Foundations and Trends® in
Computer Graphics and Vision
Vol. 4, No. 1 (2008) 1–73

DOI: 10.1561/0600000020

now
the essence of knowledge

Bilateral Filtering: Theory and Applications

By Sylvain Paris, Pierre Kornprobst, Jack Tumblin and Frédo Durand

Contents

1 Introduction **2**

2 From Gaussian Convolution to Bilateral Filtering **4**

2.1 Terminology and Notation 4
2.2 Image Smoothing with Gaussian Convolution 5
2.3 Edge-preserving Filtering with the Bilateral Filter 6

3 Applications **11**

3.1 Denoising 12
3.2 Contrast Management 16
3.3 Depth Reconstruction 22
3.4 Data Fusion 22
3.5 3D Fairing 25
3.6 Other Applications 28

4 Efficient Implementation **33**

4.1 Brute Force 33
4.2 Separable Kernel 34
4.3 Local Histograms 35
4.4 Layered Approximation 36
4.5 Bilateral Grid 37

4.6 Bilateral Pyramid 40
4.7 Discussion 43

5 Relationship between Bilateral Filtering and Other Methods or Framework 44

5.1 Bilateral Filtering is Equivalent to Local Mode Filtering 44
5.2 The Bilateral Filter is a Robust Filter 47
5.3 Bilateral Filtering is Equivalent Asymptotically to the Perona and Malik Equation 51

6 Extensions of Bilateral Filtering 57

6.1 Accounting for the Local Slope 57
6.2 Using Several Images 62

7 Conclusions 65

Acknowledgments 67

References 68

Foundations and Trends® in
Computer Graphics and Vision
Vol. 4, No. 1 (2008) 1–73

DOI: 10.1561/0600000020

now
the essence of knowledge

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.

(a) Input image (b) Output of the bilateral filter

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).

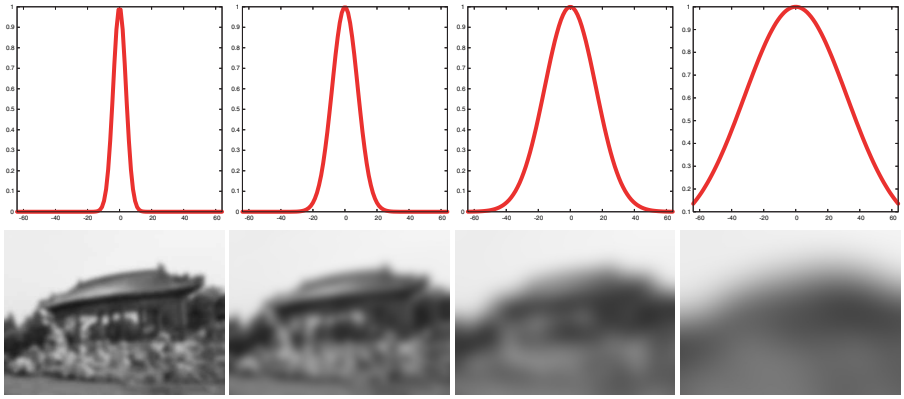


Fig. 2.1 Example of Gaussian linear filtering with different σ. Top row shows the profile of a 1D Gaussian kernel and bottom row the result obtained by the corresponding 2D Gaussian convolution filtering. Edges are lost with high values of σ because averaging is performed over a much larger area.

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_{σ_s} is a spatial

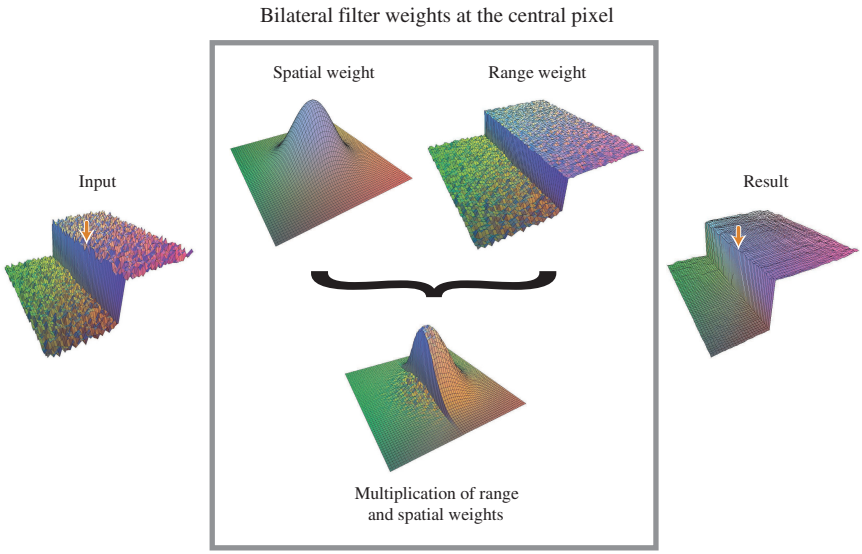


Fig. 2.2 The bilateral filter smooths an input image while preserving its edges. Each pixel is replaced by a weighted average of its neighbors. Each neighbor is weighted by a spatial component that penalizes distant pixels and range component that penalizes pixels with a different intensity. The combination 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.

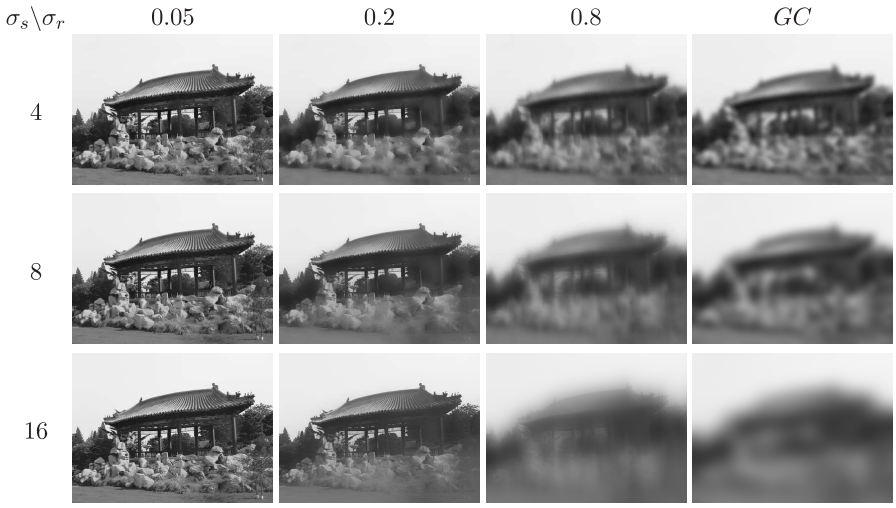


Fig. 2.3 The bilateral filter's range and spatial parameters provide more versatile control than Gaussian convolution. As soon as either of the bilateral filter weights reaches values near zero, no smoothing occurs. As a consequence, increasing the spatial sigma will not blur an edge as long as the range 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, $\sigma_s = 8$ and $\sigma_r = 0.1$.

(a) Input (b) Bilateral filter (c) Residual

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.

3

Applications

This section discusses the uses of the bilateral filter for a variety of applications:

- Denoising (Section 3.1): This is the original, primary goal of the bilateral filter, where it found broad applications that include medical imaging, tracking, movie restoration, and more. We discuss a few of these, and present a useful extension known as the cross-bilateral filter.
- Texture and Illumination Separation, Tone Mapping, Retinex, and Tone Management (Section 3.2): Bilateral filtering an image at several different settings decomposes that image into large-scale/small-scale textures and features. These applications edit each component separately to adjust the tonal distribution, achieve photographic stylization, or match the adjusted image to the capacities of a display device.
- Data Fusion (Section 3.4): These applications use bilateral filtering to decompose several source images into components and then recombine them as a single output image that inherits selected visual properties from each of the source images.

- 3D Fairing (Section 3.5): In this counterpart to image denoising, bilateral filtering applied to 3D meshes and point clouds smooths away noise in large areas and yet keeps all corners, seams, and edges sharp.
- Other Applications (Section 3.6): New applications are emerging steadily in the literature; we highlight several new trends indicated by recently published papers.

3.1 Denoising

One of the first roles of bilateral filtering was image denoising. Later, the bilateral filter became popular in the computer graphics community because it is edge preserving, easy to understand and set up, and because efficient implementations were recently proposed (see Section 4).

The bilateral filter has become a standard interactive tool for image denoising. For example, Adobe Photoshop® provides a fast and simple bilateral filter variant under the name "surface blur" (Figure 3.1). Instead of Gaussian functions, it uses a square box function as its spatial weight, and a "tent" (linear) function as the range weight. Unlike Gaussian convolution that smooths images without respecting their visual structures, the bilateral filter preserves the object contours and produces sharp results. The surface blur tool is often used by portrait photographers to smooth skin while preserving sharp edges and details in the subject's eyes and mouth.

(a) Input (b) Noisy image (c) "Surface blur" filter

Fig. 3.1 Denoising using the "surface blur" filter from Adobe Photoshop®: We added noise (b) to the input image (a) and applied the "surface blur" filter. As the input image was corrupted by noise, some signal loss is inevitable, but the filtered version is significantly improved.

Qualitatively, the bilateral filter represents an easy way to decompose an image into a cartoon-like component and a texture one. This cartoon-like image is the denoised image which can be used in several applications as shown in this section. Qualitatively, such a decomposition could be obtained by any simplifying filter. But this decomposition is not trivial from a mathematical perspective if one considers the mathematical structure of images. In this respect, we refer to Meyer [44], Vese and Osher [67], Aujol et al., [3] for more details about approaches dedicated to precise texture-cartoon decompositions.

The cartoon-like effect can also be a drawback depending on the application. Buades et al. [14] have shown that although bilateral filtering preserves edges, the preservation is not perfect and some edges are sharpened during process, introducing an undesirable "staircase effect". We discuss this effect in more detail in Section 6.1.3. In summary, the bilateral filter can be the right approach for many applications, but it is not always the best solution nor the best denoising filter available.

As a final comment, the bilateral filter is related to several approaches and frameworks proposed in the literature. We revisit the most important ones in Section 5. These analogies are interesting to notice, as they give theoretical foundations to bilateral filtering and show alternative formulations.

3.1.1 Medical Imagery

In the domain of medical imagery, Wong et al. [73] improved the structure preservation abilities of the bilateral filter by explicitly describing the structure with an additional weight, one that depends on the local shape and orientation of the sensed image data.

3.1.2 Videos

Bennett and McMillan [10] show that bilateral filtering can be used for videos. In this context, the bilateral filter is applied along the time axis, that is, pixels at the same location in successive frames are averaged together. The fact that the bilateral filter does not average together pixels of different colors prevents mixing data from different objects that appear at the same location but at different times. For instance,

(a) Input (b) Naive histogram stretching (c) Output of Bennett and McMillan [9]

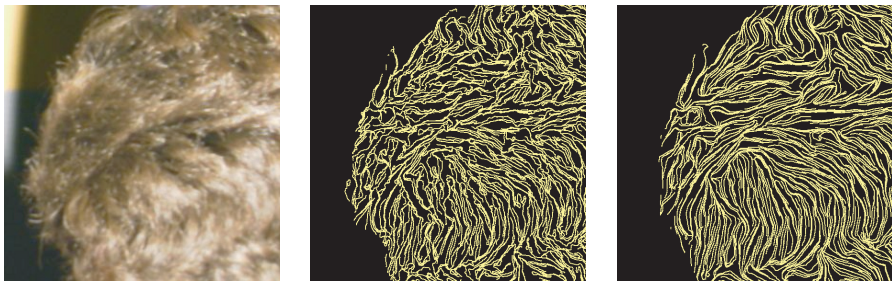
Fig. 3.2 Bennett and McMillan [10] describe how to combine spatial and temporal bilateral filterings to achieve high-quality video denoising and exposure correction. Figure reproduced from Bennett and McMillan [10].

if a red ball passes in front of green tree, the ball and tree pixels are not mixed together, thanks to the range weight of the bilateral filter. However, pixels that change color often, for instance due to a rapidly moving object, may not have any similar neighbors along the time axis. Bennett and McMillan compensate for this case by looking for spatial neighbors when there are not enough temporal similar pixels. Figure 3.2 shows sample results.

3.1.3 Orientation Smoothing

Paris et al. [49] use the bilateral filter to smooth the 2D orientation field computed from optical measurements for hairstyle modeling. Their measuring scheme yields a per-pixel evaluation of the local orientation, but these measures are noisy and at times ambiguous due to the complex nature of hair images. Paris et al. evaluated the success of their measurements at pixel $\mathbf{p}$ using the variance at $V_{\mathbf{p}}$ and incorporated it into their filter. In Paris' setup, several illumination conditions offer orientation estimates for each pixel, and they use the maximum difference Γ among all these estimates. As the orientation angle α varies cyclically between 0 and π, they map their averaging onto a complex exponential: $\alpha \in [0,\pi[\mapsto \exp\big(2i\alpha\big) \in \mathbb{C}$, leading to the filter:

$$\begin{aligned}&\rho \exp(2i\, F_{\mathrm{Paris}}(\alpha)_{\mathbf{p}}) \\ &\quad = \sum_{\mathbf{q}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p}-\mathbf{q}\|)\; G_{\sigma_V}(V_{\mathbf{p}}/V_{\mathbf{q}})\; G_{\sigma_\Gamma}\big(\Gamma(\mathbf{p},\mathbf{q})\big)\; \exp\left(2i\,\alpha_{\mathbf{q}}\right) \qquad (5)\end{aligned}$$



(a) Zoom on input image (b) Orientations before bilateral filtering (c) Orientations after bilateral filtering

Fig. 3.3 Paris et al. [49] smooth their orientation measurements using a variant of bilateral filtering mapped to the complex plane $\mathbb{C}$. Figure reproduced from Paris et al. [49].

This filter acts upon orientation mapped to the complex plane. Although Paris' application needs only the phase argument of the result and discards the amplitude ρ, if needed it could act as the standard deviation in the scalar case Watson, [70]. This filter illustrates how bilateral filtering can adapt to incorporate application-specific knowledge.

3.1.4 Discussion and Practical Consideration

Denoising usually relies on small spatial kernels σ_s and the range sigma σ_r is usually chosen to match the noise level.

The bilateral filter might not be the most advanced denoising technique but its strength lies in its simplicity and flexibility. The weights can be adjusted to take into account any metric on the difference between two pixels and information about the reliability of a given pixel can be included by reducing the weights assigned to it.

In the case of "salt-and-pepper" or impulse noise, the bilateral filter may need to mollify the input image before use. Though the noise may be sparse, the affected pixels' intensity values may span the entire image range (e.g., [0–1]), and their values might be too different from their neighbors to be filtered out. To mollify these images, compute the range Gaussian weights on a median-filtered version of the image [21]. If M describes median filtering, this gives:

$$BF[I]_\mathbf{p} = \frac{1}{W_\mathbf{p}} \sum_{\mathbf{q}\in\mathcal{S}} G_{\sigma_\mathrm{s}}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_\mathrm{r}}(|M[I]_\mathbf{p} - M[I]_\mathbf{q}|)\, I_\mathbf{q}, \tag{6}$$

$$W_{\mathbf{p}} = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_\mathrm{s}}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_\mathrm{r}}(|M[I]_{\mathbf{p}} - M[I]_{\mathbf{q}}|). \tag{7}$$

This practice is commonplace in robust statistics: users apply a very robust estimator such as the median filter first to obtain a suitable initial estimate, then apply a more precise estimator (the bilateral filter) to find the final result.

3.2 Contrast Management

Bilateral filtering has been particularly successful as a tool for contrast management tasks such as detail enhancement or reduction. Oh et al. [48] describe how to use the bilateral filter to separate an image into a large-scale component and a small-scale component by subtracting filtered results. With this decomposition, they edit texture in a photograph. Several earlier nonlinear coarse/fine decompositions were already in use in various local tone mapping operators (e.g., Stockham [62], Chiu et al. [17], Schlick [58], Pattanaik et al. [51], Tumblin and Turk [64]) but Durand and Dorsey [21] were the first to apply the method using the bilateral filter. Elad [24] followed the same strategy to estimate the illumination and albedo of the photographed scene. Bae et al. [5] extended this approach to manipulate the look of a photograph, and Fattal et al. [25] describe a multi-scale image decomposition that preserves edges and allows for combining multiple images to reveal object details. We describe these applications in the next sections.

3.2.1 Texture and Illumination Separation

In the context of image-based modeling, Oh et al. [48] used the structure-removal aspect of the bilateral filter. By using a sufficiently large range parameter σ_r, the bilateral filter successfully removes the variations due to reflectance texture while preserving larger discontinuities stemming from illumination changes and geometry. Their technique is motivated by the fact that illumination variations typically occur at a larger scale than texture patterns, as observed by Land in his "Retinex" theory of lightness perception [39, 38]. To extract the illumination component, they derive a variant of the iterated bilateral filter

for which the initial image is always filtered. The successive estimates are used only to refine the range weight:

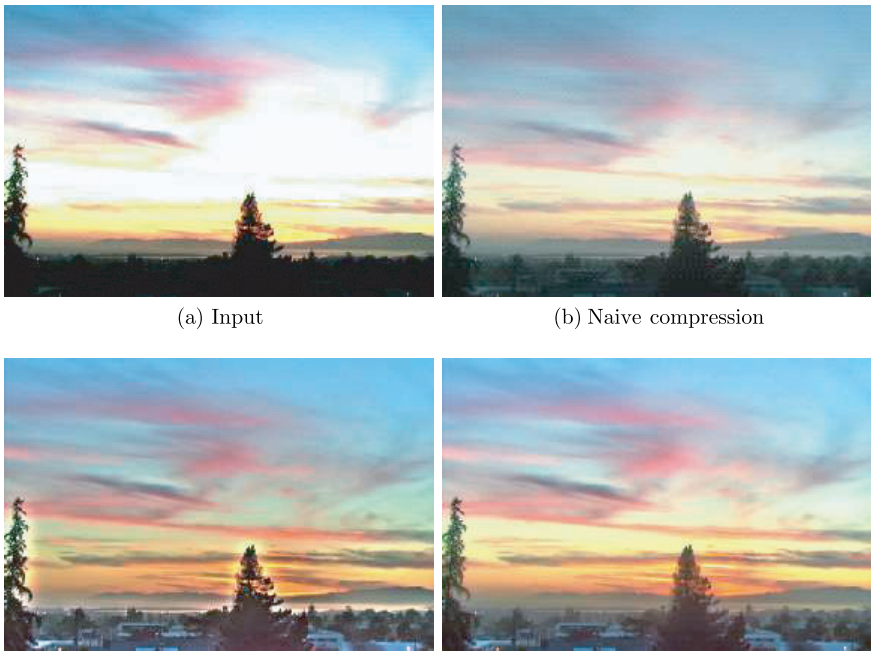
$$\widetilde{BF}_{i+1}[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}\in\mathcal{S}} G_{\sigma_\mathrm{s}}(\|\mathbf{p}-\mathbf{q}\|)\; G_{\sigma_\mathrm{r}}\Big(\widetilde{BF}_i[I]_{\mathbf{p}} - \widetilde{BF}_i[I]_{\mathbf{q}}\Big)\, I_{\mathbf{q}},$$

$$\text{with} \quad \widetilde{BF}_0[I] = I.$$

In addition, because a depth estimate is available at each image pixel, they adapt the spatial Gaussian size and shape to account for depth foreshortening. At each pixel they estimate a tangent plane to the local geometry, and choose an oriented spatial Gaussian that is isotropic in this tangent plane, which results in an anisotropic Gaussian once projected onto the image plane.

3.2.2 Tone Mapping

Durand and Dorsey [21] show that the use of bilateral filtering can be extended to isolate small-scale signal variations including texture and also small details of an image. They demonstrate this property to construct a tone mapping process whose goal is to compress the intensity values of an high-dynamic-range image to fit the capabilities of a low-dynamic-range display. In accordance with earlier local tone mapping operators, they note that naive solutions such as uniform scaling or gamma reductions to compress contrasts yield unsatisfactory results because the severe reductions needed for high contrast features cause subtle textures and scene details to vanish. While earlier tone mapping operators used multi-scale filter banks, wavelets, nonlinearities modeled on neural processes, and diffusion PDEs to separate visually compressible and incompressible components of log luminance, Durand and Dorsey used the bilateral filter for a fast, much simpler and visually pleasing result. They apply the bilateral filter on the log-intensities of the HDR image, scale down uniformly the result, and add back the filter residual, thereby ensuring that the small-scale details have not been compressed during the process. Some earlier methods such as Pattanaik et al. [51] used weighted multi-scale decompositions that model psychophysical models of visual appearance or relied on user interaction to achieve best-looking results (e.g., Jobson et al. [32], Tumblin and



(a) Input (b) Naive compression

(c) Compression after Gaussian decomposition (d) Output of Durand and Dorsey [2002]

Fig. 3.4 Tone Mapping: Direct display of an HDR image (a) is not satisfying because over- and under-exposed areas hide image features. Contrast compression maps all scene intensities to the display, but details in clouds and in the city below the horizon are barely visible (b). Isolating the details using Gaussian convolution brings back the details, but incurs halos near contrasted edges (e.g., near the tree silhouettes) (c). Durand and Dorsey use the bilateral filter to isolate the small variations of the input image without incurring halos (d). Figure reproduced from Durnad and Dorsey [21].

Turk [64]), but as shown in Figure 3.4 HDR images tone-mapped with Durand and Dorsey's technique are less difficult to achieve yet maintain a plausible, visually pleasing appearance.

3.2.3 Retinex

Elad [24] proposes a different interpretation of the tone mapping technique of Durand and Dorsey using the *Retinex* theory of Edwin Land that seeks a separation of images into illumination and albedo. Under the assumption that scene objects are "mostly-diffuse" reflectors that do not emit light, illumination values are greater than the measured intensities because objects always absorb part of the incoming light.

Elad adapts the bilateral filter to ensure that the filtered result fulfills this requirement and forms an upper envelope of the image intensities. He replaces the range weight G_{σ_r} by a truncated Gaussian $H \times G_{\sigma_\mathrm{r}}$, where H is a step function whose value is 1 for non-negative inputs and is 0 otherwise. As a consequence, at any given pixel $\mathbf{p}$, the local averaging includes only values greater than intensity at $\mathbf{p}$ and guarantees a filtered value at or above the local image intensity.

3.2.4 Tone Management

Bae et al. [5] build upon the separation between the large scale and the small scale offered by the bilateral filter, and describe a technique to transfer the visual look of an artist's picture onto a casual photograph. They explored a larger space of image modifications by applying an arbitrary, user-specified monotonic transfer function to the large-scale component of the source image. With histogram matching, they construct a transfer function that matches the global contrast and brightness of the model photograph. They also show that the small-scale component can be modified to vary the amount of texture visible in the image. To this end, they introduce the notion of *textureness* to quantify the local texture amplification they wish to induce in an image by cross-bilateral filtering (cf. Section 3.4.1 for detail). With the small-scale components (or high frequencies) H of the image's log-intensity $\log I$, the textureness is defined by:

$$\frac{1}{W_\mathbf{p}} \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_\mathrm{s}}(\|\mathbf{p} - \mathbf{q}\|)\; G_{\sigma_\mathrm{r}}(|\log I_\mathbf{p} - \log I_\mathbf{q}|)\, |H|_\mathbf{q}\,. \tag{8}$$

Said another way, textureness is the amplitude of the high frequencies that were locally averaged while respecting the edges of the input image.

Later, Chen et al. [16] sped up the bilateral filter computation using graphics hardware and achieved real-time results on high-definition videos, thereby enabling on-the-fly control of the photographic style.

3.2.5 Detail Enhancement

Fattal et al. [25] extend the small-scale/large-scale decomposition to multiple layers to allow for finer control and selection of enhanced

(a) Input

(b) Result after contrast and "textureness" increase

Fig. 3.5 Bae et al. [5] use the bilateral filter to separate the large-scale and small-scale variations of an image, and then processes them separately. In this example, users chose to increase the global image contrast and increase the texture as well for a more dramatic image result. Figure reproduced from Bae et al. [5].

(a) Sample input images

(b) Output with enhanced details

Fig. 3.6 Fattal et al. [25] use the bilateral filter to create a multi-scale decomposition of images. They first decompose several images of the same scene under different lighting conditions (a) and construct a new pyramid that generates a new image with enhanced details (b). Figure reproduced from Fattal et al. [25].

details. They use their decomposition on several images taken from the same point of view but under different lighting conditions and demonstrate a variety of effects by combining portions of bilateral image pyramids obtained from these lighting variations. They describe how these combinations can be controlled to reveal the desired details while avoiding the halo artifacts (Figure 3.6). They also describe a numerical scheme to efficiently compute image pyramids using the bilateral filter.

3.2.6 High-Dynamic-Range Hallucination

Wang et al. [69] use a bilateral filter decomposition to allow users to generate a high-dynamic-range image from a single low-dynamic-range one. They seek to reconstruct data in over- and under-exposed areas of the image. They use the bilateral filter to create a decomposition into texture and illumination inspired by Oh et al.'s [48] work. This allows them to apply user-guided texture synthesis to the detail (texture) layer, after bilateral filtering removed the large-scale illumination variations. Similarly, they can apply smooth interpolation to the large scale (illumination) layer because high-frequency texture has been decoupled.

3.2.7 Discussion and Practical Considerations

Contrast management relies on large spatial kernels to create large-scale/small-scale decompositions, because the small scale needs to include high- and medium-frequency components. The human visual system is not very sensitive to low frequencies but is quite sensitive to medium frequencies. As the large-scale component is typically the one that gets its contrast reduced, medium frequencies must be excluded from it to avoid attenuation as well.

For contrast management, the bilateral filter is usually applied to the log of the original image because the human visual system's response to light is approximately multiplicative. Using the log domain makes the range sigma act uniformly across different levels of intensity: edges where filtering should stop are defined in terms of multiplicative contrast. Similarly, relighting applications deal with a multiplicative process where illumination is multiplied by reflectance. The use of the log domain is not without its problems, as zero intensity maps to minus infinity and in dark regions noise in sensed intensity may be magnified in the log domain. Accordingly, many users add small constant on the order of the noise level to the input intensities before taking the log. The new color space proposed by Chong et al. [18] is particularly promising to handle these and other multiplicative processes. Using the luminance channel of the CIE-Lab color space is another useful alternative. Instead of a log curve, it is based on a cubic root that does not

model exactly these multiplicative processes but is numerically simpler to handle.

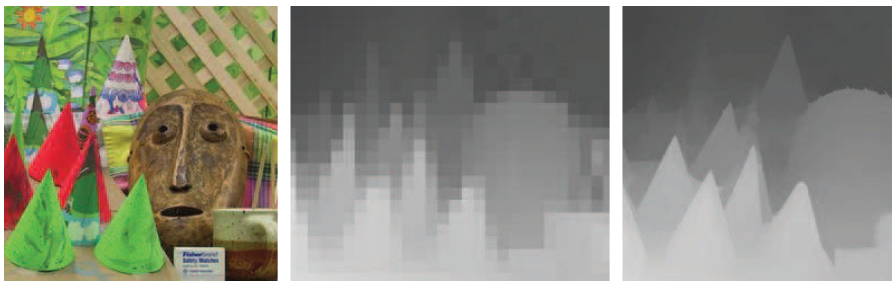
3.3 Depth Reconstruction

Yáng et al. [75, 76] and Yoon and Kweon [78] applied the bilateral filter to aid in stereo reconstruction, the recovery of depth values from correspondences between pixels different views. Ideally we wish to find a corresponding point in the right image for every pixel in the left image. As the distance between these point pairs, the disparity, is inversely proportional to the depth at that pixel, this information is equivalent to recovering the scene geometry. To pair the pixels with points in the other image, stereo algorithms typically compute a similarity score such as color differences or local correlation. Yáng et al. and Yoon and Kweon show that locally aggregating these scores using bilateral weights significantly improves the accuracy and reduces noise in the recovered depth maps. Yáng et al. [75] have tested many similarity scores and pairing strategies and found that the bilateral aggregation always improves their results.

3.4 Data Fusion

3.4.1 Flash/No-flash Imaging

Eisemann and Durand [22] and Petschnigg et al. [53] describe similar techniques to produce satisfying pictures in low-light conditions by combining a flash and a no-flash photograph. Their work is motivated by the fact that, although the flash image has unpleasantly direct and hard-looking lighting, its signal-to-noise ratio is higher than the no-flash image. On the other side, the no-flash image has more pleasing and natural-looking lighting, but its high frequencies are corrupted by noise and the camera may require a longer exposure time and increase the likelihood of blurring from an unsteady camera. The key idea is to extract the details of the flash image and combine them with the large-scale component of the no-flash picture. A variant of the bilateral filter performs this separation.



(a) Sample input image (b) Coarse resolution computation (c) Refinement using bilateral aggregation

Fig. 3.7 Yáng et al. [75] use the bilateral filter to achieve stereo reconstruction from photographs (a). First, they build a coarse depth map (b) and then use a scheme inspired from the bilateral filter to aggregate local information and compute a refined, more accurate depth map (c). Figure reproduced from Yang et al. [75].

Both articles introduced the *cross (joint) bilateral filter* to better process the no-flash photograph whose noise level is often too high to enable an accurate edge detection. As the flash image F represents the same scene, it is used to define the edges and the filtered no-flash image is obtained as:

$$CBF[N,F]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}\in\mathcal{S}} G_{\sigma_\mathrm{s}}(\|\mathbf{p}-\mathbf{q}\|)\; G_{\sigma_\mathrm{r}}(|F_{\mathbf{p}}-F_{\mathbf{q}}|)\; N_{\mathbf{q}}, \qquad (9)$$

where N is the original no-flash image. Figure 3.8 gives an overview of the process, and Figures 3.9 and 3.10 show sample results.

3.4.2 Multispectral Fusion

Bennett et al. [9] show how to exploit infrared data in addition to standard RGB data to denoise low-light video streams. They use the *dual bilateral filter*, a variant of the bilateral filter with a modified range weight that accounts for both the visible spectrum (RGB) and the infrared spectrum:

$$\begin{aligned} DBF[RGB]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}\in\mathcal{S}} & G_{\sigma_\mathrm{s}}(\|\mathbf{p}-\mathbf{q}\|)\; G_{\sigma_{RGB}}(\|RGB_{\mathbf{p}}-RGB_{\mathbf{q}}\|) \\ & \times G_{\sigma_{IR}}(|IR_{\mathbf{p}}-IR_{\mathbf{q}}|)\; RGB_{\mathbf{q}}, \end{aligned} \qquad (10)$$

where $RGB_{\mathbf{p}}$ is a 3-vector representing the RGB component at pixel $\mathbf{p}$, and $IR_{\mathbf{p}}$ the measured infrared intensity at the same pixel $\mathbf{p}$.

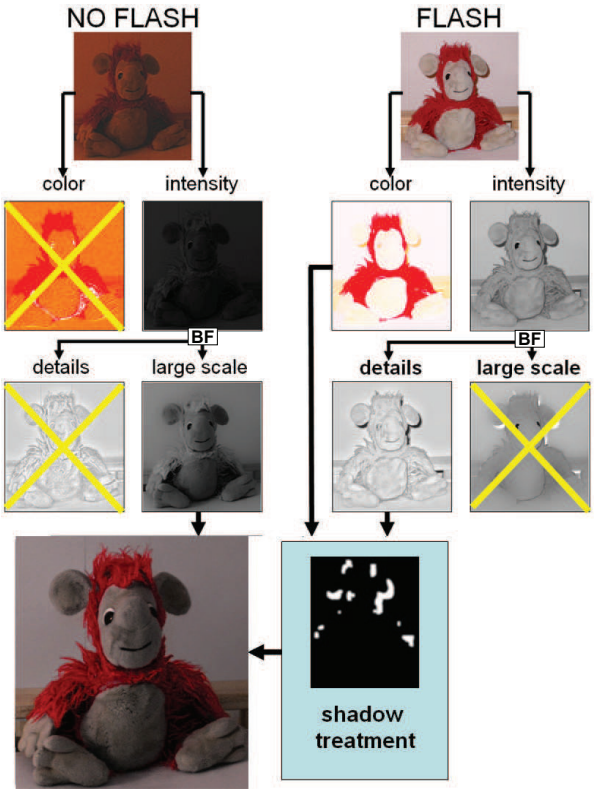


Fig. 3.8 Denoising of low-light images: Overview of the flash/no-flash combination of Eisemann and Durand [22]. The bilateral filter is used to combine the illumination component of the no-flash picture and the structure component of the flash picture. Figure reproduced from Eisemann and Durand [22].

(a) Photograph with flash (b) Photograph without flash (c) Combination

Fig. 3.9 By combining a flash photograph (a) and a no-flash photograph (b), Eisemann and Durand render a new photograph (c) that has both the warm lighting of the no-flash picture and the crisp details of the flash image. Figure reproduced from Eisemann and Durand [22].

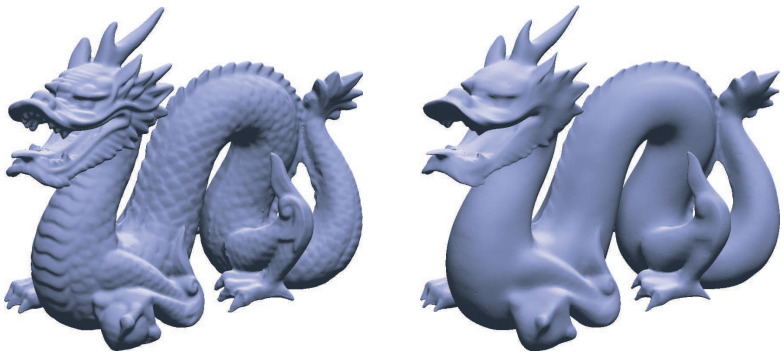
(a) Flash picture (b) No-flash picture (c) Output of Petschnigg *et al.* [53]

Fig. 3.10 By combining a flash photograph (a) and a no-flash photograph (b), Petschnigg et al. render a new photograph (c) that has both the warm lighting of the no-flash picture and the crisp details of the flash image. Figure reproduced from Petschnigg et al. [53].

Bennett et al. show that this combination better detects edges because it is sufficient for an edge to appear in just one of the channels (RGB or infrared) to form a sharp boundary in the result. In combination with temporal filtering, they demonstrate that it is possible to obtain high-quality video streams from noisy sequences of moving objects shot in very low light.

3.5 3D Fairing

Jones et al. [34] extend bilateral filtering to meshes. The difficulty compared to images is that all three xyz coordinates are subject to noise, data are not regularly sampled, and the z coordinate is not a function of x and y unlike the pixel intensity. To smooth a mesh, Jones et al. assume that it is locally flat. Under this assumption and in the absence of noise, a vertex $\mathbf{p}$ belongs to the plane tangent to the mesh at any nearby vertex $\mathbf{q}$. With $\pi_{\mathbf{q}}(\mathbf{p})$ the projection of $\mathbf{p}$ onto the plane tangent to the mesh at $\mathbf{q}$, ideally we have $\mathbf{p} = \pi_{\mathbf{q}}(\mathbf{p})$. However, because of noise and because the mesh is not flat everywhere, this relationship does not hold in general. To smooth the mesh, Jones et al. average the position of $\mathbf{p}$ predicted by $\pi_{\mathbf{q}}(\mathbf{p})$, they apply a spatial weight $G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)$ which ensures that only nearby points contribute to the estimate. They add a term $G_{\sigma_r}(\|\mathbf{p} - \pi_{\mathbf{q}}(\mathbf{p})\|)$ that reduces the weights of outliers, i.e., the predictions $\pi_{\mathbf{q}}(\mathbf{p})$ that are far away from the original position $\mathbf{p}$. Using



(a) Input mesh (b) Smoothed mesh

Fig. 3.11 Jones et al. [34] have adapted the bilateral filter to smooth 3D meshes while preserving their most prominent features. Figure reproduced from Jones et al. [34].

a term $a_{\mathbf{q}}$ to account for the sampling density, the resulting filter is:

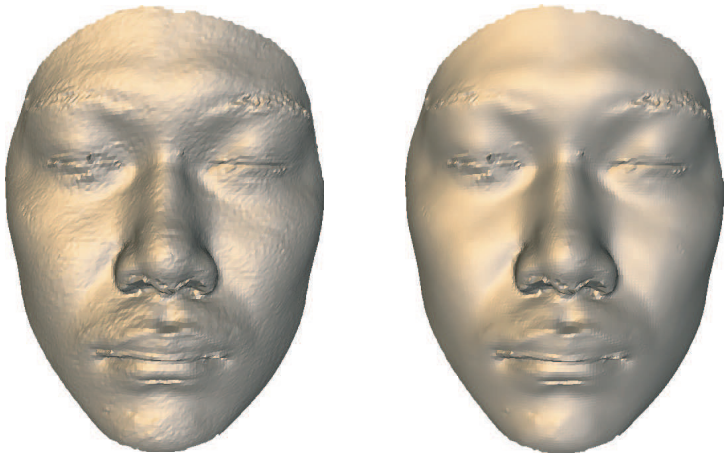
$$F_{\text{Jones}}(\mathbf{p}) = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}} a_{\mathbf{q}}\, G_{\sigma_{\text{s}}}(\|\mathbf{p}-\mathbf{q}\|)\; G_{\sigma_{\text{r}}}(\|\mathbf{p}-\pi_{\mathbf{q}}(\mathbf{p})\|)\; \pi_{\mathbf{q}}(\mathbf{p}). \quad (11)$$

To improve the results, they mollify the mesh normals used to estimate the tangent planes [30, 47], that is, they apply a low-pass filter on the normals. This mollification is analogous to the pre-filtering step described by Catté et al. [15] for PDE filters. Figure 3.11 shows a sample result.

Fleishman et al. [26] simultaneously proposed a similar approach (Figure 3.12). The main difference between the techniques of Jones et al. and Fleishman et al. [26] is the way Jones expresses his flat neighborhood assumption. Fleishman et al. use the mesh normal $\mathbf{n_p}$ at $\mathbf{p}$ and project neighbors onto it. With $\mathbf{q}$ is such a neighbor, $\mathbf{q}$ should project on $\mathbf{p}$, that is: $\mathbf{p} + [(\mathbf{q}-\mathbf{p})\cdot\mathbf{n_p}]\,\mathbf{n_p} = \mathbf{p}$. This results in the following variant of the bilateral filter:

$$F_{\text{Fleishman}}(\mathbf{p}) = \mathbf{p} + \frac{\mathbf{n_p}}{W_{\mathbf{p}}} \sum_{\mathbf{q}} G_{\sigma_{\text{s}}}(\|\mathbf{p}-\mathbf{q}\|)\; G_{\sigma_{\text{r}}}(|(\mathbf{q}-\mathbf{p})\cdot\mathbf{n_p}|)[(\mathbf{q}-\mathbf{p})\cdot\mathbf{n_p}]. \quad (12)$$

The projection on the normal can be rewritten using the plane projection operator π used by Jones et al.: $[(\mathbf{q}-\mathbf{p})\cdot\mathbf{n_p}]\,\mathbf{n_p} = \mathbf{q} - \pi_{\mathbf{p}}(\mathbf{q})$.



(a) Input (b) Output of Fleishman *et al.* [2003]

Fig. 3.12 Fleishman et al. [34] have adapted the bilateral filter to smooth 3D meshes while preserving their most prominent features. Figure reproduced from Fleishman et al. [26].

This leads to the following expression equivalent to Equation (12):

$$F_{\text{Fleishman}}(\mathbf{p}) = \mathbf{p} + \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}} G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_r}(\|\mathbf{q}-\pi_{\mathbf{p}}(\mathbf{q})\|)\big(\mathbf{q}-\pi_{\mathbf{p}}(\mathbf{q})\big). \quad (13)$$

These two formulations underline the differences between the approaches of Jones et al. and Fleishman et al. Equation (12) shows that, unlike Jones et al., Fleishman et al. guarantee no vertex drift by moving $\mathbf{p}$ only along its normal $\mathbf{n}_{\mathbf{p}}$. On the other hand, Fleishman et al. do not compensate for the density variations described by Jones et al. Furthermore, Equation (13) shows that the weights between both approaches are similar except that Jones et al. project $\mathbf{p}$ on the tangent plane at $\mathbf{q}$ and thus exploit both the position and normal of all neighbors $\mathbf{q}$, whereas Fleishman et al. project $\mathbf{q}$ on the tangent plane at $\mathbf{p}$, thereby exploiting first-order information only from the vertex $\mathbf{p}$. This suggests a hybrid filter that we have not yet evaluated:

$$F_{\text{hybrid}}(\mathbf{p}) = \mathbf{p} + \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}} a_{\mathbf{q}} G_{\sigma_s}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_r}(\|\mathbf{p}-\pi_{\mathbf{q}}(\mathbf{p})\|)(\mathbf{q}-\pi_{\mathbf{p}}(\mathbf{q})). \quad (14)$$

In addition to these differences in estimating the vertex positions, Fleishman et al. advocate iterating the filter three times for further smoothing of the mesh geometry. Wang [68] refines the process by explicitly detecting the sharp-edge vertices to preserve them. He remeshes the model at these edges to ensure that sharp features are correctly represented by an edge between two triangles.

Later, Jones et al. [33] refined their technique to filter normals. Applying a geometric transformation f to the 3D space given by $\mathbf{x} \in \mathbb{R}^3 \mapsto F(\mathbf{x})$, Jones transforms the normals by the transposed inverse of the Jacobian of F. The Jacobian of F is a 3×3 matrix that captures the first-order deformation induced by F and is defined by $J_{ij}(F) = \partial F_i / \partial \mathbf{x}_j$ where F_i is the ith coordinate of F, and $\mathbf{x}_j$ the jth coordinate of $\mathbf{x}$. Jones et al. show that iteratively transforming the normals by $J^{-\mathrm{T}}(F_{\mathrm{Jones}})$ smooths the normals of a model while respecting its edges and without moving its vertices. They argue that not moving the vertices yields a better preservation of the fine details of the meshes.

Miropolsky and Fischer [45] propose a variant of bilateral filtering to smooth and decimate 3D point clouds. They assume that a normal $\mathbf{n_p}$ is known for each point $\mathbf{p}$. They overlay a regular 3D grid on top of the points and determine a representative point for each grid cell by taking into account the point location and normal. With $\mathbf{c}$ the cell center and $\mathbf{n_c}$ the mean normal of the cell points, they propose:

$$F_{\mathrm{Miropolsky}}(\mathbf{c}) = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{c} - \mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(\mathbf{n_c} \cdot \mathbf{n_q})\mathbf{q} \qquad (15)$$

3.6 Other Applications

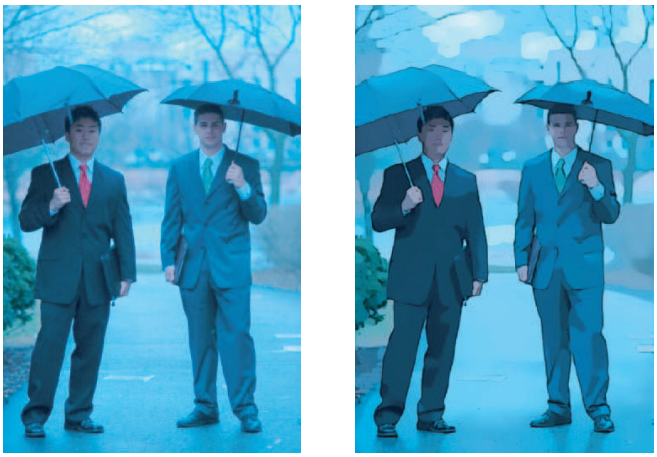
3.6.1 Depth Map from Luminance

Khan et al. [35] use bilateral filtering to process the luminance channel of an image and obtain a pseudo-depth map that is sufficient for altering the material appearance of the observed object. The originality of this use of the bilateral filter is that the smoothing power of the bilateral filter determines the geometric characteristics of an object. For instance, a smaller intensity tolerance σ_{r} results in a depth map that looks "engraved" with the object texture, because the intensity

patterns are well preserved and directly transferred to the map as depth variations.

3.6.2 Video Stylization

Winnemöller et al. [72] iterate the bilateral filter to simplify video content and achieve a cartoon look (Figure 3.13). They demonstrate that the bilateral filter can be computed in real time at video resolution using the numerical scheme of Pham and van Vliet [55] on modern graphics hardware. Later, Chen et al. [16] ported the bilateral filter on the GPU using the *bilateral grid* and achieved similar results on high-definition videos. Winnemöller et al. demonstrate that bilateral filtering is an effective preprocessing for edge detection: filtered images trigger fewer spurious edges. To modulate the smoothing strength of the bilateral filter, they modify it to control the degree of edge preservation. The range weight G_{σ_r} is replaced by $(1 - m) \cdot G_{\sigma_r} + m \cdot u$ where m is a function varying between 0 and 1 to control edge preservation, and u defines the local importance of the image. To define u and m, Winnemöller et al. suggest using an eye tracker [20], a computational model of saliency [31], or a user-painted map.



(a) Input (b) Abstracted output

Fig. 3.13 Sample abstraction result from the method by Winnemöller et al. [72]. Reproduced from Winnemöller et al. [72].

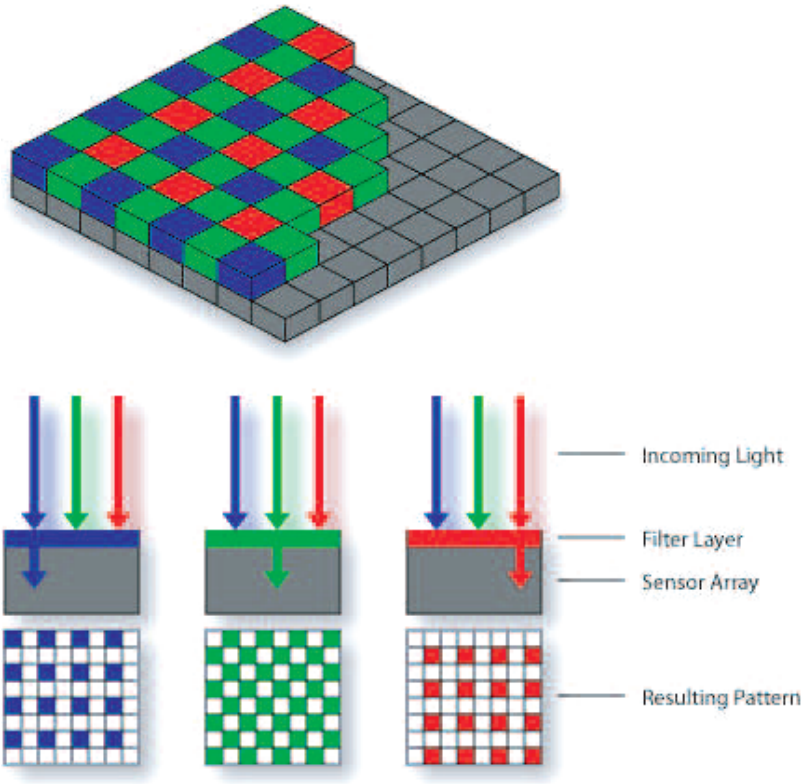


Fig. 3.14 Bayer patterns are such that, although each pixel is missing two color channels, adjacent pixels have measures in these missing channels. Figure reproduced from Wikipedia (http://en.wikipedia.org/wiki/Bayer_filter).

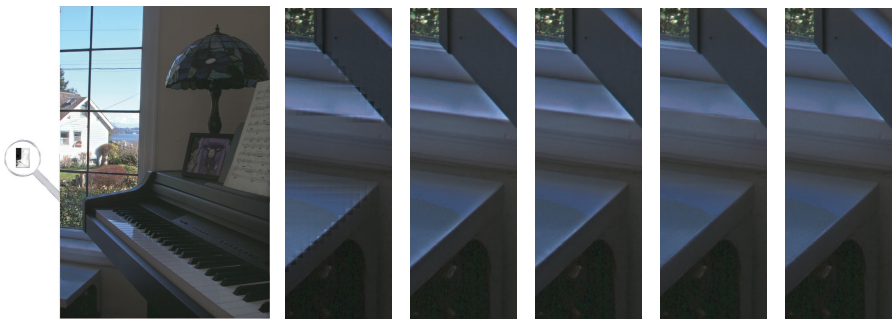
3.6.3 Demosaicking

Demosaicking is the process of recovering complete color information from partial color sampling through a Bayer filter (see Figure 3.14). Ramanath and Snyder [56] interpolate missing color values of Bayer patterns [8]. These patterns are used in digital cameras where each sensor measures only a single value among red, green, and blue. Bayer patterns are such that, although each pixel is missing two color channels, adjacent pixels have measures in these missing channels. Demosaicking is thus a small-scale interpolation problem where values are interpolated from neighbor pixels. Directly interpolating the values

yields blurry images because edges are ignored. Ramanath and Snyder start from such an image and refine the result with bilateral filtering. They use a small spatial neighborhood to consider only the pixels within the 1-ring of the filtered pixel, and also ensure that measured values are not altered. The validation shows that the obtained results compare favorably to state-of-the-art techniques although the computational cost is higher.

3.6.4 Optical Flow

Xiao et al. [74] apply bilateral filtering to regularize the optical flow computation. They use an iterative scheme to refine the flow vectors between a pair of images. Each iteration consists of two steps: first the vectors are adjusted using a scheme akin to Lucas and Kanade [42], then the flow vectors are smoothed using a modified version of bilateral filtering that has two additional terms, one accounting for flow similarity, and one that ensures that occluded regions are ignored during averaging. This scheme also "fills in" occluded regions, estimating depth for pixels visible in one image of the pair but hidden in the other. These occluded points gather information from pixels outside the occluded region covered by the bilateral filter kernel, and the range



(a) Upsampled result (b) Nearest neighbor (c) Bicubic (d) Gaussian (e) Joint bilateral (f) Ground truth

Fig. 3.15 Sample use of *joint bilateral upsampling* [37] to tone map a high-resolution HDR image. In this context, the method is used to upsample the exposure map (a) applied to the pixel values to obtain the output (e) that is close to the ground-truth result (f) and does not exhibit the defects of other upsampling methods (b–d). Figure reproduced from Kopf et al. [37].

weight ensures that only similar points contribute, thereby avoiding data diffused from the "wrong side of the occlusion". An important feature of this technique is that it actually regularizes the computation, i.e., the bilateral filter does not optimize a trade-off between a data term and smoothness term, it only makes the data smoother. Nonetheless, the process as a whole is a regularization because it interleaves bilateral filtering with an optimization step, and can be seen as a progressive refinement of the initial guess of a steepest-slope optimization. Sand and Teller [57] accelerate this technique by restricting the use of bilateral filtering near the flow discontinuities.

3.6.5 Upsampling

Kopf et al. [37] describe *joint bilateral upsampling*, a method inspired from the bilateral filter to upsample image data. The advantage of their approach is that it is generic and can potentially upsample any kind of data such as the exposure map used for tone mapping or hues for colorization. Given a high-resolution image and a downsampled version, one can compute the data at low resolution and then upsample them using a weighted average. High-resolution data are produced by averaging the samples in a 5×5 window at low resolution. The weights are similar to those defined by the bilateral filter, as each neighboring pixels' influence decreases with distance and color difference. As a result, Kopf's scheme interpolates low-resolution data while respecting the discontinuities of the high-resolution input image. This filter is fast to evaluate because it only considers a small spatial footprint.

4

Efficient Implementation

A naive implementation of the bilateral filter can be extremely slow, especially for large spatial kernels. Several approaches have been proposed to speed up the computation. They all rely on approximations that yield various degrees of acceleration and accuracy. In this section, we describe these efficient algorithms and compare their performances. We begin with the brute force approach as reference. We then describe the techniques based on separable kernels of Pham [55] and Pham and van Vliet [54], the local histogram of Weiss [71], and the bilateral grid [16, 50]. Figure 4.3 at the end of this section provides a visual comparison of the achieved results.

4.1 Brute Force

A direct implementation of the bilateral filter consists of two nested loops, as presented in Table 4.1.

The complexity of this algorithm is $\mathcal{O}\left(|\mathcal{S}|^2\right)$, where $|\mathcal{S}|$ the size of the spatial domain (i.e., the number of pixels). This quadratic complexity quickly makes the computational cost explode for large images.

Table 4.1 Algorithm for the direct implementation of bilateral filter.

For each pixel $\mathbf{p}$ in $\mathcal{S}$

(1) Initialization: $BF[I]_{\mathbf{p}} = 0$, $W_{\mathbf{p}} = 0$

(2) For each pixel $\mathbf{q}$ in $\mathcal{S}$

(a) $w = G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)\ G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)$

(b) $BF[I]_{\mathbf{p}} \mathrel{+}= w I_{\mathbf{q}}$

(c) $W_{\mathbf{p}} \mathrel{+}= w$

(3) Normalization: $BF[I]_{\mathbf{p}} = I_{\mathbf{p}} / W_{\mathbf{p}}$

A classical improvement is to restrict the inner loop to the neighborhood of the pixel $\mathbf{p}$. Typically, one considers only the pixels $\mathbf{q}$ such that $\|\mathbf{p} - \mathbf{q}\| \leq 2\sigma_s$. The rationale is that the contributions of pixels farther away than $2\sigma_s$ is negligible because of the spatial Gaussian. This leads to a complexity on the order of $\mathcal{O}\left(|\mathcal{S}|\,{\sigma_s}^2\right)$. This implementation is efficient for small spatial kernels, that is, small values of σ_s but become quickly prohibitive for large kernels because of the quadratic dependence in σ_s.

4.2 Separable Kernel

Pham and van Vliet [55] propose to approximate the 2D bilateral filter by two 1D bilateral filters applied one after the other. First, they filter each image column and then each row. Each time, they use the brute force algorithm restricted to a 1D domain, that is, the inner loop on pixels $\mathbf{q}$ is restricted to pixels on the same column (or row) as the pixel $\mathbf{p}$. As a consequence, the complexity becomes $\mathcal{O}\left(|\mathcal{S}|\,\sigma_s\right)$ because the considered neighborhoods are 1D instead of 2D. This approach yields significantly faster running times but the performance still degrades linearly with the kernel size. Furthermore, this approach computes an axis-aligned separable approximation of the bilateral filter kernel. Although this approximation is satisfying for uniform areas and straight edges, it forms a poor match to more complex features such as textured

regions. As a consequence, axis-aligned "streaks" may appear with large kernels in such regions (Figure 4.3). Pham [54] describes how to steer the separation according to the local orientation in the image to reduce these streaks. This approach improves the quality of the results, especially on slanted edges, but is computationally more involved because the 1D filters are no longer axis aligned.

4.3 Local Histograms

Weiss [71] considers the case where the spatial weight is a square box function, that is, he rewrites the bilateral filter as:

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in \mathcal{N}_{\sigma_\mathrm{s}}(\mathbf{p})} G_{\sigma_\mathrm{r}}(|I_{\mathbf{p}} - I_{\mathbf{q}}|) I_{\mathbf{q}} \tag{16a}$$

$$W_{\mathbf{p}} = \sum_{\mathbf{q} \in \mathcal{N}_{\sigma_\mathrm{s}}(\mathbf{p})} G_{\sigma_\mathrm{r}}(|I_{\mathbf{p}} - I_{\mathbf{q}}|), \tag{16b}$$

where $\mathcal{N}_{\sigma_\mathrm{s}}(\mathbf{p}) = \{\mathbf{q}, \|\mathbf{p} - \mathbf{q}\|_1 \leq \sigma_\mathrm{s}\}$. In this case, the result depends only on the histogram of the neighborhood $\mathcal{N}_{\sigma_\mathrm{s}}(\mathbf{p})$ because the actual position of the pixel within the neighborhood is not taken into account.

Following this remark, Weiss exposes an efficient algorithm to compute the histogram of the square neighborhoods of an image. We refer to his article for the detail of the algorithm. The intuition behind his approach is that the neighborhoods $\mathcal{N}_{\sigma_\mathrm{s}}(\mathbf{p}_1)$ and $\mathcal{N}_{\sigma_\mathrm{s}}(\mathbf{p}_2)$ of two adjacent pixels $\mathbf{p}_1$ and $\mathbf{p}_2$ largely overlap. Based on this remark, Weiss describes how to efficiently compute the histogram of $\mathcal{N}_{\sigma_\mathrm{s}}(\mathbf{p}_1)$ by exploiting the similarity with the histogram of $\mathcal{N}_{\sigma_\mathrm{s}}(\mathbf{p}_2)$. Once the histogram of $\mathcal{N}_{\sigma_\mathrm{s}}(\mathbf{p})$ is known for a pixel $\mathbf{p}$, the result of the bilateral filter $BF[I]_{\mathbf{p}}$ (Equation (16a)) can be computed because each histogram bin indicates how many pixels $\mathbf{q}$ have a given intensity value I. A straightforward application of this technique produces band artifacts near strong edges, *a.k.a.* Mach bands, because the frequency spectrum of the box filter is not band-limited. Weiss addresses this issue by iterating his filter three times, which effectively smooths away the artifacts.

Weiss [71] then demonstrates that his algorithm has a complexity on the order of $\mathcal{O}(|\mathcal{S}| \log \sigma_\mathrm{s})$ which makes it able to handle any kernel size in short times. Furthermore, his algorithm is designed such that

it can take advantage of the vector instruction set of modern CPUs, thereby yielding running times on the order of one second for images of several megapixels each. Unfortunately, the algorithm processes color images independently for each channel, which can introduce bleeding artifacts; in addition, it is unclear how to extend this filter for use in cross bilateral filtering applications.

4.4 Layered Approximation

Durand and Dorsey [21] propose a fast approximation based on the intuition that the bilateral filter is almost a convolution of the spatial weight $G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)$ with the product $G_{\sigma_r}(|I_\mathbf{p} - I_\mathbf{q}|)\, I_\mathbf{q}$ (Equation (3)). But the bilateral filter is not a convolution because the range weight $G_{\sigma_r}(|I_\mathbf{p} - I_\mathbf{q}|)$ depends on the pixel value $I_\mathbf{p}$. Durand and Dorsey overcame this by picking a fixed intensity value i, computing the product for it, $G_{\sigma_r}(|i - I_\mathbf{q}|)\, I_\mathbf{q}$, and convolving it with the Gaussian kernel G_{σ_r}. After normalization, this gives the exact result of the bilateral filter at all pixels $\mathbf{p}$ such that $I_\mathbf{p} = i$. Computing the bilateral filter this way would be extremely slow because it requires a convolution for each possible pixel value i.

Instead, Durand and Dorsey propose a two-step speed-up. First, they select a sparse subset $\{i_0, \ldots, i_n\}$ of the intensity values. For each value i_k, they evaluate the product $G_{\sigma_r}(|i_k - I_\mathbf{q}|)I_\mathbf{q}$. This produces layers $\tilde{L}_0, \ldots, \tilde{L}_n$. Each $\tilde{L}_k$ is then convolved with the spatial kernel G_{σ_s} and normalized to form a new layer $\hat{L}_k$ that contains the exact results of the bilateral filter for pixels with intensity equal to i_k. For pixels whose intensities have not been sampled, the result is linearly interpolated from the two closest layers. To further speed up the process, they downsample the image I prior to computing the product with the range weight G_{σ_r} and convolving with the spatial kernel G_{σ_s}. The final layers $\hat{L}_0, \ldots, \hat{L}_n$ are obtained by upsampling the convolution outputs. The bilateral filter results are still obtained by linearly interpolating between the two closest layers (Table 4.2).

Durand and Dorsey's approximation dramatically speeds up the computation. Whereas a brute force implementation requires several minutes of computation for a megapixel image, their scheme runs in

Table 4.2 Reformulation proposed by Durand and Dorsey [21].

1. Given a 2D image I, compute a low-resolution version $\tilde{I}$, pick a set of intensities $\{i_0, \ldots, i_n\}$, and compute layers $\tilde{L}_0, \ldots, \tilde{L}_n$:

$$\tilde{L}_k(\mathbf{q}) = G_{\sigma_r}(|i_k - \tilde{I}_\mathbf{q}|)\, \tilde{I}_\mathbf{q}.$$

2. Convolve each layer with the spatial kernel and normalize the result:

$$\bar{L}_k = (G_{\sigma_s} \otimes \tilde{L}_k) \div (G_{\sigma_s} \otimes G_{\sigma_r}),$$

where $\div$ indicates a per-pixel division and $G_{\sigma_s} \otimes G_{\sigma_r}$ corresponds to the sum of the weights at each pixel.

3. Upsample the layers $\bar{L}_k$ to get $\hat{L}_k$.
4. For each pixel $\mathbf{p}$ with intensity $I_\mathbf{p}$, find the two closest values i_{k_1} and i_{k_2}, and output the linear interpolation:

$$BF[I]_\mathbf{p} \approx \frac{I_\mathbf{p} - i_{k_1}}{i_{k_2} - i_{k_1}} \hat{L}_{k_2} + \frac{i_{k_2} - I_\mathbf{p}}{i_{k_2} - i_{k_1}} \hat{L}_{k_1}.$$

about one second. The downside of this approach is that in practice, the achieved result can be significantly different from the reference brute force implementation, and there is no formal characterization of this difference. In the next section, we discuss the scheme of Paris and Durand [50] that is inspired by the layered approximation, and achieves an equivalent speed-up but with significantly better accuracy. We discuss the relationship between both approaches at the end of the following section.

4.5 Bilateral Grid

Inspired by the layered approximation of Durand and Dorsey [21], Paris and Durand [50] have reformulated the bilateral filter in a higher dimensional homogeneous space. They described a new image representation

where a gray-level image is represented in a volumetric data structure that they named the *bilateral grid*. In this representation, a 2D image I is represented by a 3D grid Γ where the first two dimensions of the grid corresponds to the pixel position $\mathbf{p}$ and the third dimension correspond to the pixel intensity $I_{\mathbf{p}}$. In addition, this 3D grid stores homogeneous values, that is, the intensity value I is associated with a non-negative weight w and stored as an homogeneous vector (wI, w). Using this concept, Paris and Durand [50] showed that the bilateral filter corresponds to a Gaussian convolution applied to the grid, followed by sampling and normalization of the homogeneous values.

More precisely, the authors consider the $\mathcal{S} \times \mathcal{R}$ domain and represent a gray-scale image I as a 3D grid Γ:

$$\Gamma(p_x, p_y, r) = \begin{cases} \big(I(p_x, p_y), 1\big) & \text{if } r = I(p_x, p_y) \\ (0,0) & \text{otherwise} \end{cases}. \tag{17}$$

With this representation, they demonstrate that bilateral filtering exactly corresponds to convolving Γ with a 3D Gaussian whose parameters are $(\sigma_{\mathrm{s}}, \sigma_{\mathrm{s}}, \sigma_{\mathrm{r}})$: $\tilde{\Gamma} = \Gamma \otimes G_{\sigma_{\mathrm{s}}, \sigma_{\mathrm{s}}, \sigma_{\mathrm{r}}}$. They show that the bilateral filter output is $BF[I](p_x, p_y) = \tilde{\Gamma}\big(p_x, p_y, I(p_x, p_y)\big)$. This process is illustrated in Figure 4.1 and detailed in Table 4.3.

The benefit of this formulation is that the Gaussian-convoluted grid $GC[\Gamma]$ is a band-limited signal because it results from a Gaussian convolution with a low-pass filter. Paris and Durand use this argument to downsample the grid Γ. As a result, they deal with fewer stored data points and achieve performance on the order of one second for images with several megapixels. Chen et al. [16] further improved the performances by mapping the algorithm onto modern graphics hardware, obtaining running times on the order of a few milliseconds. Paris and Durand recommend using the Gaussian width parameters σ_{s} and σ_{r} to set the sampling rates for the 3D grid. This yields a complexity of $\mathcal{O}\left(|\mathcal{S}| + \frac{|\mathcal{S}|}{{\sigma_{\mathrm{s}}}^2}\frac{|\mathcal{R}|}{\sigma_{\mathrm{r}}}\right)$ where $|\mathcal{S}|$ is the size of the spatial domain (i.e., the number of pixels) and $|\mathcal{R}|$ is the size of the range domain (i.e., the extent of the intensity scale).

This approach can be easily adapted to cross bilateral filtering and color images. The downside is that color images require a 5D grid which

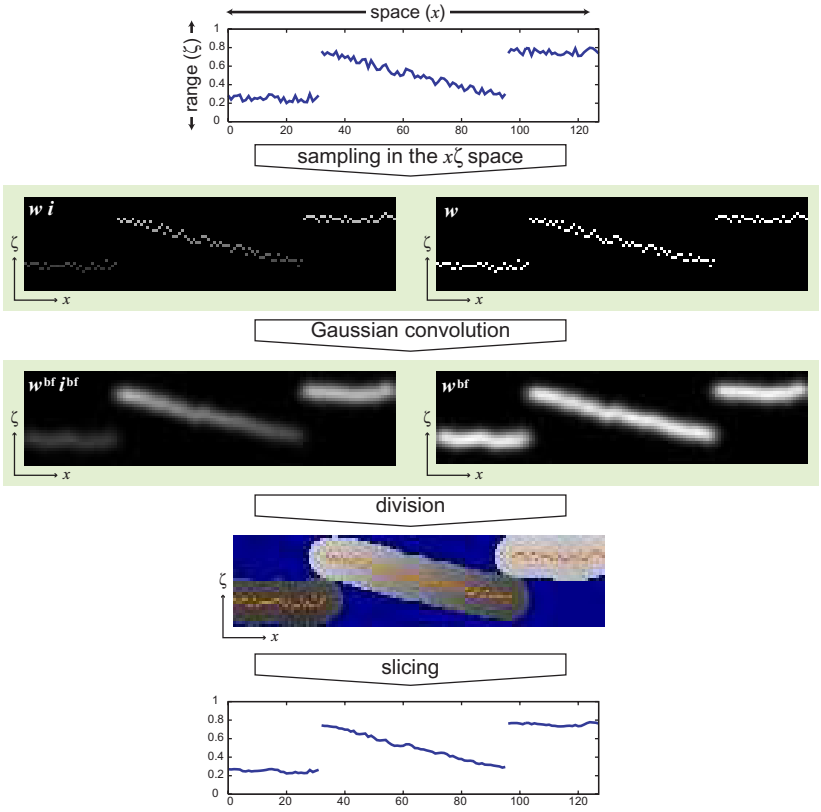


Fig. 4.1 Overview on a 1D signal of the reformulation of the bilateral filter as a linear convolution in a homogeneous, higher dimensional space. Figure reproduced from Paris and Durand [50].

no longer maps nicely onto graphics hardware and that requires large amount of memory for small kernels (10 pixels or less).

4.5.1 Link with the Layered Approximation

The bilateral grid and the layered approximation share the idea of subsampling along the intensity axis and downsampling in the spatial domain. The major difference is in the way the downsampling is performed. The layered approximation encounters difficulties at discontinuities: it averages adjacent pixels with different values, e.g., a white and a black pixel ends up being represented by one gray value that

Table 4.3 Approximation proposed by Paris and Durand [50]. In practice, localized downsampling and upsampling eliminates the need to build the entire high-resolution grid in memory.

1. Given a 2D image I, build the grid $\Gamma : \mathcal{S} \times \mathcal{R} \to \mathbb{R}^2$ that contains homogeneous values:

$$\Gamma(p_x, p_y, r) = \begin{cases} (I(p_x, p_y), 1) & \text{if } r = I(p_x, p_y) \\ (0, 0) & \text{otherwise} \end{cases}.$$

2. Downsample Γ to get $\tilde{\Gamma}$.
3. Perform a Gaussian convolution of $\tilde{\Gamma}$, for each component independently

$$GC\left[\tilde{\Gamma}\right](p_x, p_y, r) = G_{\sigma_\mathrm{s}, \sigma_\mathrm{r}} \otimes \tilde{\Gamma}(p_x, p_y, r),$$

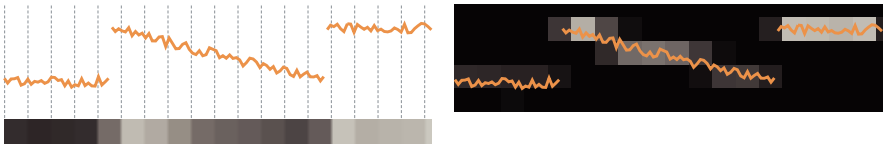
   where $G_{\sigma_\mathrm{s}, \sigma_\mathrm{r}}$ is a 3D Gaussian with σ_s as parameter along the two spatial dimensions and σ_r along the range dimension.
4. Upsample $GC\left[\tilde{\Gamma}\right]$ to get $\hat{\Gamma}$.
5. Extracting the result: For a pixel $\mathbf{p}$ with initial intensity $I\mathbf{p}$, we denote $(\widetilde{wI}, \tilde{w})$ the value at position $(p_x, p_y, I_\mathbf{p})$ in $\hat{\Gamma}$. The result of the bilateral filter is

$$BF[I]_\mathbf{p} \approx \widetilde{wI}/\tilde{w}.$$

poorly represents the original signal. In comparison, the bilateral grid subsampling strategy preserves adjacent pixels with different intensities, because they are far apart along the intensity axis. In the white and black pixels case, the bilateral grid retains the two different values involved and thus is able to produce better results. Figure 4.2 illustrates this behavior. The bilateral grid should be preferred over the layered approximation, because both approaches perform equivalently fast.

4.6 Bilateral Pyramid

For several applications such as detail enhancement [25], it is desirable to decompose the image into more than two layers. Fattal et al. [25]



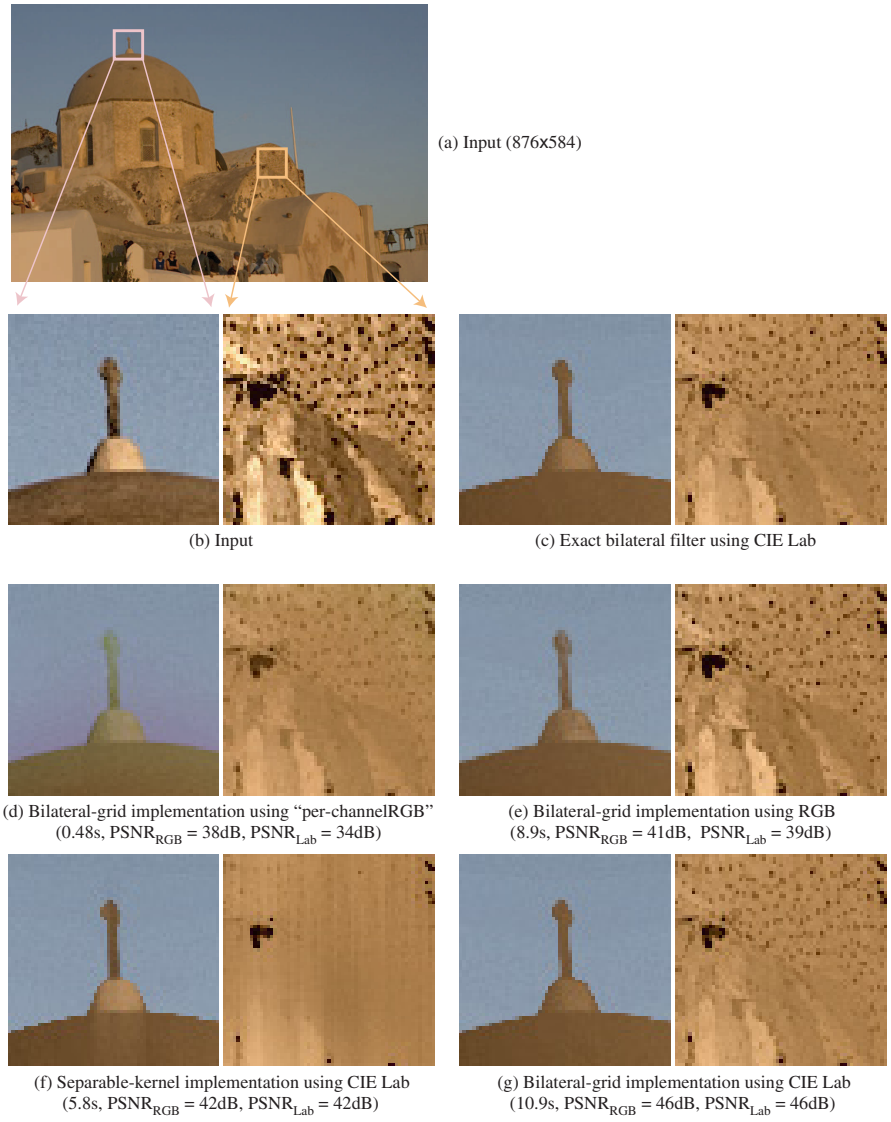
(a) Downsampling of the layered approximation (b) Downsampling of bilateral grid approximation

Fig. 4.2 Compared to the layered approximation, the bilateral grid better represents discontinuities and thus yields superior results. This figure is reproduced from Paris and Durand [50].

propose to compute such a decomposition by successively applying the bilateral filter to the image with varying parameters: the spatial parameter σ_s is doubled at each level and the range parameter σ_r is halved. Based on this scenario, they describe a dedicated numerical scheme. Intuitively, instead of computing each level "from scratch," they use the result from the previous level and rely on the fact that this image has already been smoothed to simplify the computation. For each level, they compute a bilateral filter based on a 5×5 kernel. At the first level they apply the bilateral filter with a small kernel $\sigma_s = 1$, and at each subsequent level they double the spatial extent of the kernel. A naive approach would use more coefficients, e.g., a 9×9 kernel, but Fattal et al. keep the cost constant by using 5×5 samples and inserting zeros. For instance, they approximate a 9×9 kernel using 5×5 samples interleaved with zeros, such that a $1 - 4 - 6 - 4 - 1$ row becomes $1 - 0 - 4 - 0 - 6 - 0 - 4 - 0 - 1$. This proven strategy, known as an *algorithme à trous*, yields minimal errors when applied to band-limited signals [43]. In this particular case, the signal is not

Table 4.4 Complexity summary for Bilateral Filter algorithms.

Brute force (Section 4.1)	$\mathcal{O}\left(\lvert\mathcal{S}\rvert^2\right)$
Separable kernel (Section 4.2)	$\mathcal{O}\left(\lvert\mathcal{S}\rvert\sigma_s\right)$
Local histograms (Section 4.3)	$\mathcal{O}\left(\lvert\mathcal{S}\rvert\log\sigma_s\right)$
Layered approximation (Section 4.4)	$\mathcal{O}\left(\lvert\mathcal{S}\rvert + \frac{\lvert\mathcal{S}\rvert}{\sigma_s^2}\frac{\lvert\mathcal{R}\rvert}{\sigma_r}\right)$
Bilateral grid (Section 4.5)	$\mathcal{O}\left(\lvert\mathcal{S}\rvert + \frac{\lvert\mathcal{S}\rvert}{\sigma_s^2}\frac{\lvert\mathcal{R}\rvert}{\sigma_r}\right)$



(a) Input (876x584)

(b) Input

(c) Exact bilateral filter using CIE Lab

(d) Bilateral-grid implementation using "per-channelRGB" (0.48s, $PSNR_{RGB}$ = 38dB, $PSNR_{Lab}$ = 34dB)

(e) Bilateral-grid implementation using RGB (8.9s, $PSNR_{RGB}$ = 41dB, $PSNR_{Lab}$ = 39dB)

(f) Separable-kernel implementation using CIE Lab (5.8s, $PSNR_{RGB}$ = 42dB, $PSNR_{Lab}$ = 42dB)

(g) Bilateral-grid implementation using CIE Lab (10.9s, $PSNR_{RGB}$ = 46dB, $PSNR_{Lab}$ = 46dB)

Fig. 4.3 Comparison of different strategies for filtering a color source image (a,b). Processing the red, green, and blue channels independently is fast but can cause color bleeding that removes the cross from the sky in (d). Filtering RGB vectors is slower but improves results although some bleeding remains (e). Using a perceptually motivated color space such as CIE-Lab addresses those artifacts (c,g). The separable-kernel implementation is fast but incurs axis-aligned streaks (f) that may undesirable in a number of applications. These remarks are confirmed by the numerical precision evaluated with the PSNR computed the RGB and CIE-Lab color spaces. The contrast of the close-ups has been increased for clarity purpose. This figure is reproduced from Paris and Durand [50].

band-limited because bilateral filtering preserves edges. Yet, Fattal's results show that in practice, this approximation achieves good results without visual defects.

4.7 Discussion

The choice of implementation is crucial to achieving satisfying results with good performance. Table 4.4 summarizes the complexity of the various implementations we described.

When graphics hardware is available, we recommend the bilateral grid method of Chen et al. [16], because it achieves high-quality outputs and real-time performances even on high-resolution images and videos. If only the CPU is available, the choice is split between the local-histogram method of Weiss [71] and the bilateral grid of Paris and Durand [50]. To process color images or compute a cross bilateral filter, the bilateral grid provides a satisfying solution, especially with large kernels. To process gray-level images with kernels of any size, e.g., in an image-editing package where users can arbitrarily choose the kernel size, the local-histogram approach is preferable because it consistently yields short running times. On color images, this approach can yield less satisfying results because channels are processed independently, which may cause some color bleeding (Figure 4.3).

5

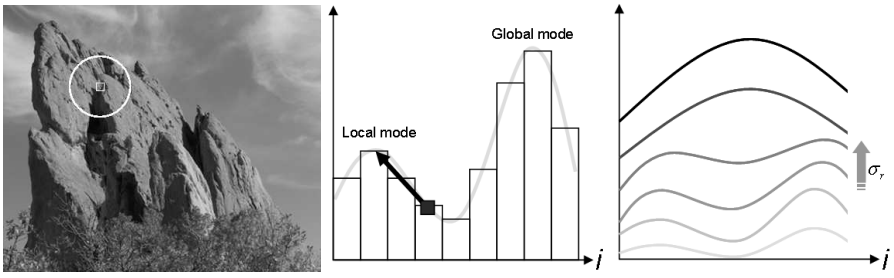
Relationship between Bilateral Filtering and Other Methods or Framework

Filtering an image while preserving its edges has been addressed in many ways in computer vision. Interestingly, some methods give results that are qualitatively very similar to those from bilateral filtering. So the natural question is to investigate what kind of relationships may exist between bilateral filtering and other existing methods. In this section we focus on local mode filtering, robust statistics and PDE-based approaches.

5.1 Bilateral Filtering is Equivalent to Local Mode Filtering

Local mode filtering was introduced by Van de Weijer and van den Boomgaard [65] as an extended filtering method to preserve edges and details. In this section, we demonstrate that the bilateral filtering is a local mode seeking approach. Based on this histogram interpretation, Weiss [71] proposed a fast numerical scheme, and Chen et al. [16] showed that the bilateral grid can be used for local histogram equalization. Refer to Section 4.3 for more details.

Given a pixel and its associated local histogram, local mode filtering is an iterative procedure which converges to the closest highest mode of



(a) (b) (c)

Fig. 5.1 (a) Image and local neighborhood for a given pixel. (b) In the local mode filtering, proposed by Van de Weijer and van den Boomgaard [65], each pixel "moves" toward the maximum of the local mode it belongs to. In this example, the intensity of the center pixel will move toward the maximum of the mode made of low-intensity pixels. (c) Effect on the local histogram of the range parameter.

the local histogram, starting from the value of the pixel at the center of the neighborhood. This is illustrated in Figure 5.1(a) and (b). Choosing the closest local mode instead of the global mode allows details to be preserved.

Like the bilateral filter, local mode filtering depends on two parameters: one which defines the neighborhood for the local histogram estimation, and one which is the smoothness parameter of the histogram. The influence of the latter parameter is illustrated in Figure 5.1(c): when the smoothing parameter increases, local modes and the global mode merge into a single global mode which corresponds to the standard Gaussian smoothed value. In that case, details are not preserved.

To define local mode filtering, given a gray-scale image $I : \Omega \to \mathcal{R}$, one can start with the definition of a histogram:

$$H^1(i) = \sum_{q \in \mathcal{S}} \delta(I_{\mathbf{q}} - i), \quad \forall i \in \mathcal{R},$$

where δ is the Dirac function so that $\delta(s) = 1$ if $s = 0$, and $\delta(s) = 0$ otherwise. A classical operation consists in smoothing histograms, so that we define:

$$H^2(i, \sigma_{\mathrm{r}}) = H^1 \otimes G_{\sigma_{\mathrm{r}}}(i) = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_{\mathrm{r}}}(I_{\mathbf{q}} - i),$$

where σ_{r} denotes the smoothing done on the intensity values, i.e., on the range. A step further, one can define an histogram locally, i.e., around a given position $\mathbf{p}$. To do it, one can introduce a spatial Gaussian kernel centered around $\mathbf{p}$:

$$H^3(\mathbf{p}, i, \sigma_{\mathrm{r}}, \sigma_{\mathrm{s}}) = \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_r}(I_{\mathbf{q}} - i), \tag{18}$$

where σ_{s} determines the spatial neighborhood around $\mathbf{p}$. Local histograms can be used to study image properties [36] but also to perform image restoration. The idea of local mode filtering is to make the intensity $I_{\mathbf{p}}$ of the center pixel evolve toward the closest local maximum. So, $I_{\mathbf{p}}$ will verify:

$$\frac{\partial H^3}{\partial i}(\mathbf{p}, i, \sigma_{\mathrm{r}}, \sigma_{\mathrm{s}})\Big|_{i=I_{\mathbf{p}}} = \; 0. \tag{19}$$

Taking into account Equation (18) and the expression of the Gaussian kernel, Equation (19) becomes:

$$\sum_{\mathbf{q} \in \mathcal{S}} (I_{\mathbf{q}} - i)\, G_{\sigma_{\mathrm{s}}}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(I_{\mathbf{q}} - i) = 0,$$

so that $I_{\mathbf{p}}$ should verify the following implicit equation:

$$\begin{aligned} I_{\mathbf{p}} &= i \text{ where } \; i \text{ is such that } i \\ &= \frac{\sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(I_{\mathbf{q}} - i) I_{\mathbf{q}}}{\sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(I_{\mathbf{q}} - i)}. \end{aligned} \tag{20}$$

To solve this implicit equation, one can propose the following iterative scheme: Given $I^0_{\mathbf{p}} = I$, estimate:

$$I^{t+1}_{\mathbf{p}} = \frac{\sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(I^t_{\mathbf{q}} - I^t_{\mathbf{p}}) I^t_{\mathbf{q}}}{\sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_{\mathrm{s}}}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_{\mathrm{r}}}(I^t_{\mathbf{q}} - I^t_{\mathbf{p}})} \;\; \text{for all } \mathbf{p}. \tag{21}$$

Interestingly, the right-hand side term of Equation (21) corresponds to the definition of the bilateral filter: Consequently, bilateral filtering can be considered as a local mode seeking method.

Remark. Another important relation established by van de Weijier and van den Boomgaard [65] is the correspondence between local

mode filtering and the framework of robust statistics. In fact, maximizing H^3 is equivalent to minimizing a residual error $\varepsilon(\mathbf{p}, i, \sigma_\mathrm{r}, \sigma_\mathrm{s}) = 1 - H^3(\mathbf{p}, i, \sigma_\mathrm{r}, \sigma_\mathrm{s})$. We explain this idea in more detail later, but focus more on the link between the bilateral filter and robust statistics (see Section 5.2).

5.2 The Bilateral Filter is a Robust Filter

Robust statistics offers a general background to model a large class of problems, including image restoration (see Ref. [30, 29, 40, 28, 27] for more details). Expressed as optimization problems in a discretized space, it is possible to define some edge-preserving restoration formulations. In this section, we show that bilateral filtering corresponds to a gradient descent of a robust minimization problem.

Image restoration can be formulated as a minimization problem in the following way: Given a noisy image I^n, the problem is to find the minimizer of the discrete energy:

$$\min_I \sum_{\mathbf{p}\in\mathcal{S}} \left((I_\mathbf{p} - I^n_\mathbf{p})^2 + \sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})} \rho(I_\mathbf{q} - I_\mathbf{p}) \right), \tag{22}$$

where $\mathcal{N}(\mathbf{p})$ is a neighborhood $\mathbf{p}$, and ρ is a weighting function (also called error norm).

Energy in Equation (22) contains two kinds of terms. The first term is a fidelity-of-attachment term which prevents the solution from drifting too far away from the noisy input values. The second term is a regularization term that will penalize differences of intensities between neighboring pixels, with a strength that depends on the function ρ. Thus the regularity of the solution will depend on function ρ. In particular, this method will be robust if we can preserve significant intensity differences such as edges, i.e., if we can distinguish the difference between inliers and outliers. Several possible ρ functions have been proposed in literature, as we are going to show in this section.

Let us now focus on the regularization term of Equation (22) to show the relationship with the bilateral filter. To do so, we introduce the following reweighted version of the regularization term, so that the

minimization problem becomes:

$$\min_{I} \sum_{\mathbf{p}\in\mathcal{S}} \sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})} G_{\sigma_\mathrm{s}}(\|\mathbf{q}-\mathbf{p}\|)\ \rho(I_\mathbf{q}-I_\mathbf{p}) \tag{23}$$

To minimize Equation (23), one can iterate the following gradient descent scheme:

$$I_\mathbf{p}^{t+1} = I_\mathbf{p}^t + \frac{\lambda}{|\mathcal{N}(\mathbf{p})|} \sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})} G_{\sigma_\mathrm{s}}(\|\mathbf{q}-\mathbf{p}\|)\ \rho'(I_\mathbf{q}^t - I_\mathbf{p}^t). \tag{24}$$

By choosing $\rho(s) = 1 - G_{\sigma_\mathrm{r}}(s)$, we obtain:

$$I_\mathbf{p}^{t+1} = I_\mathbf{p}^t + \frac{\lambda}{|\mathcal{N}(\mathbf{p})|} \sum_{\mathbf{q}\in\mathcal{N}(\mathbf{p})} G_{\sigma_\mathrm{s}}(\|\mathbf{q}-\mathbf{p}\|)\ G_{\sigma_\mathrm{r}}(I_\mathbf{q}^t - I_\mathbf{p}^t)(I_\mathbf{q}^t - I_\mathbf{p}^t). \tag{25}$$

This equation has in fact some similarities with the bilateral filtering expression, which corresponds to a weighted average of the data, that we remind here:

$$I_\mathbf{p}^{t+1} = \frac{\sum_\mathbf{q} G_{\sigma_\mathrm{s}}(\|\mathbf{q}-\mathbf{p}\|)\ G_{\sigma_\mathrm{r}}(I_\mathbf{q}^t - I_\mathbf{p}^t) I_\mathbf{q}^t}{\sum_\mathbf{q} G_{\sigma_\mathrm{s}}(\|\mathbf{q}-\mathbf{p}\|)\ G_{\sigma_\mathrm{r}}(I_\mathbf{q}^t - I_\mathbf{p}^t)} \tag{26}$$

and, interestingly, it has been shown that Equations (24) and (26) are indeed two equivalent ways to solve the same minimization approach (see, e.g., [29]). Intuitively, one can remark that both formulas average the same pixels using the same weights, and the only difference is the weight of the center pixel $I_\mathbf{p}^t$. The conclusion is that the bilateral filter is a special case of a robust filter.

More generally, Durand and Dorsey [21] studied the bilateral filter in the framework of robust statistics [29, 30] in a similar manner as the work of Black et al. [11] on PDE filters. The authors showed that the range weight can be seen as a robust metric, that is, it differentiates between *inliers* and *outliers*. The bilateral filter replaces each pixel by a weighted average of its neighbors. The weight assigned to each neighbor determines its influence on the result and is crucial to the output quality. In this context, robust statistics estimates if a pixel is relevant, i.e., is an inlier, or if it is not, i.e., is an outlier. The strategy followed by the bilateral filter is that pixels with different intensities

are not related and should have little influence on each other, whereas pixels with similar intensities are closely related and should strongly influence each other. The way that this intensity difference actually contributes is defined by the range weight. The most common choice is a Gaussian function G_{σ_r}.

However, Durand and Dorsey [21] have underscored that this Gaussian function is only one of the possible choices among a variety of robust weighting functions (cf. Figure 5.2-top), *a.k.a.* stopping functions. These functions define the weights assigned to a pixel according to its difference of intensity with the center pixel. For instance, a classical non-robust mean assigns the same weight to all pixels. In comparison, robust functions have a bell profile that assign lower weights to pixels with a different intensity. The differences lie in the fall-off rate which defines how narrow is the transition between inliers and outliers, and in the tail value: either non-zero, meaning that outliers still have some limited influence, or zero, meaning that outliers are completely

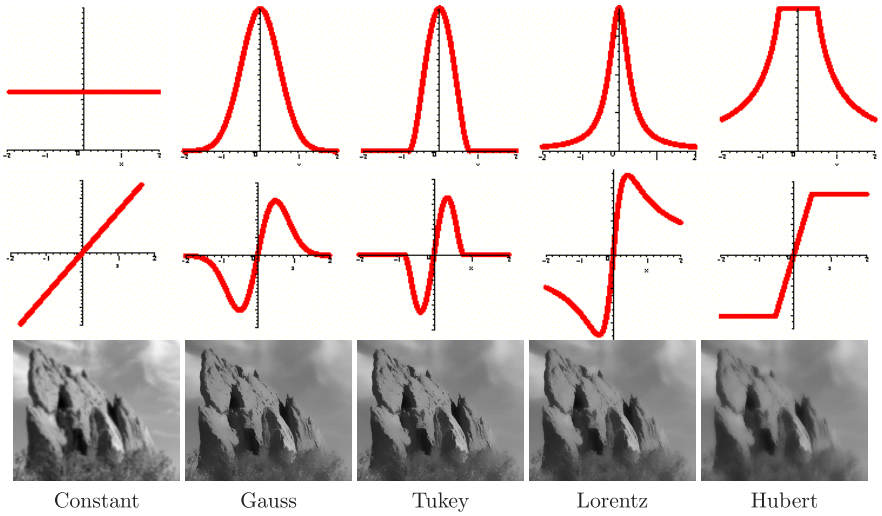


Fig. 5.2 Qualitative illustration of the influence of weighting functions for image restoration. The first two rows show respectively different choices of weighting functions ρ and their corresponding influence functions ρ'. These graphs were adapted from Black et al. [11]; Durand and Dorsey [21]. Last rows show results obtained on the image presented in Figure 5.1 with the corresponding weighting functions.

ignored. This behavior is better observed on the *influence function* (Figure 5.2-bottom) that shows the variations of the output depending on the pixel intensity. The constant weight of classical averaging is not robust because its influence function is unbounded which reflects the fact that a single pixel can have an unlimited influence on the mean value, e.g., a single very bright pixel can make the average arbitrarily high. In contrast, robust influence functions are bounded, showing that a single pixel cannot modify the output beyond a certain point. Some robust functions such as the Gauss, Tukey, and Lorentz functions are even redescending, reflecting the fact that pixels with a large intensity difference are considered "irrelevant" and ignored, i.e., they have no influence on the output.

Durand and Dorsey [21] showed that these concepts can be applied to the bilateral filter and that the choice of the range function defines how the pixels across are handled (see some results in Figure 5.2). For instance, with the classical Gaussian function, pixels across edges still have some influence though very limited; with a Tukey function, these pixels would be ignored. However, according to Durand and Dorsey's experiments, the Gauss and Tukey functions perform better for their tone-mapping operator. As far as we know, these options have not been tested with other applications.

The energy function defined by robust norms is usually not convex and can lead to local minima, similar to the local modes of histograms discussed above. Which local minimum is most desirable depends on the application. The bilateral filter performs one step toward the minimum closest to the input value. This is usually desirable because most applications seek to smooth low-amplitude noise while retaining local structure. However, some cases might require a different treatment, such as impulse noise where the value of a pixel can be severely corrupted. In this case, the robust statistics literature advocates initialization with an estimator that is very robust but might not be very precise, such as the median. For impulse noise removal, a median filter can be used to steer the bilateral filter at a pixel toward a local minimum that is consistent with its neighbors. In practice, this involves computing the range Gaussian based on the difference between a pixel and the median-filtered image rather

than the difference with the input pixel value. See Section 3.1.4 for detail.

Remark. As connections can be established between robust statistics and nonlinear PDEs, then we have also the same interpretations of bilateral filtering as a "robust" nonlinear operator in the continuous framework of PDEs. This is further explained in Section 5.3.

5.3 Bilateral Filtering is Equivalent Asymptotically to the Perona and Malik Equation

Bilateral filtering smooths an image while preserving strong edges. Interestingly, many research projects were carried out in the field of nonlinear partial differential equations (PDEs) to achieve the same goal, and some models such as [52] give results very similar to bilateral filtering. In this section we revisit several contributions showing the links between bilateral filtering and Perona–Malik model in the discrete setting, and more generally between neighborhood filters and PDE-based approaches in the continuous setting.[1] Of course, the field of PDE-based approaches is very large and one may find better approaches than bilateral filtering depending on the application. Intensive research has been carried out in this area, including nonlinear approaches for image restoration (we refer to [2] for a review). Here we focus on the nonlinear operators that are related to bilateral filtering.

5.3.1 Results in the Discrete Setting

Anyone studying PDE-based approaches for image processing came across the famous nonlinear one by Perona and Malik [52]. Starting from the heat equation and based on the remark that $\triangle I = \mathrm{div}(\nabla I)$, the authors proposed to introduce a weighting coefficient depending on

[1] Until now, we considered an image as a discrete set of pixels. Instead, in this section, we will need to consider an image defined continuously, i.e., an analog image where space is no longer discretized. The motivation becomes clear when one needs for instance to define a notion of derivative. Formally, keeping the same notations, this introduces only minor changes in the formulation of the bilateral filter. The only difference here is that sums are replaced by integrals: Positions $\mathbf{p}$ and $\mathbf{q}$ now vary on a continuous domain.

the image gradient to prevent edges to be smoothed. Their model is written in the continuous setting:

$$\frac{\partial I}{\partial t} = \operatorname{div}\big(c(\|\nabla I\|^2)\,\nabla I\big), \tag{27}$$

where $c : [0, +\infty[\rightarrow]0, +\infty]$ is a smooth decreasing function. We refer to Perona and Malik [52] for more details.

In the discrete setting, Durand and Dorsey [21] showed that if the bilateral filter is restricted to the four adjacent neighbors of each pixel, then it actually corresponds to a discrete version of the Perona and Malik [52] model.

This result has been extended by Elad [23] and Barash and Comaniciu [7] who have demonstrated that the bilateral filter can be seen as the sum of several Perona–Malik filters at different scales, that is, the image derivatives are computed with pixels at a distance, not only with adjacent pixels.

5.3.2 Results in the Continuous Setting

Another important contribution came from Buades et al. [13] who proved rigorously that for "small neighborhoods", the Yaroslavsky filter:

$$Y_{\sigma_\mathrm{s},\sigma_\mathrm{r}}[I](\mathbf{p}) = \frac{1}{W(\mathbf{p})} \int_{B(\mathbf{p},\sigma_\mathrm{s})} \exp\left(-\frac{|I(\mathbf{q}) - I(\mathbf{p})|^2}{{\sigma_\mathrm{r}}^2}\right) I(\mathbf{q}) d\mathbf{q}, \tag{28}$$

i.e., a bilateral filter using a box function as spatial weight, behaves as the Perona–Malik filter. Such a result can only be established locally, that is when "small neighborhoods" are considered, because the action of PDE is very local (local structure is taken into account through derivatives). So the proof of Buades et al. is based on an asymptotic study which relies on the fact that the image is well approximated by its second-order Taylor expansion; their result holds for any neighborhood as long as it covers a sufficiently regular area such as a region of skin or sky.

In this section, we present the results by Buades et al. [13] who revisited the notion of the bilateral filter by studying the more general "neighborhood filter" (see also [2] for more details). Here the notion of

neighborhood must be understood broadly: neighboring pixels, neighboring or similar intensities, or "neighboring neighborhoods." Each of these meanings will correspond to a specific filter. Interestingly, the authors also proved the link between these filters and well-known PDEs such as the heat equation and the Perona–Malik equation.

A general neighborhood filter can be described as follows. Let I be an image to be filtered or denoised and let $w_{\sigma_\mathrm{s}} : \mathbb{R}^+ \to \mathbb{R}^+$ and $w_{\sigma_\mathrm{r}} : \mathbb{R}^+ \to \mathbb{R}^+$ be two functions whose roles will be to enforce, respectively, photometric and geometric locality (in Section 2, w_{σ_s} and w_{σ_r} are both Gaussian kernels). Parameters σ_s and σ_r will measure the amount of filtering for the image I. The filtered image at scale $(\sigma_\mathrm{r}, \sigma_\mathrm{s})$ is given by:

$$BF[I](\mathbf{p}) = \frac{1}{W(\mathbf{p})} \int_{\mathcal{S}} w_{\sigma_\mathrm{r}}(|I(\mathbf{q}) - I(\mathbf{p})|)\; w_{\sigma_\mathrm{s}}(\|\mathbf{p} - \mathbf{q}\|) I(\mathbf{q}) d\mathbf{q},$$

where $W(\mathbf{p})$ is a normalization factor.

$$W(\mathbf{p}) = \int_{\mathcal{S}} w_{\sigma_\mathrm{r}}(|I(\mathbf{q}) - I(\mathbf{p})|)\; w_{\sigma_\mathrm{s}}(\|\mathbf{p} - \mathbf{q}\|) d\mathbf{q}.$$

For simplicity we suppose that the image has been extended from the domain image $\mathcal{S}$ (a rectangle) to the whole of $\mathbb{R}^2$, by symmetry and periodicity.

With this formalism we can easily recover the classical spatial linear Gaussian filtering by choosing $w_{\sigma_\mathrm{r}} \equiv 1$ and $w_{\sigma_\mathrm{s}}(t) = \exp\big(-\frac{t^2}{{\sigma_\mathrm{s}}^2}\big)$.

Now let us consider bilateral filtering. As mentioned before, the idea is to take an average of the values of pixels that are both close in gray level value and spatial distance. Of course many choices are possible for the kernels w_{σ_r} and w_{σ_s}. Classical choices are:

$$w_{\sigma_\mathrm{r}}(t) = \exp\left(-\frac{t^2}{{\sigma_\mathrm{r}}^2}\right),$$

and

$$w_{\sigma_\mathrm{s}}(t) = \exp\left(-\frac{t^2}{{\sigma_\mathrm{s}}^2}\right) \quad \text{or} \quad w_{\sigma_\mathrm{s}}(t) = \chi_{B(\mathbf{p},\sigma_\mathrm{s})}(t),$$

where $\chi_{B(\mathbf{p},\sigma_\mathrm{s})}$ denotes the characteristic function of the ball of center $\mathbf{p}$ and radius σ_s. With the former choice of w_{σ_s}, we get the SUSAN filter [59] or the bilateral filter [63] (see Section 2). With the latter

choice of w_{σ_s}, we recover the Yaroslavsky filter defined in Equation (28). The SUSAN and Yaroslavsky filters have similar behaviors. Inside a homogeneous region, the gray level values slightly fluctuate because of the noise. Near sharp boundaries between a dark and a bright region, both filters compute averages of pixels belonging to the same region as the reference pixel: edges are not blurred.

Interestingly, the estimation of the residue $I_{\sigma_s,\sigma_r}(\mathbf{p}) - I(\mathbf{p})$ gives some analogies with well-known PDEs. This is more precisely stated in the following theorem.

Theorem 5.1. Suppose $I \in \mathcal{C}^2(\mathcal{S})$ and let σ_s, σ_r, and $\alpha > 0$ such that

$$\sigma_s, \sigma_r \to 0 \quad \text{and} \quad \sigma_r = O({\sigma_s}^{\alpha}).$$

Let us consider the continuous functions $g(t) = \frac{1}{3}\frac{t\exp(-t^2)}{E(t)}$ for $t \neq 0$, $g(0) = \frac{1}{6}$ where $E(t) = \int_0^t \exp(-s^2)ds$, and $f(t) = 3g(t) + 3\frac{g(t)}{t^2} - \frac{1}{2t^2}$, for $t \neq 0$ and $f(0) = \frac{1}{6}$.

Then for $x \in \mathcal{S}$,

- if $\alpha < 1$, $Y_{\sigma_s,\sigma_r}[I](\mathbf{p}) - I(\mathbf{p}) \approx \frac{\Delta I(\mathbf{p})}{6}{\sigma_s}^2$,
- if $\alpha = 1$, $Y_{\sigma_s,\sigma_r}[I](\mathbf{p}) - I(\mathbf{p}) \approx \big[g(\frac{\sigma_s}{\sigma_r}\|\nabla I(\mathbf{p})\|)I_{TT}(\mathbf{p}) + f(\frac{\sigma_s}{\sigma_r}\|\nabla I(\mathbf{p})\|)\, I_{NN}(\mathbf{p})\big]{\sigma_s}^2$,
- if $1 < \alpha < \frac{3}{2}$, $Y_{\sigma_s,\sigma_r}[I](\mathbf{p}) - I(\mathbf{p}) \approx g({\sigma_s}^{1-\alpha}\|\nabla I(\mathbf{p})\|)\big[I_{TT}(\mathbf{p}) + 3I_{NN}(\mathbf{p})\big]{\sigma_s}^2$,

where $I_{TT} = D^2u\left(\frac{\nabla I^{\perp}}{\|\nabla I\|}, \frac{\nabla I^{\perp}}{\|\nabla I\|}\right)$ and $I_{NN} = D^2u\left(\frac{\nabla I}{\|\nabla I\|}, \frac{\nabla I}{\|\nabla I\|}\right)$.

We refer to [13, 12] for the proof of the theorem. It is not difficult, somewhat technical, and relies on a Taylor expansion of $I(\mathbf{q})$ and the exponential function.

More interesting is the interpretation of this theorem. For α ranging from 1 to $\frac{3}{2}$ an iterated procedure of the Yaroslavsky filter behaves asymptotically as an evolution PDE involving two terms respectively proportional to the direction $T = \frac{\nabla I^{\perp}(\mathbf{p})}{\|\nabla I(\mathbf{p})\|}$, which is tangent to the level passing through $\mathbf{p}$ and to the direction $N = \frac{\nabla I(\mathbf{p})}{\|\nabla I(\mathbf{p})\|}$, which is orthogonal

to the level passing through $\mathbf{p}$. In fact, we may write:

$$\frac{Y_{\sigma_s,\sigma_r}[I](\mathbf{p}) - I(\mathbf{p})}{{\sigma_s}^2} = c_1 I_{TT} + c_2 I_{NN}.$$

The filtering or enhancing properties of the model depend on the sign of c_1 and c_2. Following Theorem 5.1, we have:

- If $\alpha < 1$, then $\frac{Y_{\sigma_s,\sigma_r}[I](\mathbf{p})-I(\mathbf{p})}{{\sigma_s}^2} \approx \frac{\Delta I(\mathbf{p})}{6}$, which corresponds to a Gaussian filtering.
- If $\alpha = 1$, the neighborhood filter acts as a filtering/enhancing algorithm. As the function g is positive (and decreasing) there is always a diffusion in the tangent direction, but because the function I can take positive or negative values, we may have filtering/enhancing effects depending on the values of $\|\nabla I(\mathbf{p})\|$. For example, if $\|\nabla I(\mathbf{p})\| > a\frac{\sigma_r}{\sigma_s}$, where a is such that $I(a) = 0$, then we get an enhancing effect. Let us remark that because $g(t) \to 0$ as $t \to \infty$, points with large gradient are preserved.
- If $1 < \alpha < \frac{3}{2}$, then $\frac{\sigma_s}{\sigma_r}$ tends to infinity and $g(\frac{\sigma_s}{\sigma_r}\|\nabla I\|)$ tends to zero and consequently the original image is hardly deteriorated.

Finally, let us observe that when $\alpha = 1$, the Yaroslavsky filter behaves asymptotically like the Perona–Malik (Equation (27)) which can be rewritten as:

$$\frac{\partial I}{\partial t} = \operatorname{div}(c(\|\nabla I\|^2)\nabla I) = c(\|\nabla I\|^2) I_{TT} + b(\|\nabla I\|^2) I_{NN}, \tag{29}$$

where $b(t) = 2t\,c'(t) + c(t)$. By choosing $c(t) = g(\sqrt{t})$ in (29) we get:

$$\frac{\partial I}{\partial t} = g(\|\nabla I\|^2) I_{TT} + h(\|\nabla I\|^2) I_{NN},$$

with $h(t) = g(t) + t\,g'(t)$. We have $h(t) \neq f(t)$ but the coefficients in the tangent direction for the Perona–Malik equation and the Yaroslavsky filter are equal, and the functions h and f have the same behavior. This explains why the bilateral filter and Perona–Malik models share the same qualitative properties.[2]

[2] In particular, both suffer from shock formation, *a.k.a.* over-sharpening, that creates aliased edges from smooth ones. In Section 6.1.3 we will show neighborhood filter can be extended to avoid this effect.

Remark. The weight defined in the bilateral filter is inversely proportional to the total distance (both in space and range, see also Figure 4.1) from the center sample. This idea is also similar in spirit to the "Beltrami flow" proposed by Sochen et al. [61]. There, the effective weight is the "geodesic distance" between the samples. More precisely, the authors introduced the notion of image manifolds where an image I is represented by a manifold M embedded in $\mathcal{S} \times \mathcal{R}$:

$$(p_x, p_y) \in \mathcal{S} \mapsto M(p_x, p_y) \quad = \quad \big(p_x, p_y, I(p_x, p_y)\big) \in \mathcal{S} \times \mathcal{R} \qquad (30)$$

With this representation, Barash [6] and Barash and Comanieiu [7] demonstrated that bilateral filtering is based on the Euclidean distance of $\mathcal{S} \times \mathcal{R}$ instead of the manifold geodesic distance. Note that Paris and Durand [50] used a similar metric but in a signal-processing context (see Section 4.5). Sochen et al. [60] have also shown that bilateral filtering is an approximation to Gaussian filtering using the geodesic metric (i.e., using distances measured on the image manifold M) when the Gaussian kernel is small.

6

Extensions of Bilateral Filtering

This section describes two main extensions of the bilateral filter. First, variants have been developed to better handle gradients by taking the slope of the signal into account or avoid the staircase effect (Section 6.1). Second, bilateral filtering has been extended to handle several images to better control the way edges are detected (Section 6.2).

6.1 Accounting for the Local Slope

Humans consistently identify at least three visually distinctive image features as edges or boundaries: a sharp, step-like intensity change, a sharp, ridge- or valley-like gradient change, or both. The bilateral filter is particularly good at preserving step-like edges, because the *range* domain $\mathcal{R}$ filter averages together all similar values within the neighborhood *space* domain, and also assigns tiny weights to dissimilar values on the opposite side of the step, as shown in Figure 2.2 helps maintain the step-like changes without smoothing.

Several researchers [14, 19, 23] have proposed extensions to the bilateral filter to improve edge-preserving results for ridge- and valley-like

edges as well. As explained by Elad [23] most noted that the bilateral filter smooths images toward a piecewise constant intensity approximation of the original signal, and instead, each proposes smoothing toward piecewise constant-gradient (or low curvature) results instead.

6.1.1 Trilateral Filter

Sharp changes in gradients and large, high-gradient areas degrade the desirable smoothing abilities of the bilateral filter. As shown for one image scan-line in Figure 6.1(b), we can approximate the extent of the combined spatial and range filters as a rectangle centered around each input pixel: position within this rectangle sets the weight assigned to all its neighboring pixels. At ridge- or valley-like edges, gradients change abruptly but intensities do not, as shown in Figure 6.2 feature (1).

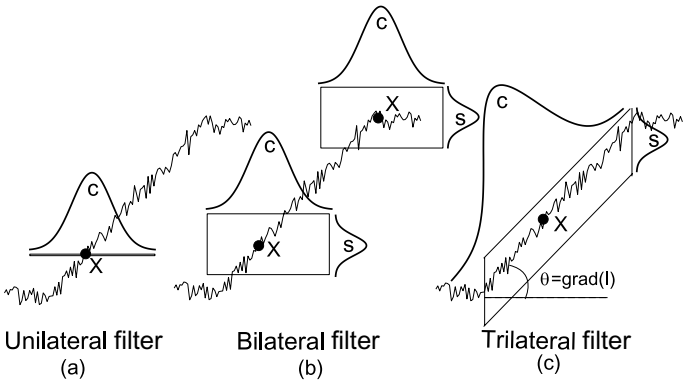


Fig. 6.1 Filter extent for one scan-line of an image.

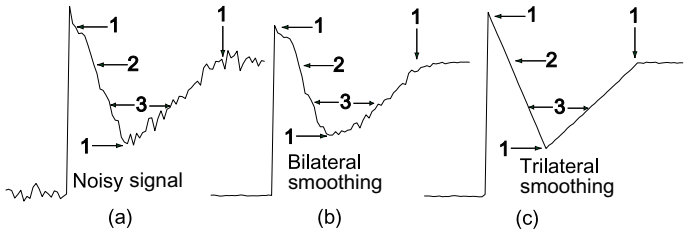


Fig. 6.2 Difficult image features: (1) Ridge-like and valley-like edges, (2) high-gradient regions, (3) similar intensities in disjoint regions.

Applying the Bilateral filtering here is troublesome, because the rectangular filter extent encloses pixels that span the peak of the ridge or valley, and the filter blends these intensities to form a "blunt" feature instead of the sharp, clean edge with disjoint gradients. High-gradient regions between ridge- or valley-like edges also reduce the bilateral filter's effectiveness. As shown in Figure 6.1(b) and Figure 6.2 feature (2) the spatial filter extent (the box width) has little effect, as only a narrow portion of the input signal falls within the box, and the range filter's extent (box height) dominates the filtering. Figure 6.2 feature (3) also shows that applying the bilateral filter near sharply peaked valley- or ridge-like features may permit the spatial extent (box width) to include disjoint portions of the input signal, averaging together image regions that may belong to unrelated objects in the image.

The trilateral filter introduced by Choudhury and Tumblin [19] addressed these problems by combining modified bilateral filters with a pyramid-based method to limit filter extent. First, they applied a bilateral filter to the image gradients to help estimate the slopes any separate image regions. Using these slopes, they "tilt" the filter extent of a bilateral filter applied to image intensity; this affine transform of the range filter, as shown in Figure 6.1(c), restores the effectiveness of the spatial filter term. Finally, for each output pixel, they limit the extent of this tilted bilateral filter to a connected set of pixels that share similar filtered-gradient values. To reduce the substantial computing time required to find these connected components, they describe a pyramid-like structure suitable for fast evaluation. They also automatically set all but two of the parameters of their filtering method, so that the user control resembles the bilateral filter's two parameters. Figure 6.3 shows

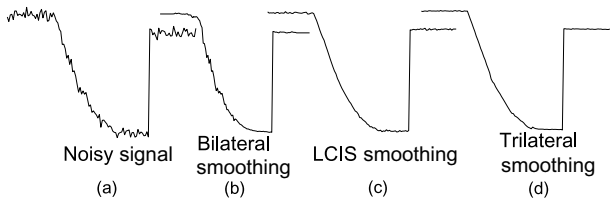


Fig. 6.3 Large, smoothly varied gradients can cause "stair-stepping" in isotropic diffusion, and weak smoothing in the bilateral filter. Higher order PDEs (e.g., LCIS) and bilateral variants that smooth toward piecewise linear results form stairsteps in gradients instead.

Fig. 6.4 Sample tone mapping results obtained with the trilateral filter (top two rows) and sample mesh denoising (bottom row).

comparisons between the trilateral filter and other approaches. When applied to tone mapping or mesh fairing, the trilateral filter results in Figures 6.4 are visibly comparable or better than the bilateral filter alone, but these improvements come at a high computational cost.

6.1.2 Symmetric Bilateral Filter

Elad [23] proposes to account for the image "slope" by comparing the intensity of the filtered pixel with the average of another pixel and its symmetric point:

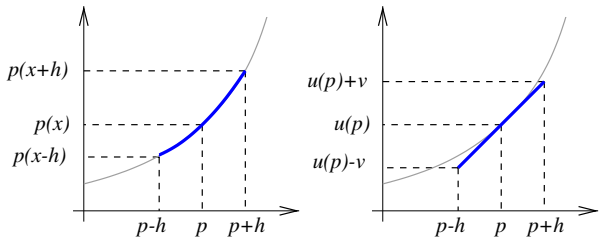
$$SBF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q}\in\mathcal{S}} G_{\sigma_\mathrm{s}}(\|\mathbf{p}-\mathbf{q}\|)\; G_{\sigma_\mathrm{r}}(\|I_{\mathbf{p}} - I_\mathrm{s}\|)\; I_\mathrm{s}, \tag{31}$$

where I_s is the average between the pixel $\mathbf{q}$ and its symmetric with respect to $\mathbf{p}$, that is: $I_\mathrm{s} = \frac{1}{2}\big(I(\mathbf{q}) + I(2\mathbf{p}-\mathbf{q})\big)$. As far as we know, the performance of this extension is unclear because it has not been extensively tested.

6.1.3 Regression Filter

The origin of the staircase effect can be explained with a 1-D convex increasing signal (respectively a 1-D increasing concave signal) (Figure 6.5). For each p, the range of points q such that $I(p) - h < I(q) \leq I(p)$ is larger (respectively smaller) than the number of points satisfying $I(p) \leq I(q) \leq I(p) + h$. Thus, the average value $Y_{\sigma_\mathrm{s},h}$ is smaller (respectively larger) than $I(p)$. As edges correspond to inflection points (i.e., points where $I'' = 0$) the signal is enhanced there; the discontinuities become more marked.

To overcome this difficulty, Buades et al. [14] introduced an intermediate regression correction to better approximate the signal locally.



(a) Yaroslavsky filter (b) Linear regression

Fig. 6.5 Why the Yaroslavsky filter (and similarly the bilateral filter) creates stepwise functions: (a) The 1D illustrations show that for each p, the range of points q such that $I(p) - h < I(q) \leq I(p)$, is larger than the number of points satisfying $I(p) \leq I(q) \leq I(p) + h$. Thus the average will be biaised. (b) This can be avoided with a locally linear approximation.

For every $\mathbf{p}$ in a 2D image, one searches for a triplet (a, b, c) minimizing:

$$\int_{B(\mathbf{p},\sigma_\mathrm{s})} w(\mathbf{p},\mathbf{q})(I(\mathbf{q}) - aq_1 - bq_2 - c)^2 d\mathbf{q}, \tag{32}$$

where $w(\mathbf{p},\mathbf{q}) = \exp\frac{-|I(\mathbf{q})-I(\mathbf{p})|^2}{{\sigma_\mathrm{r}}^2}$, and then replacing $I(\mathbf{p})$ by $(ap_x - bp_y - c)$. Let us denote this improved version of the original Yaroslavsky filter (see also Section 5.3.2) by $L_{\sigma_\mathrm{s},\sigma_\mathrm{r}}$.

Theorem 6.1. Suppose $I \in C^2(\mathcal{S})$, and let $\sigma_\mathrm{s}, h > 0$ be such that $\sigma_\mathrm{s}, \sigma_\mathrm{r} \to 0$ and $O(\sigma_\mathrm{s}) = O(\sigma_\mathrm{r})$. Let g be the continuous function defined by $g(0) = \frac{1}{6}$ and $g(t) = \frac{8t^2e^{-t^2} - 8te^{-t^2}E(t) + 2E(t)^2}{t^2(4E(t)^2 - 8te^{-t^2}E(t))}$, for $t \neq 0$, where $E(t) = \int_0^t e^{-s^2} ds$. Then:

$$L_{\sigma_\mathrm{s},\sigma_\mathrm{r}}[I](\mathbf{p}) - I(\mathbf{p}) \approx \left[\frac{1}{6} I_{\xi\xi} + g\left(\frac{\sigma_\mathrm{s}}{\sigma_\mathrm{r}} \|DI\|\right) I_{\eta\eta}\right] {\sigma_\mathrm{s}}^2. \tag{33}$$

According to Theorem 6.1, the enhancing effect has disappeared; the coefficient in the normal direction is now always positive and decreasing. When the gradient is large, the weighting function in the normal direction tends to zero and the image is filtered only in the tangent direction. Figure 6.6 shows how regression can improve the results.

6.2 Using Several Images

6.2.1 Cross and Joint Bilateral Filter

Eisemann and Durand [22] and Petschnigg et al. [53] introduced simultaneously the *cross bilateral filter*, also known as the *joint bilateral filter*, a variant of the bilateral filter that decouples the notion of edges to preserve from the image to smooth. Given an image I, the cross bilateral filter smooths I while preserving the edges of a second image E. In practice, the range weight is computed using E instead of I:

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

$$\text{with} \quad W_\mathbf{p} = \sum_{\mathbf{q}\in\mathcal{S}} G_{\sigma_\mathrm{s}}(\|\mathbf{p}-\mathbf{q}\|)\, G_{\sigma_\mathrm{r}}(E_\mathbf{p} - E_\mathbf{q}).$$

Figure 6.7 shows a simple use of cross bilateral filter to filter a low-light picture.

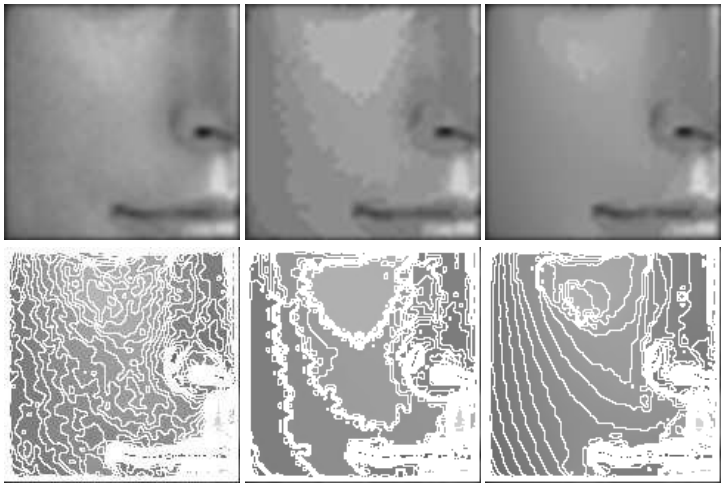


Fig. 6.6 The staircase effect can be eliminated with regression (see Section 6.1.3). First row shows the results. Second rows represents the level lines giving a clear representation of the image smoothness degree. Figure reproduced from Buades et al. [14].

Fig. 6.7 Simple example of cross bilateral filtering. The low-light image (a) is too noisy to yield satisfying result if filtered on its own with bilateral filtering, see result in (b). Using a flash picture of the same content (c) and cross bilateral filtering produces a better result (d). Eisemann and Durand [22] and Petschnigg et al. [53] propose more sophisticated techniques to handle this flash/no-flash scenario. Figure reproduced from Paris and Durand [50].

6.2.2 Dual Bilateral Filter

Bennett et al. [9] introduced *dual bilateral filtering* as a variant of bilateral filtering and cross bilateral filtering. As the cross bilateral filter, the dual bilateral filter takes two images I and J as input and filters I_1. The difference is that both I and J are used to define the

edges whereas the cross bilateral filter uses only J. The dual bilateral is defined by:

$$DBF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_I}(\|I_{\mathbf{p}} - I_{\mathbf{q}}\|)\, G_{\sigma_J}(\|J_{\mathbf{p}} - J_{\mathbf{q}}\|)\, I_{\mathbf{q}} \tag{34}$$

The advantage of this formulation is any edge visible in I or J is taken into account. Bennett et al. have demonstrated the usefulness of this strategy in the context of low-light imaging where I is a classical RGB video stream and J comes infrared cameras. The infrared camera captures more edges but lacks the colors of a standard RGB camera. In this context, the strength of dual bilateral filtering is that the noise properties of I and J can be accounted for separately by setting σ_I and σ_J independently.

From a formal standpoint, the dual bilateral filter can be interpreted as a "normal" bilateral filter based on extended range data (I, J), that is, the channels of I are "glued" to those of J to form a single image with more channels. The range weight is then a classical one except that it involves higher dimensional data. A minor difference with the formulation of Bennett et al. is that the J data are filtered as well, but one can discard them if needed to obtain the exact same result.

7

Conclusions

We have presented the bilateral filter, its applications, its variants, reviewed our current theoretical understanding of it, and explained fast algorithms to evaluate it. We believe that the success of the bilateral filter lies in its combination of simplicity, good results, and efficient algorithms. Although alternatives exist for each of these points, few, if any, combine all these advantages.

The filter is very flexible because the range weight can be adapted to accommodate any notion of pixel value difference, including arbitrary color spaces, data from other images, or any information about the relevance of one pixel to another pixel.

The original goal of the filter was denoising, in which case a small spatial kernel suffices and the residual of the filter is discarded as the noise component. In contrast, many new applications leverage the bilateral filter to create two-scale decompositions that rely on large spatial kernels and where the residual of the filter is preserved because it is much more relevant to the human visual system. The use of large spatial support has motivated a variety of accelerations schemes and the bilateral filter can now be applied in real time to large inputs.

Our review of bilateral filtering highlights several avenues for future research. Although the simple edge model based on color difference that subsumes the bilateral filter is often sufficient, there is room for a better characterization of the important contours to be preserved. The bilateral filter is also often used to extract the texture of an image. This is another direction where a more sophisticated model of what is texture would be beneficial. On the theoretical side, while the link with PDEs is well understood when the spatial kernel is shrunk to zero, the full implications of large spatial supports deserve more attention. While efficient implementations exist, they are often limited to a low-dimensional range, and the extension of these techniques to higher dimensional data is an exciting challenge. The bilateral filter is most often employed to yield a two-scale decomposition, but fully multiscale approaches deserve more investigation because the interplay between the spatial and range terms make such definitions non-trivial.

Acknowledgments

The work of Sylvain Paris at MIT and Frédo Durand 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," and a grant from Royal Dutch/Shell Group. Frédo Durand acknowledges a Microsoft Research New Faculty Fellowship and a Sloan Fellowship. Jack Tumblin's work was supported in part by National Science Foundation grants NSF-IIS 0535236 and NSF-SGER 0645973. He also acknowledges and thanks Adobe Systems, Inc. and Mitsubishi Electric Research Laboratories (MERL) for their support of his research by two unrestricted gifts to support research on topics in computational photography.

References

[1] M. Aleksic, M. Smirnov, and S. Goma, "Novel bilateral filter approach: Image noise reduction with sharpening," in *Proceedings of the Digital Photography II Conference*, volume 6069, SPIE, 2006.

[2] G. Aubert and P. Kornprobst, *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (Second edition)*. volume 147 *of Applied Mathematical Sciences*. Springer-Verlag, 2006.

[3] J. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle, "Image decomposition into a bounded variation component and an oscillating component," *Journal of Mathematical Imaging and Vision*, vol. 22, no. 1, January 2005.

[4] V. Aurich and J. Weule, "Non-linear gaussian filters performing edge preserving diffusion," in *Proceedings of the DAGM Symposium*, pp. 538–545, 1995.

[5] S. Bae, S. Paris, and F. Durand, "Two-scale tone management for photographic look," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 637–645, Proceedings of the ACM SIGGRAPH conference, 2006.

[6] D. Barash, "A fundamental relationship between bilateral filtering, adaptive smoothing and the nonlinear diffusion equation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 844–847, 2002.

[7] D. Barash and D. Comaniciu, "A Common framework for nonlinear diffusion, adaptive smoothing, bilateral filtering and mean shift," *Image and Video Computing*, vol. 22, no. 1, pp. 73–81, 2004.

[8] B. E. Bayer, "Color imaging array," US Patent 3971065, 1976.

[9] E. P. Bennett, J. L. Mason, and L. McMillan, "Multispectral bilateral video fusion," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1185–1194, May 2007.

[10] E. P. Bennett and L. McMillan, "Video enhancement using per-pixel virtual exposures," *ACM Transactions on Graphics*, vol. 24, no. 3, pp. 845–852, Proceedings of the ACM SIGGRAPH conference, July, 2005.

[11] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 421–432, March 1998.

[12] A. Buades, "Image and film denoising by non-local means," PhD thesis, Universitat de les Illes Balears, 2006.

[13] A. Buades, B. Coll, and J.-M. Morel, "Neighborhood filters and PDE's," *Numerische Mathematik*, vol. 105, no. 1, pp. 1–34, November 2006.

[14] A. Buades, B. Coll, and J.-M. Morel, "The staircasing effect in neighborhood filters and its solution," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1499–1505, 2006.

[15] F. Catté, P.-L. Lions, J.-M. Morel, and T. Coll, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM Journal of Numerical Analysis*, vol. 29, no. 1, pp. 182–193, February 1992.

[16] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 103, Proceedings of the ACM SIGGRAPH conference, 2007.

[17] K. Chiu, M. Herf, P. Shirley, S. Swamy, C. Wang, and K. Zimmerman, "Spatially nonuniform scaling functions for high contrast images," in *Proceedings of Graphics Interface '93*, pp. 245–254, May 1993.

[18] H. Chong, S. Gortler, and T. Zickler, "A perception-based color space for illumination-invariant image processing," *ACM Transactions on Graphics*, vol. 27, no. 3, pp. 1–7, Proceedings of the ACM SIGGRAPH conference, 2008.

[19] P. Choudhury and J. Tumblin, "The trilateral filter for high contrast images and meshes," in *Proceedings of the Eurographics Symposium on Rendering*, pp. 1–11, 2003.

[20] D. DeCarlo and A. Santella, "Stylization and abstraction of photographs," in *Proceedings of the ACM SIGGRAPH conference*, pp. 769–776, 2002.

[21] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Transactions on Graphics*, vol. 21, no. 3, pp. 257–266, Proceedings of the ACM SIGGRAPH conference, 2002.

[22] E. Eisemann and F. Durand, "Flash photography enhancement via intrinsic relighting," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 673–678, Proceedings of the ACM SIGGRAPH conference, July, 2004.

[23] M. Elad, "On the bilateral filter and ways to improve it," *IEEE Transactions on Image Processing*, vol. 11, no. 10, pp. 1141–1151, October 2002.

[24] M. Elad, "Retinex by two bilateral filters," in *Proceedings of the Scale-Space conference*, pp. 217–229, 2005.

[25] R. Fattal, M. Agrawala, and S. Rusinkiewicz, "Multiscale shape and detail enhancement from multi-light image collections," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 51, Proceedings of the ACM SIGGRAPH conference, 2007.

[26] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," *ACM Transactions on Graphics*, vol. 22, no. 3, pp. 950–953, Proceedings of the ACM SIGGRAPH conference, July, 2003.

[27] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, 1984.

[28] G. Gimel'farb, *Image Textures and Gibbs Random Fields*. Kluwer Academic Publishers, 1999. ISBN 0792359615.

[29] F. R. Hampel, E. M. Ronchetti, P. M. Rousseeuw, and W. A. Stahel, *Robust Statistics — The Approach Based on Influence Functions*. Wiley Interscience, 1986. ISBN 0-471-73577-9.

[30] P. J. Huber, *Robust Statistics. Probability and Statistics*. Wiley-Interscience, February 1981. ISBN: 9780471418054.

[31] L. Itti and C. Koch, "Computational modeling of visual attention," *Nature Reviews Neuroscience*, vol. 2, no. 3, pp. 194–203, 2001.

[32] D. J. Jobson, Z. Rahman, G. A. Woodell, N. Center, and V. A. Hampton, "A multiscale Retinex for bridging the gap between color images and the human observation of scenes," *IEEE Transactions on Image Processing*, vol. 6, no. 7, pp. 965–976, 1997.

[33] T. Jones, F. Durand, and M. Zwicker, "Normal improvement for point rendering," *IEEE Computer Graphics & Applications*, vol. 24, no. 4, pp. 53–56, 2004.

[34] T. R. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature-preserving mesh smoothing," *ACM Transactions on Graphics*, vol. 22, no. 3, Proceedings of the ACM SIGGRAPH conference, July, 2003.

[35] E. A. Khan, E. Reinhard, R. Fleming, and H. Buelthoff, "Image-based material editing," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 654–663, Proceedings of the ACM SIGGRAPH conference, 2006.

[36] J. J. Koenderink and A. J. Van Doorn, "The structure of locally orderless images," *International Journal of Computer Vision*, vol. 31, no. 2–3, pp. 159–168, 1999.

[37] J. Kopf, M. Uyttendaele, O. Deussen, and M. Cohen, "Capturing and viewing gigapixel images," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 93, Proceedings of the ACM SIGGRAPH conference, 2007.

[38] E. H. Land and J. J. McCann, "Lightness and Retinex theory," *Journal of the Optical Society of America*, vol. 61, no. 1, pp. 1–11, 1971.

[39] H. Land, Edwin, "The Retinex," *American Scientist*, vol. 52, pp. 247–264, 1964.

[40] S. Li, *Markov Random Field Modeling in Computer Vision*. Springer-Verlag, 1995. ISBN 4-431-70145-1.

[41] C. Liu, W. T. Freeman, R. Szeliski, and S. Kang, "Noise estimation from a single image," in *Proceedings of the Conference on IEEE Computer Vision and Pattern Recognition,* volume 1, pp. 901–908, 2006.

[42] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proceedings of the International Joint Conference on Artificial Intelligence,* volume 81, pp. 674–679, 1981.

[43] S. Mallat, *A Wavelet Tour of Signal Processing*. Academic Press, 1999. ISBN: 0-12-466606-X.

[44] Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations,* volume 22 of *University Lecture Series*. American Mathematical Society, 2001.

[45] A. Miropolsky and A. Fischer, "Reconstruction with 3D geometric bilateral filter," in *Proceedings of the ACM Symposium on Solid Modeling and Applications*, pp. 225–229, 2004.

[46] P. Mrázek, J. Weickert, and A. Bruhn, *Geometric Properties from Incomplete Data*. On Robust Estimation and Smoothing with Spatial and Tonal Kernels. Springer, 2006. ISBN: 978-1-4020-3857-0.

[47] D. A. Murio, *The Mollification Method and the Numerical Solution of Ill-Posed Problems*. Wiley-Interscience, 1993. ISBN: 0471594083.

[48] B. M. Oh, M. Chen, J. Dorsey, and F. Durand, "Image-based modeling and photo editing," in *Proceedings of the ACM SIGGRAPH Conference*, pp. 433–442, 2001.

[49] S. Paris, H. Briceño, and F. Sillion, "Capture of hair geometry from multiple images," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 712–719, Proceedings of the ACM SIGGRAPH conference, 2004.

[50] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 24–52, 2009.

[51] S. N. Pattanaik, J. A. Ferwerda, M. D. Fairchild, and D. P. Greenberg, "A multiscale model of adaptation and spatial vision for realistic image display," in *Proceedings of the ACM SIGGRAPH conference*, pp. 287–298, 1998.

[52] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Transactions Pattern Analysis Machine Intelligence*, vol. 12, no. 7, pp. 629–639, July 1990.

[53] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama, "Digital photography with flash and no-flash image pairs," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 664–672, Proceedings of the ACM SIGGRAPH Conference, 2004.

[54] T. Q. Pham, "Spatiotonal adaptivity in super-resolution of undersampled image sequences," PhD thesis, Delft University of Technology, 2006.

[55] T. Q. Pham and L. J. van Vliet, "Separable bilateral filtering for fast video preprocessing," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2005.

[56] R. Ramanath and W. E. Snyder, "Adaptive demosaicking," *Journal of Electronic Imaging*, vol. 12, no. 4, pp. 633–642, 2003.

[57] P. Sand and S. Teller, "Particle video: Long-range motion estimation using point trajectories," *International Journal of Computer Vision*, vol. 80, no. 1, pp. 72–91, 2008.

[58] C. Schlick, "Quantization techniques for visualization of high dynamic range pictures," in *Proceedings of the Eurographics Rendering Workshop*, pp. 7–20, 1994.

[59] S. M. Smith and J. M. Brady, "SUSAN — A new approach to low level image processing," *International Journal of Computer Vision*, vol. 23, no. 1, pp. 45–78, May 1997.

[60] N. Sochen, R. Kimmel, and A. M. Bruckstein, "Diffusions and confusions in signal and image processing," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 3, pp. 237–244, 2001.

[61] N. Sochen, R. Kimmel, and R. Malladi, "A general framework for low level vision," *IEEE Transactions in Image Processing*, vol. 7, pp. 310–318, 1998.

[62] T. G. Stockham, "Image processing in the context of a visual model," *Proceedings of the IEEE*, vol. 60, no. 7, pp. 828–842, 1972.

[63] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 839–846, 1998.

[64] J. Tumblin and G. Turk, "Low curvature image simplifiers (LCIS): A boundary hierarchy for detail-preserving contrast reduction," in *Proceedings of the ACM SIGGRAPH Conference*, pp. 83–90, 1999.

[65] J. van de Weijer and R. van den Boomgaard, "Local mode filtering," in *Proceedings of the conference on IEEE Computer Vision and Pattern Recognition*, pp. 428–433, 2001.

[66] J. van de Weijer and R. van den Boomgaard, "On the equivalence of local-mode finding, robust estimation and mean-shift analysis as used in early vision tasks," in *Proceedings of the International Conference on Pattern Recognition*, pp. 927–930, 2002.

[67] L. Vese and S. Osher, "Modeling textures with total variation minimization and oscillating patterns in image processing, Journal of Scientific Computing," *Journal of Scientific Computing*, vol. 19, pp. 553–572, 2003.

[68] C. C. Wang, "Bilateral recovering of sharp edges on feature-insensitive sampled meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 4, pp. 629–639, 2006.

[69] L. Wang, L.-Y. Wei, K. Zhou, B. Guo, and H.-Y. Shum, "High dynamic range image hallucination," in *Proceedings of the Eurographics Symposium on Rendering*, pp. 321–326, 2007.

[70] G. S. Watson, *Statistics on Spheres*. John Wiley and Sons, 1983.

[71] B. Weiss, "Fast median and bilateral filtering," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 519–526, Proceedings of the ACM SIGGRAPH conference, 2006.

[72] H. Winnemöller, S. C. Olsen, and B. Gooch, "Real-time video abstraction," *ACM Transactions on Graphics*, vol. 25, no. 3, pp. 1221–1226, Proceedings of the ACM SIGGRAPH conference, 2006.

[73] W. C. K. Wong, A. C. S. Chung, and S. C. H. Yu, "Trilateral filtering for biomedical images," in *Proceedings of the IEEE International Symposium on Biomedical Imaging*, pp. 820–823, 2004.

[74] J. Xiao, H. Cheng, H. Sawhney, C. Rao, and M. Isnardi, "Bilateral filtering-based optical flow estimation with occlusion detection," in *Proceedings of the European Conference on Computer Vision*, pp. 211–224, 2006.

[75] Q. Yáng, R. Yang, J. Davis, and D. Nistér, "Spatial-depth super resolution for range images," in *Proceedings of the conference on IEEE Computer Vision and Pattern Recognition*, pp. 1–8, 2007.

[76] Q. Yáng, R. Yang, H. Stewénius, and D. Nistér, "Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling," in *Proceedings of the Conference on IEEE Computer Vision and Pattern Recognition*, pp. 2347–2354, 2006.

[77] L. P. Yaroslavsky, *Digital Picture Processing. An Introduction*. Springer Verlag, 1985.

[78] K. Yoon and I. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 4, pp. 650–656, 2006.