to the central pixel (under the arrow). The figure is reproduced from [21].

Gaussian weighting that decreases the influence of distant pixels, G_{σ_r} is a range Gaussian that decreases the influence of pixels $\mathbf{q}$ when their intensity values differ from $I_{\mathbf{p}}$. Figure 1.1 shows a sample output of the bilateral filter and Figure 2.2 illustrates how the weights are computed for one pixel near an edge.

2.3.1 Parameters

The bilateral filter is controlled by two parameters: σ_s and σ_r. Figure 2.3 illustrates their effect.

- As the range parameter σ_r increases, the bilateral filter gradually approximates Gaussian convolution more closely because the range Gaussian G_{σ_r} widens and flattens, i.e., is nearly constant over the intensity interval of the image.
- Increasing the spatial parameter σ_s smooths larger features.

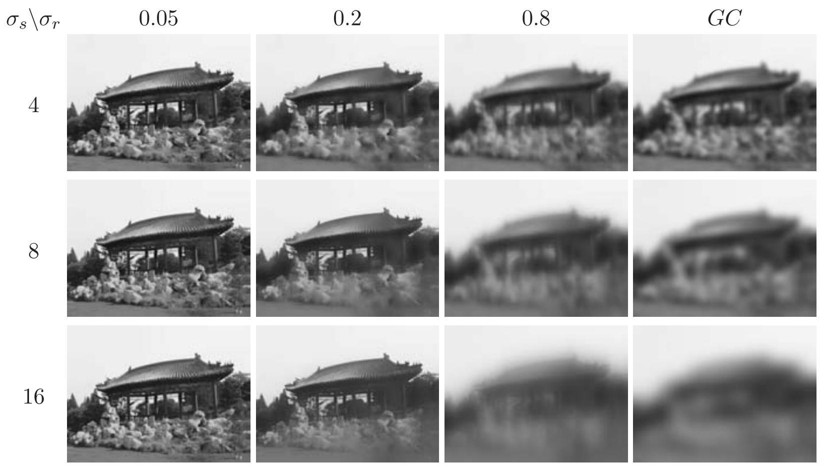

Fig. 2.3 The bilateral filter's range and spatial parameters provide more versatile control than Gaussian convolution. As soon as either of the bilateral filter weights reaches values near zero, no smoothing occurs. As a consequence, increasing the spatial sigma will not blur an edge as long as the range sigma is smaller than the edge amplitude. For example, note the rooftop contours are sharp for small and moderate range settings σ_{r}, and that sharpness is independent of the spatial setting σ_{s}. The original image intensity values span $[0,1]$.

In practice, in the context of denoising, Liu et al. [41] show that adapting the range parameter σ_{r} to estimates of the local noise level yields more satisfying results. The authors recommend a linear dependence: $\sigma_{\mathrm{r}} = 1.95\ \sigma_{\mathrm{n}}$, where σ_{n} is the local noise level estimate.

An important characteristic of bilateral filtering is that the weights are multiplied: if either of the weights is close to zero, no smoothing occurs. As an example, a large spatial Gaussian coupled with narrow range Gaussian achieves limited smoothing despite the large spatial extent. The range weight enforces a strict preservation of the contours.

2.3.2 Computational cost

At this stage of the presentation, skeptical readers may have already decided that the bilateral filter is an unreasonably expensive algorithm to compute when the spatial parameter σ_{s} is large, as it constructs each output pixel from a large neighborhood, requires the calculation of two weights, their products, and a costly normalizing step as well.

In Section 4 we will show some efficient approaches to implement the bilateral filter.

2.3.3 Iterations

The bilateral filter can be iterated. This leads to results that are almost piecewise constant as shown in Figure 2.4. Although this yields smoother images, the effect is different from increasing the spatial and range parameters. As shown in Figure 2.3, increasing the spatial parameters σ_s has a limited effect unless the range parameter σ_r is also increased. Although a large σ_r also produces smooth outputs, it tends to blur the edges whereas iterating preserves the strong edges such as the border of the roof in Figure 2.4 while removing the weaker details such as the tiles. This type of effect is desirable for applications such as stylization [72] that seek to abstract away the small details, while computational photography techniques [5, 10, 21] tend to use a single iteration to be closer to the initial image content.

2.3.4 Separation

The bilateral filter can split an image into two parts: the filtered image and its "residual" image. The filtered image holds only the large-scale features, as the bilateral filter smoothed away local variations without affecting strong edges. The residual image, made by subtracting the filtered image from the original, holds only the image portions that the filter removed. Depending on the settings and the application,

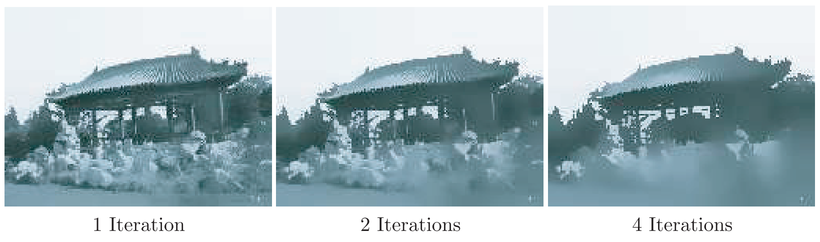

Fig. 2.4 Iterations: the bilateral filter can be applied iteratively, and the result progressively approximates a piecewise constant signal. This effect can help achieve a limited-palette, cartoon-like rendition of images [72]. Here, $\sigma_\mathrm{s} = 8$ and $\sigma_\mathrm{r} = 0.1$.

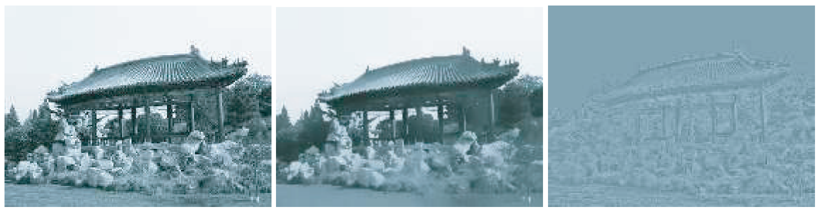

Fig. 2.5 Separation: The residual image holds all input components (a) removed by the bilateral filter (b), and some image structure is visible here (c). For denoising tasks, the ideal residual image would contain only noise, but here the σ_r setting was large enough to remove some fine textures that are nearly indistinguishable from noise, and still yields acceptable results for many denoising tasks.

this removed small-scale component can be interpreted as noise or texture, as shown in Figure 2.5. Applications such as tone management and style transfer extend this decomposition to multiple layers (see Section 3).

☛ *To conclude, bilateral filtering is an effective way to smooth an image while preserving its discontinuities (see Sections 3.1 and 3.5) and also to separate image structures of different scales (see Section 3.2). As we will see, the bilateral filter has many applications, and its central notion of assigning weights that depend on both space and intensity can be tailored to fit a diverse set of applications (see Section 6).*

Remark. The reader may know that the goal of edge-preserving image restoration has been addressed for many years by partial differential equations (PDEs), and one may wonder about their relationship with bilateral filters. Section 5.1 will explore those connections in detail.
