Int J Comput Vis (2015) 112:307–318
DOI 10.1007/s11263-014-0764-y

Constant Time Median and Bilateral Filtering

Qingxiong Yang · Narendra Ahuja · Kar-Han Tan

Received: 12 October 2013 / Accepted: 6 September 2014 / Published online: 30 September 2014

Abstract This paper formulates both the median filter and bilateral filter as a cost volume aggregation problem whose computational complexity is independent of the filter kernel size. Unlike most of the previous works, the proposed framework results in a general bilateral filter that can have arbitrary spatial[1] and arbitrary range filter kernels. This bilateral filter takes about 3.5 s to exactly filter a one megapixel 8-bit grayscale image on a 3.2 GHz Intel Core i7 CPU. In practice, the intensity/range and spatial domain can be downsampled to improve the efficiency. This compression can maintain very high accuracy (e.g., 40 dB) but over 100× faster.

Keywords Bilateral filtering · Edge-preserving smoothing · Recursive filtering

1 Introduction

The objective of bilateral filtering is to smooth images. It is done by replacing the intensity (color) value of a pixel by the average of the values of other pixels weighted by their spatial distance and intensity similarity to the original pixel. Zucker and Rosenfeld (1977) used this idea by identifying similar pixels by first detecting edges. They iteratively replaced the intensity of a pixel by the average of all the pixels in a small (3×3) neighborhood, and on the same side of the edge as the pixel itself. Davis and Rosenfeld (1978) identified similar pixels differently, by choosing those six of the nine pixels in the neighborhood that are closest in intensity to the original pixel, and used their median to obtain the smoothed value. Ahuja (1996) proposed a transform to compute the net similarity between a pixel and all other pixels in the image, as well as the direction in which the largest number of a pixel's most similar pixels are located. The latter was captured by computing a force vector at the pixel. Tabb and Ahuja (1997) presented a detailed algorithm for multi-scale image segmentation using the force transform and demonstrated the performance advantages of the similarity measure incorporated in the force transform. Tomasi and Manduchi (1998) used the same definition of similarity as proposed by Ahuja (1996) and used it for image smoothing. They replaced pixel values with similarity-weighted averages and called it bilateral filtering.

Bilateral filter has been demonstrated to be very effective for many computer vision and computer graphics tasks but its brute-force implementation is known to be computationally intensive when the filter kernel is large (Paris et al. 2009). Several methods Durand and Dorsey (2002), Elad (2002), Pham and van Vliet (2005), Weiss (2006), Paris and Durand (2009) enable it to be computed at either $O(r)$ or $O(log(r))$ runtime in the radius of the filter r. By filtering on the subsampled image, Paris and Durand (2009) prove that the runtime of Durand and Dorsey's method Durand and Dorsey (2002) decreases as the filter size increases because the subsampling factor can be increased without significantly impacting the accuracy of the result. This method is relatively slow when the filter size is small. Chen et al. (2007) later show that the GPU implementation of Durand and Dorsey (2002) can achieve video rate.

Communicated by C. Schnörr.

Q. Yang (✉)
City University of Hong Kong, Kowloon Tong, Hong Kong, China
e-mail: liiton.research@gmail.com
URL: http://www.cs.cityu.edu.hk/~qiyang/

N. Ahuja
University of Illinois at Urbana-Champaign, Champaign, IL, USA

K.-H. Tan
Hewlett-Packard Laboratories, Palo Alto, CA, USA

Porikli (2008) proposes three types of $O(1)$ bilateral filters for processing grayscale images: (1) box spatial and arbitrary range kernels; (2) arbitrary spatial and polynomial range kernels; and (3) arbitrary spatial and Gaussian range kernels. Kass and Solomon (2010) use a smoothed histogram to accelerate median filtering and propose mode-based filters. Adams et al. (2009) then propose to use Gaussian KD-trees for efficient high-dimensional Gaussian filtering. This method can be directly integrated with Paris and Durand's method Paris and Durand (2009) for fast bilateral filtering. They later propose to use permutohedral lattice (Adams et al. 2010) for bilateral filtering, which is faster than Gaussian KD-trees (Adams et al. 2009) for relatively lower dimensionality but has a higher memory cost. Both Gaussian KD-trees and permutohedral lattice are relatively less efficient when used to process grayscale images and are not suitable for parallel implementation. Yang (2012b) proves that the bilateral filter can be recursively implemented so that the complexity can be linear in both input size and dimensionality. However, it is limited to only a special-range filter kernel. A number of edge-preserving filtering methods that have similar applications but lower computational complexity emerged recently. For instance, Fattal's EAW method Fattal (2009), He et al.'s guided image filtering method He et al. (2010), and Gastal and Oliveira's domain transform filtering method Gastal and Oliveira (2011). However, their behaviors are not exactly the same as the bilateral filter and thus they are not suitable for some bilateral filtering applications like highlight removal (Yang et al. 2010).

This paper extends Paris and Durand's bilateral filtering method Paris and Durand (2009) for median filtering by formulating it as a cost volume aggregation problem whose computational and memory complexity is independent of the filter kernel size. This formulation is naturally extended to a general bilateral filter that can have arbitrary spatial[1] and arbitrary range filter kernels. The resulting bilateral filter takes about 3.5 s to exactly filter a one megapixel 8-bit grayscale image on a modern CPU. Similar to Paris and Durand (2009), the intensity/range and spatial domain can be downsampled to improve the efficiency in practice. A novel spatial upsampling approach is proposed in this paper which enables this compression to maintain very high accuracy (*e.g.*, 40 dB) but over 100× faster: about 33 ms to filter a one megapixel 8-bit grayscale image. The framework can be further extended for constant time weighted median filtering (Yin et al. 1996) which is very effective for noise attenuation.

2 Constant Time Median Filtering

The median filter replaces each pixel p with the median of neighboring pixel values within a box region Ω_p. Let I_p denote the pixel value at p and I_p^M denote the corresponding median filtered value,

$$I_p^M = \arg\min_x \left(\sum_{q \in \Omega_p} |I_q - x| \right), \tag{1}$$

given the fact that median actually minimizes the sum of absolute error of the given data (Huber and Wiley 1981).

When the input signal is an 8-bit digital image, the possible median value only resides in a total of 256 digital numbers ranging from 0 to 255, that is $x \in [0, 255]$ in Eq. (1). Letting the absolute error in Eq. (1) be the cost, a cost volume $\mathcal{C}$ can then be computed given an 8-bit image:

$$\mathcal{C}_{q,x} = |I_q - x|,\ x \in [0, 255] \tag{2}$$

and the median filter is formulated as a standard cost aggregation problem followed by a winner-take-all selection. As a result, the main computational cost of the median filter resides in the cost aggregation step. The integral image[2] Viola and Jones (2001) enables the cost aggregation to be independent of the size of the filter; and thus the computational complexity of the median filter is linear regarding the number of image pixels and the intensity level.

In practice, the cost volume can be downsampled in the intensity domain before aggregation to improve the efficiency. That is, 256 intensity values are quantized into only a total of $N < 256$ bins, and the cost is computed at each bin and each pixel location. Instead of using a winner-take-all selection, a quadratic interpolation step is used to upsample the intensity domain after cost aggregation. Specifically, let C^A denote the aggregated cost volume. A quadratic function

$$f(x) = ax^2 + bx + c, \tag{3}$$

is fitted at each pixel location using the lowest cost (let it be $f(x_0) = \mathcal{C}^A_{p,x_0}$) and two of its closest neighbors (let them be $f(x_1) = \mathcal{C}^A_{p,x_1}$ and $f(x_2) = \mathcal{C}^A_{p,x_2}$), and the upsampled intensity value is then computed as the x value where the minimum $f(x)$ occurs. According to calculus, this x value is $x_{\min} = -\frac{b}{2a}$, where

$$\begin{aligned}\begin{bmatrix} a \\ b \end{bmatrix} &= \begin{bmatrix} x_1^2 - x_0^2 & x_1 - x_0 \\ x_2^2 - x_0^2 & x_2 - x_0 \end{bmatrix}^{-1} \begin{bmatrix} f(x_1) - f(x_0) \\ f(x_2) - f(x_0) \end{bmatrix} \\ &= \begin{bmatrix} x_1^2 - x_0^2 & x_1 - x_0 \\ x_2^2 - x_0^2 & x_2 - x_0 \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{C}^A_{p,x_1} - \mathcal{C}^A_{p,x_0} \\ \mathcal{C}^A_{p,x_2} - \mathcal{C}^A_{p,x_0} \end{bmatrix}.\end{aligned}$$

In the extreme case, the input image consists of a single intensity value; thus the median filtered image is exactly the

[1] an IIR O (1) solution needs to be available for the kernel.

[2] Integral image is an image representation. The integral image at a pixel location contains the sum of the pixels above and to the left of the pixel.

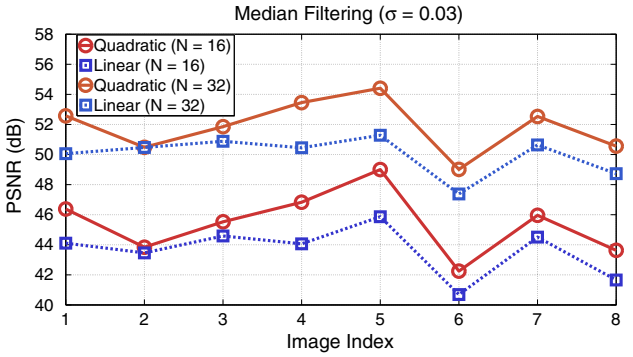


Fig. 1 Numerical comparison of the quadratic and linear interpolation for the proposed median filtering method using the eight images presented in Fig. 2. The *red solid line* corresponds to the PSNR values obtained from quadratic interpolation and the *blue dashed line* from linear interpolation. As can be seen, quadratic interpolation is more accurate over all (Color figure online)

same as the input image and the shape of the aggregated cost is exactly the same as the absolute error function presented in Eq. (2). In this case, the ground truth can be obtained by either interpolating the aggregated cost using this absolute error function or the quadratic function presented in Eq. (3). However, in real images, the intensity values within a local region are likely to be similar but normally different; thus interpolation with a quadratic function is proved to be more accurate as shown in Fig. 1. The eight images used in this experiment were selected from different categories Paris and Durand (2009) as shown in Fig. 2. These images were chosen to be as different as possible to cover a broad spectrum of content including the following:

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

An ambiguity to be clarified is how to set the number of quantization levels N to avoid visible artifacts. Apparently, when $N = 256$, the obtained filtered image is exactly the same as the ground truth. PSNR values are then computed from the median filtered image obtained from different N

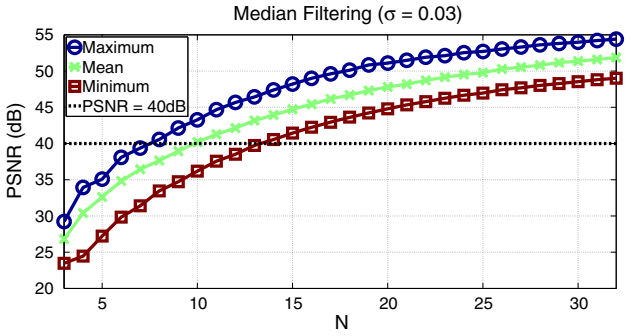


Fig. 3 Performance of the proposed median filter *w.r.t.* N when $\sigma = 0.03$. As can be seen, the performance increases *w.r.t.* N and the minimum PSNR value (shown as a *red curve*) is always over 40 dB when $N \geq 14$ (Color figure online)

values and the ground truth as shown in Fig. 3. The images presented in Fig. 2 are used in this experiment, and the minimum, average, and maximum PSNR values obtained from these images are presented as red, green, and blue curves in Fig. 3. Note that the minimum PSNR value is always higher than 40 dB when $N \geq 14$. It is concluded in Paris and Durand (2009) that a PSNR value above 40 dB often corresponds to an almost invisible difference. Thus when $N = 14$, it is guaranteed that there will be very limited visible artifacts in any of the eight median filtered images.

The performance with respect to the filter size is evaluated in Fig. 4. The filter size is represented by $\mathbf{round}(2\sigma H + 1) \times \mathbf{round}(2\sigma W + 1)$, where H and W are the height and width of the input image, and $\sigma \in [0, 1]$ controls the filter size. The number of quantization level N is set to 16 in this experiment and the eight images in Fig. 2 are used. Note that the performance increases with respect to σ over all, and the minimum PSNR value (shown as a red curve) is always over 40 dB for all $\sigma \geq 0.005$ when $N = 16$. The median filtering results of the *Tulip* test image in Fig. 2 are presented in Fig. 5a–c. Note that there are very few noticeable errors.

3 Constant Time Bilateral Filtering

A bilateral filter has two filter kernels: a spatial filter kernel $\mathcal{F}(p, q)$ that measures spatial distance between two pixel locations p and q and a range filter kernel $\mathcal{G}(I_p, I_q)$ that measures the intensity/range distance between two pixel

Fig. 2 Eight images used in evaluating the performance of the proposed filtering methods. These images are selected from different categories and used by Paris and Durand (2009) to demonstrate their fast bilateral filtering method

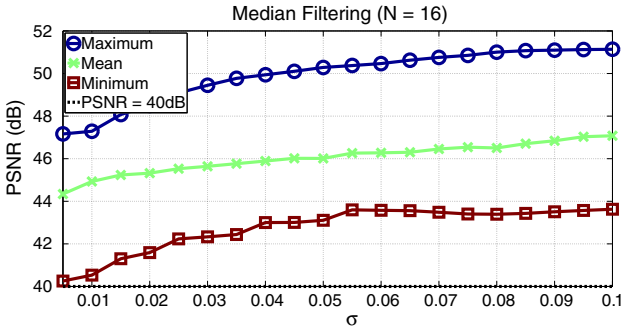


Fig. 4 Performance of the proposed median filter *w.r.t.* σ when $N = 16$. As can be seen, the performance increases *w.r.t.* σ over all, and the minimum PSNR value (shown as a *red curve*) is always over 40 dB for all $\sigma \geq 0.005$ (Color figure online)

(a) Input **(b)** Our MF(46.0 dB) **(c)** G.T. MF

(d) Our Box BF(56.7 dB) **(e)** G.T. Box BF **(f)** Our Gaussian BF(56.6 dB) **(g)** G.T. Gaussian BF

Fig. 5 Visual verification. **a** is one of the test images in Fig. 2, **b** is obtained from our median filtering method ($\sigma = 0.03$ and $N = 16$) and **c** is the ground truth; **d** and (f) are obtained from our bilateral filtering method ($\sigma_S = 0.03, \sigma_R = 0.1$ and $N = 8$) with box and Gaussian spatial filter kernels, respectively. The results obtained from the ground-truth bilateral filters are presented in (**e**) and (**g**), respectively. Experimental results for all the test images in Fig. 2 are presented at http://www.cs.cityu.edu.hk/~qiyang/ijcv-13/

values I_p and I_q. If the range filter kernel $\mathcal{G}$ is computed using an additional image T, the resulting filter becomes a joint/cross bilateral filter (Eisemann and Durand 2004; Petschnigg et al. 2004). Let I_p^T denote the joint bilateral filtered value at pixel location p,

$$I_p^T = \omega_p^{-1} \sum_{q \in \Omega_p} \mathcal{F}(p, q) \cdot \mathcal{G}(T_p, T_q) I_q, \tag{4}$$

where

$$\omega_p = \sum_{q \in \Omega_p} \mathcal{F}(p, q) \mathcal{G}(T_p, T_q) \tag{5}$$

is a normalizing parameter at the corresponding pixel location, and the Gaussian kernel is usually selected as the range filter kernel in practice.

$$\mathcal{G}(T_p, T_q) = exp\left(-\frac{|T_p - T_q|^2}{2\sigma_R^2}\right). \tag{6}$$

3.1 Bilateral Filtering Using Integral Image and Recursive Filters

In this section, we show that the computational complexity of the bilateral filter is independent of the filter kernel size, as long as its spatial filter kernel has a recursive implementation.

For simplicity, let the spatial filter kernel be a box filter so that:

$$I_p^T = \frac{\sum_{q \in \Omega_p} 1 \cdot \mathcal{G}(T_p, T_q) I_q}{\omega_p}, \tag{7}$$

where

$$\omega_p = \sum_{q \in \Omega_p} 1 \cdot \mathcal{G}(T_p, T_q). \tag{8}$$

Let $K_p = I_p^T \cdot \omega_p$ so that the normalizing parameter ω_p is ignored,

$$K_p = \sum_{q \in \Omega_p} 1 \cdot \mathcal{G}(T_p, T_q) I_q. \tag{9}$$

Let T be an 8-bit image; potential values of T_p can then be up to 256 numbers: $T_p \in [0, 255]$. Replacing T_p with a constant value $x \in [0, 255]$, Eq. (9) indeed represents a local aggregation of an image whose pixel value at pixel q is $\mathcal{G}(x, T_q) I_q$. As a result, the cost aggregation framework presented in Sect. 2 can be directly used to compute K_p by 1) changing the cost function in Eq. (2) to:

$$\mathcal{C}_{q,x} = \mathcal{G}(x, T_q) I_q, \; x \in [0, 255] \tag{10}$$

and 2) replacing the winner-take-all selection to simply set $x = T_p$ at each pixel location p to access the aggregated cost:

$$K_p = \mathcal{C}_{p,x}^A = \mathcal{C}_{p,T_p}^A. \tag{11}$$

According to Eq. (8) and (9), ω_p can be obtained in the same manner by setting the cost function to:

$$\mathcal{C}_{q,x} = \mathcal{G}(x, T_q), \; x \in [0, 255]. \tag{12}$$

Finally, K_p is normalized by ω_p to obtain the bilateral filtered value I_p^T according to Eq. (7).

As discussed above, the bilateral filter is again formulated as a cost volume aggregation problem when the spatial filter

is a box filter. As shown in Sect. 2, the cost volume can be aggregated using an integral image; the computational complexity of the bilateral filter is thus independent of the filter size. Bilateral filtering using other spatial filter kernels can actually be computed in exactly the same manner by filtering the cost volume using the corresponding spatial filter kernels. If the spatial filter kernel has a recursive implementation, the computational complexity of the spatial filter will be independent of the filter kernel size by adjusting the coefficients of the recursive system (Smith 2007). Three spatial filters are tested in this paper including the box filter, exponential filter (Smith 2007), and recursive Gaussian filter (Deriche 1992). The box filter kernel size is again defined using a single parameter $\sigma_S \in [0, 1]$: $\mathbf{round}(2\sigma_S H + 1) \times \mathbf{round}(2\sigma_S W + 1)$, where H and W are the height and width of the input image. The exponential filter and recursive Gaussian filter can be implemented recursively, and the kernel size is controlled by the coefficients of the recursive system determined also by σ_S. In our experiments, the exponential filter uses $\exp(-\sqrt{2}/\sigma_S)$ as the feedback coefficient (Smith 2007), and Gaussian filter uses σ_S as the standard deviation.

3.2 Quantization on Range Domain

Similar to the proposed median filtering method, the aggregated cost volume is downsampled in the range/intensity domain before aggregation/filtering to improve the efficiency in practice. A linear interpolation guided by the original pixel value I_p at each pixel location is used to upsample the intensity domain after aggregation/filtering. Specifically, let L^- and L^+ denote two of the successive selected intensity levels where $L^- \leq T_p \leq L^+$ and:

$$\alpha_p = \frac{T_p - L^-}{L^+ - L^-}, \tag{13}$$

the joint bilateral filtered value at pixel p is:

$$I_p^T = (1 - \alpha_p)\frac{K_p^-}{\omega_p^-} + \alpha_p \frac{K_p^+}{\omega_p^+}, \tag{14}$$

where K_p^- (and ω_p^-) and K_p^+ (and ω_p^+) are retrieved from the aggregated/filtered cost volume(s) when the intensity level is L^- and L^+, respectively.

Unlike median filtering that requires a single cost volume, two cost volumes (one for computing K_p and the other for ω_p) is used in the proposed bilateral filtering method; thus the computation cost will be theoretically twice higher. However, the numerical comparison presented in Fig. 1 and Fig. 6 shows that the performance of the proposed bilateral filter will be higher than the proposed median filter when the same amount of quantization levels N is used. The red curve in Fig. 1 shows that when $N \geq 14$, the minimum PSNR value will be greater than 40 dB which corresponds to almost invisible error, while the red curves in Fig. 6 reveal that only $N \geq 6$ is required. Only the box spatial filter is analyzed in Fig. 6 because the performance *w.r.t.* N is extremely close between different spatial filter kernels.

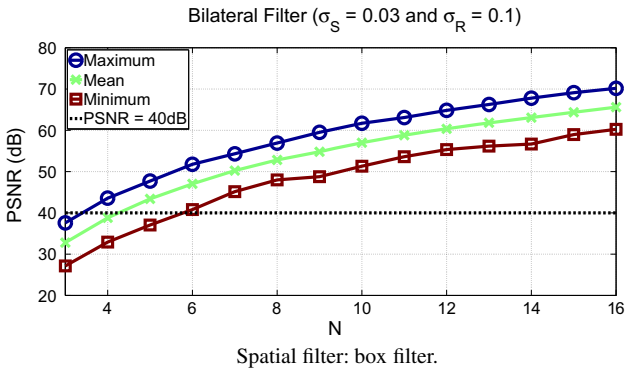


Fig. 6 Performance of the proposed bilateral filter *w.r.t.* N when $\sigma_S = 0.03$ and $\sigma_R = 0.1$. As can be seen, the performance increases *w.r.t.* N and the minimum PSNR value (shown as *red curves*) is always over 40 dB when $N \geq 6$ (Color figure online)

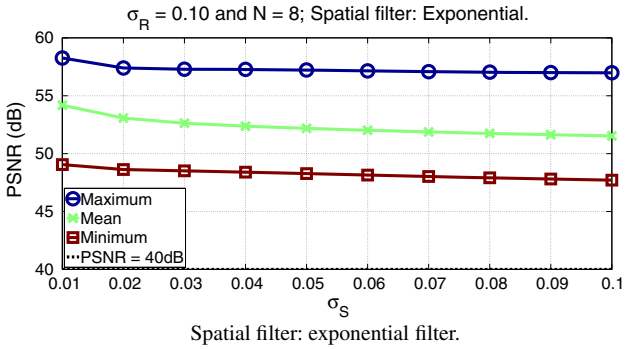


Fig. 7 Performance of the proposed bilateral filter *w.r.t.* σ_S when $N = 8$ and $\sigma_R = 0.1$. Its performance is higher when smaller filter kernels (small σ_S values) are used but is very robust when $\sigma_S \geq 0.05$. Note that the minimum PSNR value (shown as *red curves*) is always over 40 dB when $\sigma_S \leq 0.1$ (Color figure online)

Figures 7 and 8 evaluate the performance of the proposed bilateral filter with respect to the filter kernel size defined by $\sigma_S \in [0, 1]$ and $\sigma_R \in [0, 1]$[3] when the quantization level $N = 8$. Two other spatial filters—exponential filter and recursive Gaussian filter - are used. As can be seen, the performance is robust to σ_S but increases *w.r.t.* σ_R. Nevertheless, the minimum PSNR value (shown as red curves) is always over 40 dB when $\sigma_R \geq 0.05$. The bilateral filtering results of the *Tulip* test image in Fig. 2 are presented in Fig. 5d–g. In fact, there is no noticeable error.

[3] σ_R is the standard deviation of the Gaussian range filter kernel defined in Eq. (6). In this paper, the image intensity is normalized so that it ranges from 0 to 1.

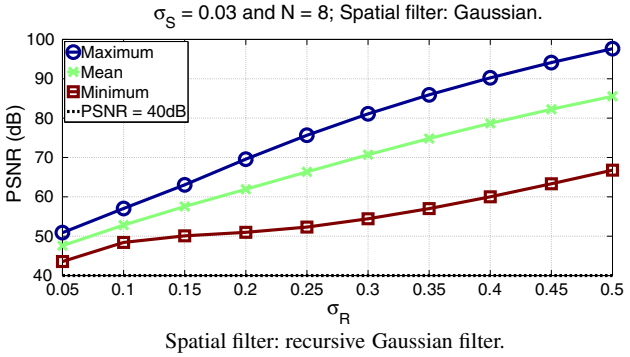


Fig. 8 Performance of the proposed bilateral filter *w.r.t.* σ_R when $N = 8$ and $\sigma_S = 0.03$. Apparently, the performance increases *w.r.t.* σ_R and the minimum PSNR value (shown as *red curves*) is always over 40 dB when $\sigma_R \geq 0.05$ (Color figure online)

3.3 Quantization on Spatial Domain

Besides quantization on the intensity domain, spatial quantization can also be used to achieve further speedup. The main differences are that (1) the cost volumes will be built using a downsampled version of the input image and (2) the aggregated low-resolution cost volume needs to be upsampled to the original resolution for computing K_p^- (and ω_p^-) K_p^+ (and ω_p^+) in Eq. (14). A simple solution is using linear upsampling. Let $K^{\downarrow}$ and $\omega^{\downarrow}$ denote the low-resolution aggregated cost volumes, $p_{\downarrow}$ denote the pixel coordinate of pixel p in the original resolution, and $p_{\downarrow 0}$, $p_{\downarrow 1}$, $p_{\downarrow 2}$, $p_{\downarrow 3}$ denote the pixel coordinates of the four nearest neighbors of pixel $p_{\downarrow}$; then:

$$\frac{K_p^-}{\omega_p^-} = \mathcal{L}\left(\frac{K_{p_{\downarrow 0}}^{\downarrow -}}{\omega_{p_{\downarrow 0}}^{\downarrow -}}, \frac{K_{p_{\downarrow 1}}^{\downarrow -}}{\omega_{p_{\downarrow 1}}^{\downarrow -}}, \frac{K_{p_{\downarrow 2}}^{\downarrow -}}{\omega_{p_{\downarrow 2}}^{\downarrow -}}, \frac{K_{p_{\downarrow 3}}^{\downarrow -}}{\omega_{p_{\downarrow 3}}^{\downarrow -}}\right), \tag{15}$$

where $\mathcal{L}$ corresponds to the linear interpolation operation. $\frac{K_p^+}{\omega_p^+}$ is computed in the same manner.

Linear upsampling works well for images full of low-texture regions even when a large downsampling factor is used. A low-texture example is presented in the first row of Fig. 9. The resolution of the input image in Fig. 9a is 1180×880. With a downsampling factor of 16, the cost volume is built from a 73×55 downsampled version of the input image in Fig. 9a. The PSNR value computed from the ground truth in Fig. 9b and the filtered image in Fig. 9d is 39 dB, which is relatively high. However, there are visible errors around tiny/thin-structured objects. For instance, the specular highlights in both eyes and the mouse are over-smoothed as shown in the close-ups in Fig. 9d. The error increases when the image mainly consists of thin-structured objects. For instance, Fig. 9f is a partial screenshot of a webpage.

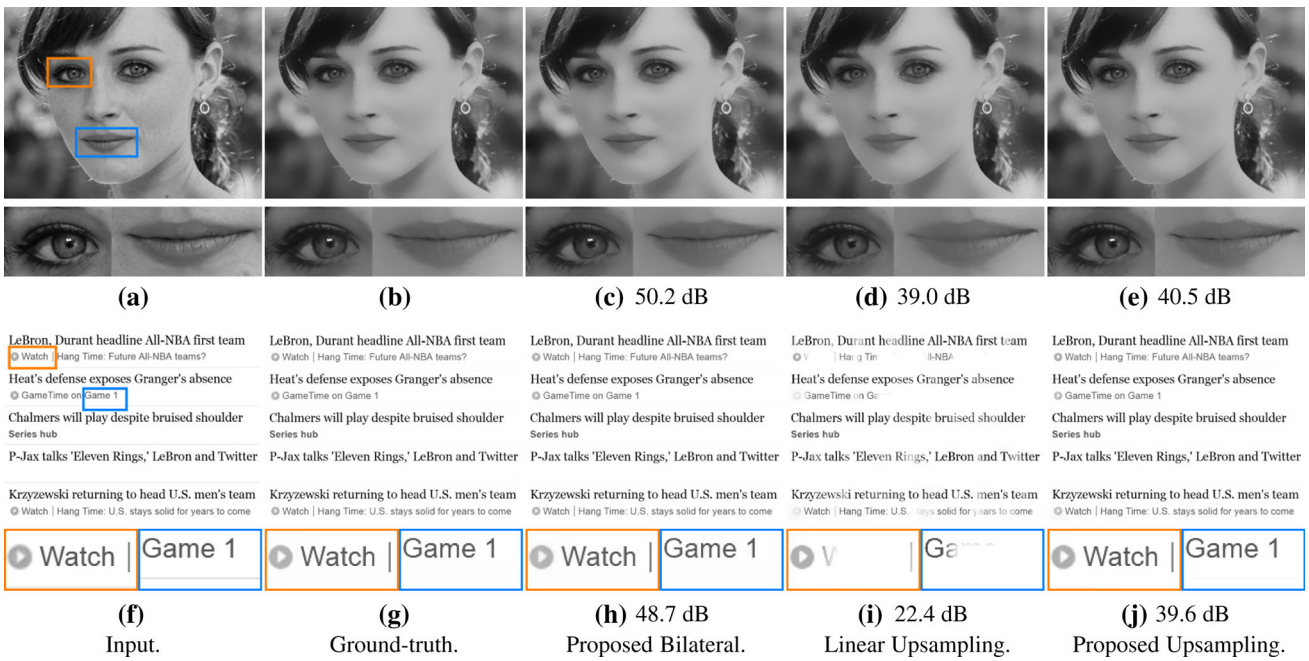
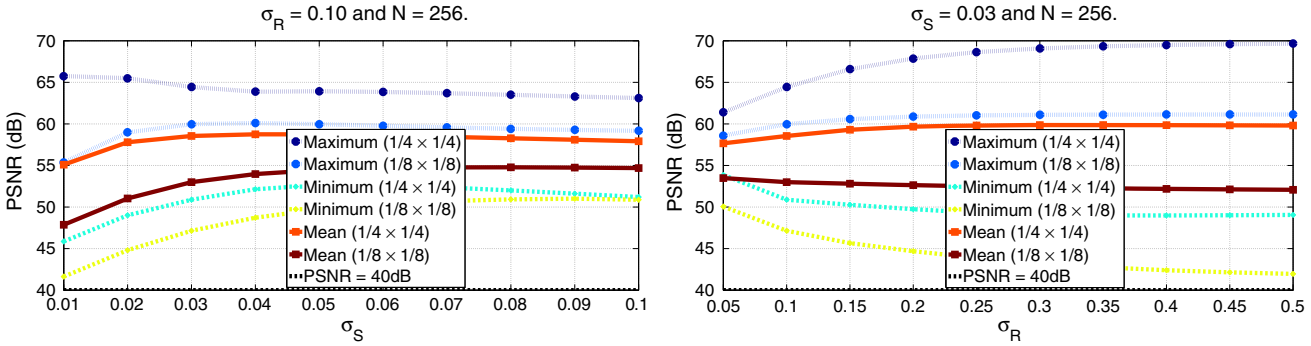


Fig. 9 Spatial quantization. The first three columns contain the input images, the ground-truth bilateral filtered images, and the filtered images obtained from the proposed bilateral filter with only intensity quantization. The last two columns are the filtered images obtained from the proposed bilateral filter with both intensity and spatial quantization: the left column uses the linear upsampling method and the right column uses the proposed upsampling method. When spatial quantization is applied with the linear upsampling method, the filtering quality around thin-structured objects is low as shown in (**d**) and (**i**). There is clear visual improvement in (**e**) and (**j**) when the proposed upsampling method is used. Parameter setting: $\sigma_S = 0.03$, $\sigma_R = 0.01$, and a $\frac{1}{16} \times \frac{1}{16}$ downsampled version of the input images in (**a**) and (**f**) are used to compute (**d**, **e**) and (**i**, **j**)



(a) PSNR *w.r.t.* σ_S when $N = 256$ and σ_R =0.1.

(b) PSNR *w.r.t.* σ_R when $N = 256$ and σ_S =0.03.

Fig. 10 Performance of the proposed bilateral filter when only spatial quantization is employed ($N = 256$). As can be seen from (**a**), the performance normally increases *w.r.t.* spatial filter kernel size (σ_S) when σ_R is fixed (to 0.1), especially when the downsampling factor is high. This is consistent with Fig. 8, which shows that the performance increases with σ_R when only range quantization is employed. However, the behavior is different when σ_S is fixed (to 0.03) as can be seen from (**b**). The *blue curves* in (b) show that the performance increases *w.r.t.* σ_R on a smooth natural photograph (e.g., Fig. 2h) while the cyan and yellow curves show that the performance decreases *w.r.t.* σ_R on a noisy artificial image (e.g., Fig. 2a). See Sect. 3.5 for details (Color figure online)

It contains mainly text. After bilateral filtering, the PSNR value computed from the ground truth in Fig. 9g and the filtered image in Fig. 9i is only 22.4 dB, which is very low. Obviously, a simple linear upsampling is not enough for this image.

Errors around thin-structured objects reside in the removal of these objects in the low-resolution version after spatial quantization and thus the contribution of these objects will not be taken into account during cost volume aggregation. To reduce the error, we simply need to restore at least part of these contributions. Taking into account both the accuracy and efficiency, only the contribution of the center pixel is restored during linear upsampling and Eq. (15) is updated as follows:

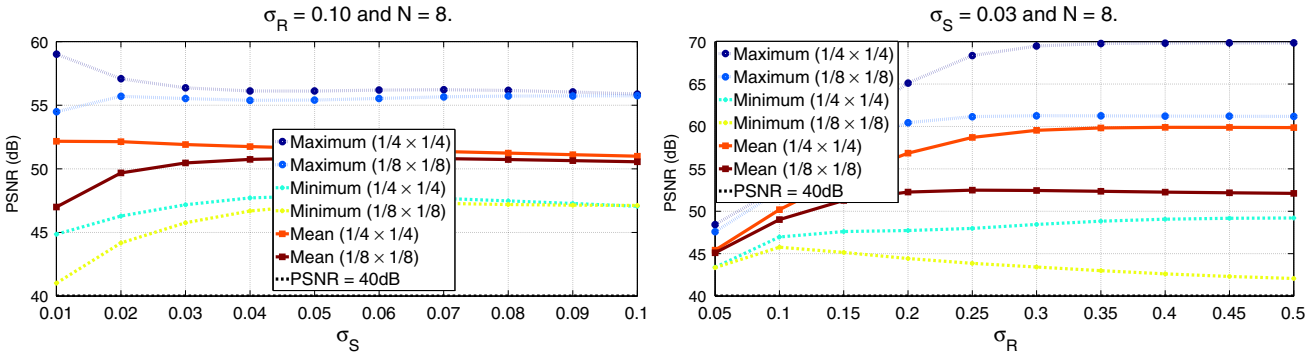
$$\frac{K_p^-}{\omega_p^-} = \mathcal{L}\left(\frac{K_{p\downarrow 0}^{\downarrow -} + \kappa I_p}{\omega_{p\downarrow 0}^{\downarrow -} + \kappa}, \frac{K_{p\downarrow 1}^{\downarrow -} + \kappa I_p}{\omega_{p\downarrow 1}^{\downarrow -} + \kappa}, \frac{K_{p\downarrow 2}^{\downarrow -} + \kappa I_p}{\omega_{p\downarrow 2}^{\downarrow -} + \kappa}, \frac{K_{p\downarrow 3}^{\downarrow -} + \kappa I_p}{\omega_{p\downarrow 3}^{\downarrow -} + \kappa}\right), \tag{16}$$

where $\kappa = \mathcal{G}(L^-, T_p)$ is computed based on the range filter kernel. This simple adjustment in the linear upsampling step restores the contribution of the center pixel. It will not increase the computational complexity of the proposed bilateral filtering method. However, it may significantly improve the filtering quality around thin-structured objects as demonstrated in Fig. 9i, j. The PSNR value is increased from 22.4 to 39.6 dB, which means that there is almost no visible difference between the ground truth in Fig. 9g and our filtered image in Fig. 9j. The specular highlights in the eyes in Fig. 9b is also maintained using the proposed upsampling method as shown in Fig. 9e.

Detailed analysis of spatial quantization using the eight images in Fig. 2 is presented in Fig. 10. The number of intensity quantization levels N is set to 256 to avoid range quantization. As can be seen, the PSNR values are always over 40 dB even when a small spatial filter kernel is employed. The images were resized to a 1600×1600 resolution so that the minimum spatial filter kernel is 5×5 (when $\sigma_S = 0.01$) with a downsampling factor of 8 ($\frac{1}{8} \times \frac{1}{8}$). As expected, the performance decreases *w.r.t.* to the spatial quantization level and increases *w.r.t.* σ_S especially when the downsampling factor is high as can be seen from Fig. 10a. This is consistent with Fig. 8, which shows that the performance increases *w.r.t.* σ_R when range quantization is employed. However, the behavior is different when σ_S is fixed. According to Fig. 10b, the performance increases *w.r.t.* σ_R on a natural photograph (e.g., Fig. 2h) and decreases on a noisy artificial image (e.g., Fig. 2a). The images in Fig. 2 are listed *w.r.t.* the PSNR values obtained with spatial quantization when $\sigma_S = 0.01$ and $\sigma_R = 0.1$. The noisy artificial image in Fig. 2a consistently obtains the lowest PNSR values as a quarter of the whole image (in the bottom left corner) contains only white noise. Sect. 3.5 presents detailed analysis *w.r.t.* different levels of noise.

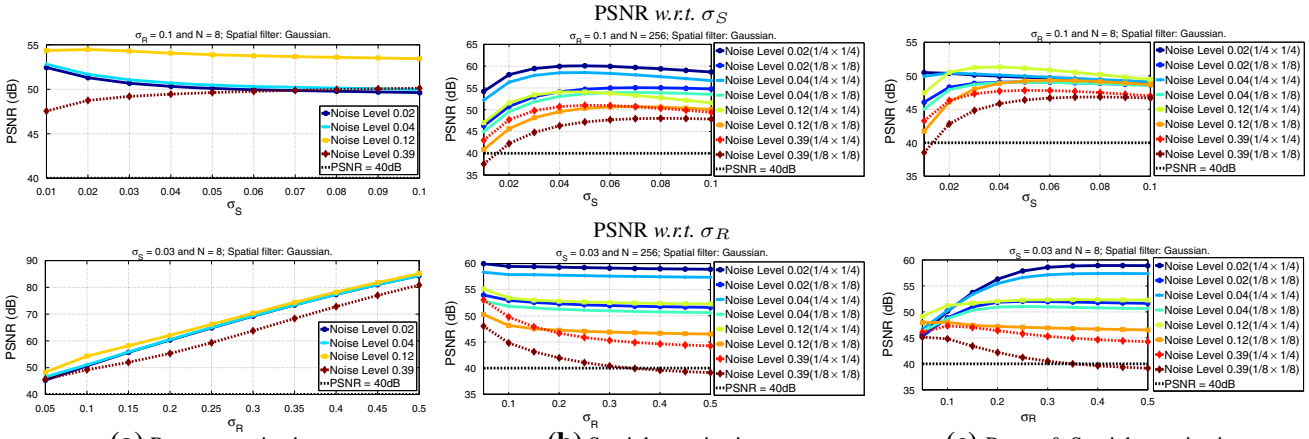
3.4 Quantization on Both Range and Spatial Domain

To perform quantization on both range and spatial domains, we can simply adopt the algorithm in Sect. 3.3 but reduce the number of intensity quantization levels N to a value smaller than 256. Similar to the analysis in Sect. 3.2, we notice that $N = 8$ is good enough in practice. The PSNR performance *w.r.t.* to σ_S and σ_R is presented in Fig. 11. As can be seen, the performance is mainly a combination of Figs. 7 and 8 (when only range quantization is employed) and Fig. 10 (when only spatial quantization is employed). Figure 7 shows that the performance is stable *w.r.t.* to σ_S when only range quantization is employed, and thus Fig. 11a mainly follows Fig. 10a



(a) PSNR *w.r.t.* σ_S when $N = 8$ and σ_R =0.1. **(b)** PSNR *w.r.t.* σ_R when $N = 8$ and σ_S =0.03.

Fig. 11 Performance of the proposed bilateral filter when both range and spatial quantization is employed. The performance is mainly a combination of Fig. 7, 8 and 10 and normally increases *w.r.t.* σ_S and σ_R except for the noisy artificial image in Fig. 2a (as indicated by the *yellow curve* in b) (Color figure online)



(a) Range quantization **(b)** Spatial quantization **(c)** Range & Spatial quantization

Fig. 12 Mean PSNR values obtained with four different levels of synthetic white Gaussian noise. **a** shows that the noise does affect the proposed filter much when only range quantization is employed. However, **b** and **c** show that the performance decreases *w.r.t.* the noise level if spatial quantization is employed

when spatial quantization is employed as well. Figure 8 confirms that the performance increases *w.r.t.* to σ_R when only range quantization is employed. However, Fig. 11b indicates that this is not always true when spatial quantization is also employed. Specifically, when both range and spatial quantization is employed, the performance generally increases *w.r.t.* to σ_R like Fig. 8. But when the spatial downsampling factor is high and the test image is of a low signal-to-noise ratio (e.g., Fig. 2a), the performance may decrease *w.r.t.* to σ_R as can be seen in the yellow curve in Fig. 11b. A detailed discussion of noisy images is presented in Sect. 3.5.

3.5 Robustness to Noise

Most of the tested images in Fig. 2 are of relatively high signal-to-noise ratio (SNR) except for the artificial image in Fig. 2a. The experimental results presented in Sects. 3.3 and 3.4 reveal that the noise in this artificial image may significantly deteriorate the performance of the proposed fast bilateral filtering algorithm. This section thus presents a detailed analysis of its robustness to the noise. Four different levels of synthetic white Gaussian noise[3] were added to the images in Fig. 2 before testing, and the mean PSNR values *w.r.t.* parameter σ_S and σ_R are presented in Fig. 12 when different types of quantization are employed.

Figure 12a presents the PSNR values *w.r.t.* to four different noise levels when only range quantization is employed. Note that the performance is relatively robust to different noise levels and achieves the highest PSNR when the noise level is close to σ_R. This performance shows that range quantization is robust to image noise. This conclusion has indeed already been demonstrated in Figs. 7 and 8. Note that Fig. 2 contains images of both high SNR and low SNR. However, the performance in Figs. 7 and 8 are very consistent, which shows that the performance is not strongly related to the image noise.

Figure 12b presents the PSNR values *w.r.t.* to four different noise levels when only spatial quantization is employed.

As expected, the performance always decreases when the noise level increases and the performance increases when spatial filter kernel (σ_S) increases. However, the performance decreases when σ_R increases if the image contains a significant amount of noise. This behavior is consistent with the yellow curve in Fig. 10b (which corresponds to the performance of the noisy artificial image in Fig. 2a when only spatial quantization is employed).

Figure 12c presents the PSNR values *w.r.t.* to four different noise levels when both range and spatial quantization are employed. It looks like a combination of Fig. 12a and b. The performance always increases *w.r.t.* parameter σ_S. However, when σ_S is fixed, the performance increases *w.r.t.* parameter σ_R (like the 2*nd* row in Fig. 12a) when the noise level is low. Nevertheless, a significant amount of noise will decrease the performance (like the 2^{nd} row in Fig. 12b).

Figure 12 demonstrates that the proposed range quantization technique is robust to noise while the spatial quantization technique is relatively vulnerable to noise. Nevertheless, Fig. 12 shows that the PSNR value is always higher than 40 dB when the noise level is lower than 0.12[3].

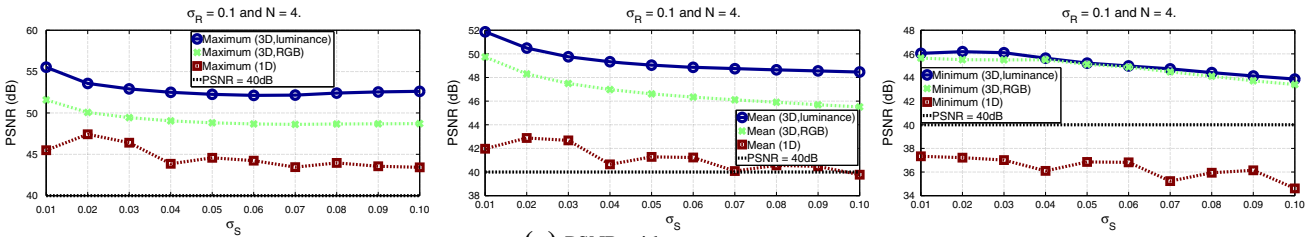
3.6 Multi-Channel Bilateral Filtering

If the guidance image T used to compute the range filter kernel is a n-channel image $\mathbf{T}$ where $n > 1$ (*e.g.* , a color image when $n = 3$), Eq. (9) becomes:

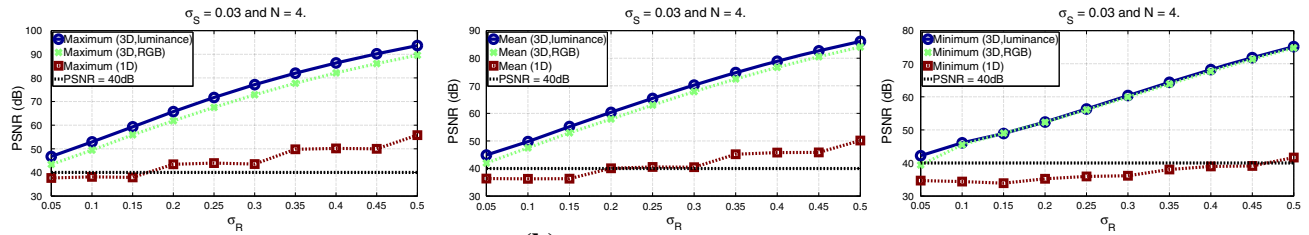
$$K_p = I_p^{\mathbf{T}} \cdot \omega_p = \sum_{q \in \Omega_p} 1 \cdot \mathcal{G}(\mathbf{T}_p, \mathbf{T}_q) I_q, \tag{17}$$

where $\mathbf{T}_p$ and $\mathbf{T}_q$ are n-channel vectors extracted from the guidance image at pixel p and q, respectively. For 8-bit images, $\mathbf{T}_p$ has up to 256^n potential vector values. Hence, the same as the one-channel case, multi-channel bilateral filtering can be formulated as a cost aggregation problem, except that the computational complexity will be exponential in the dimensionality n. In practice, the computational complexity can be reduced without introducing significant errors by downsampling the multi-dimensional space. Assuming that a total of N values are used in each dimension, the computational complexity will be $O(N^n)$ at each pixel location. However, the components of the multi-dimensional space are normally highly correlated; thus a relatively smaller N value can be used for a multi-channel bilateral filter as demonstrated in Fig. 13.

Similar to the one-channel case, a linear interpolation is performed after cost aggregation to obtain the final bilateral filtered value. The linear interpolation is straightforward in theory, except when memory cost is taken into account. Considering the one-channel bilateral filter again, let T_p denote the pixel value of the guidance image at pixel location p, L^l denote the l-th selected intensity level, ΔL denote the distance between two successive selected intensity levels, $\beta_p^l = \frac{|T_p - L^l|}{\Delta L}$ and:



(a) PSNR with respect to σ_S.



(b) PSNR with respect to σ_R.

Fig. 13 Numerical comparison between the proposed multi-channel joint bilateral filter and one-channel joint bilateral filter using the six color images in Fig. 2. For each of these color images, its luminance component is used as the input image. However, the guidance image is different. The *red dashed curves* use the luminance as the guidance image, the *green dash-dot curves* use the color image as the guidance image, and the *blue solid curves* use the luminance to form a three-channel image and use it as the guidance image (*the value is the same in every channel at a specific pixel location*). As can be seen, the multi-channel bilateral filter outperforms the one-channel bilateral filter when the same input image and parameter settings are used because the components of the multi-dimensional space are normally highly correlated. As a result, a relatively smaller N value can be used for the multi-channel bilateral filter. In practice, $N = 4$ can be used for color images as the minimum PSNR value (*shown as blue solid curves*) is over 40 dB even for very small σ_R values ($\sigma_R = 0.05$) (Color figure online)

$$\alpha_p^l = \begin{cases} 1-\beta_p^l & \text{if } \beta_p^l \le 1, \\ 0 & \text{else,} \end{cases}$$

the linear interpolation in Eq. (14) can be rewritten as follows:

$$I_p^T = \sum_{l=1}^{N} \alpha_p^l \frac{K_p^l}{\omega_p^l}. \tag{18}$$

Equation (18) can be interpolated as (1) initializing I_p^T with a zero value and then (2) accumulating it with respect to each intensity level. As a result, the memory cost is linear in the number of pixels as only two extra memory buffers are required to temporarily store K_p^l and ω_p^l and thus is independent of the number of intensity level N.

The memory cost for the multi-channel bilateral filter can be independent of the dimensionality n. Let $\mathbf{l} = \{l_1, l_2, \ldots, l_{n-1}, l_n\}$ denote an n-channel vector where l_i corresponds to the l_i-th selected intensity level in the i-th dimension, L^{l_i} denote the l_i-th selected intensity level in the i-th dimension, $\beta_p^{l_i} = \frac{|T_p - L^{l_i}|}{\Delta L}$ and:

$$\alpha_p^{l_i} = \begin{cases} 1-\beta_p^{l_i} & \text{if } \beta_p^{l_i} \le 1, \\ 0 & \text{else,} \end{cases}$$

the linear interpolation for n-channel bilateral filter with guidance image $\mathbf{T}$ can be presented as follows:

$$I_p^{\mathbf{T}} = \sum_{\mathbf{l}=\{1,1,\ldots,1,1\}}^{\{N,N,\ldots,N,N\}} \left(\prod_{i=1}^{n} \alpha_p^{l_i} \right) \frac{K_p^{\mathbf{l}}}{\omega_p^{\mathbf{l}}}, \tag{19}$$

where $K_p^{\mathbf{l}}$ and $\omega_p^{\mathbf{l}}$ are retrieved from the aggregated/filtered cost when the intensity level in the i-th dimension is l_i for any i where l_i is the i-th element of vector $\mathbf{l}$. Equation (19) can also be interpolated as (1) initializing $I_p^{\mathbf{T}}$ with a zero value and then (2) accumulating it with respect to each intensity level; thus the memory cost is independent of the dimensionality n.

4 Constant Time Weighted Median Filtering

The weighted median filter has many applications (Yin et al. 1996). This section extends the constant time median filtering framework proposed in Sect. 2 for constant time weighted median filtering. It is almost identical to the proposed median filtering framework except that the cost volume built from Eq. (2) will be aggregated/filtered with non-uniform statistical weights. For instance, filtering the cost volume using the constant time bilateral filter proposed in Sect. 3 results in a constant time bilateral weighted median filter. Other constant time filters like the guided image filter (He et al. 2010), domain transform filter (Gastal and Oliveira 2011), tree filter (Yang 2012a) and recursive bilateral filter (Yang 2012b) can also be used.

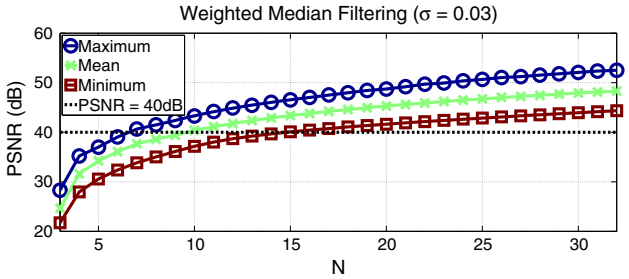


Fig. 14 Performance of the proposed weighted median filter *w.r.t.* N when $\sigma = 0.03$. As can be seen, the performance increases *w.r.t.* N and the minimum PSNR value (shown as a *red curve*) is always over 40 dB when $N \ge 16$ (Color figure online)

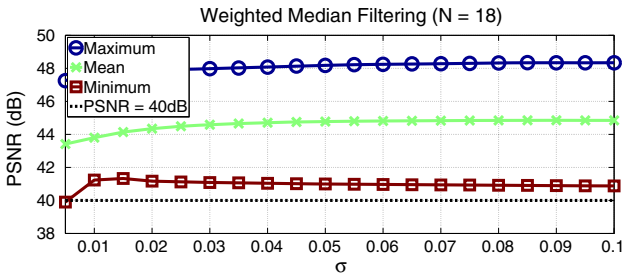


Fig. 15 Performance of the proposed weighted median filter *w.r.t.* σ when $N = 18$. As can be seen, the performance increases *w.r.t.* σ over all, and the minimum PSNR value (shown as a *red curve*) is always over 40 dB for all $\sigma \ge 0.005$ (Color figure online)

Similar to the constant time median filter presented in Sect. 2, the cost volume can be downsampled in the intensity domain to improve the efficiency in practice. A total of 256 intensity values are quantized into only a total of $N < 256$ bins, and the cost is computed at each bin and each pixel location. A quadratic interpolation step is used to upsample the intensity domain after cost aggregation. The accuracy of this approximation *w.r.t.* the number of quantization levels N is presented in Fig. 14. Note that the minimum PSNR value is always higher than 40 dB when $N \ge 16$; thus it is guaranteed that there will be very limited visible artifacts in any of the eight median filtered images when $N \ge 16$. Visual comparison between Figs. 14 and 3 shows that the performance of the weighted median filter is very similar to the median filter. The recursive bilateral filter Yang (2012b) ($\sigma_S = \sigma$ and $\sigma_R = 0.1$) is used to filter the cost volume and the eight images in Fig. 2 were tested in this experiment.

The performance *w.r.t.* the filter size is evaluated in Fig. 15. Similar to the median filter, the filter size is represented by a parameter $\sigma \in [0, 1]$. The number of quantization levels N is set to 18. Note that the performance increases *w.r.t.* σ over all, and the minimum PSNR value (shown as a red curve) is always over 40 dB for all $\sigma \ge 0.005$ when $N = 18$.

5 Computational Complexity

As discussed in Sect. 2 and Sect. 3, the proposed constant time median filter and bilateral filter can be decomposed into a number of independent $O(1)$ spatial filters like the box filter or recursive filter. According to our implementations, an $O(1)$ box filter takes about 4 ms to process a one megapixel grayscale image on a 3.2 GHz Intel Core i7 CPU. As a result, we use the number of $O(1)$ spatial filters to represent the computational complexity of the proposed $O(1)$ filters in Table 1.

The exact runtime is listed in Table 2. It is less meaningful due to the different implementation details, different operation systems and different PC configurations. However, it does show that the proposed method is comparable to the state-of-the-art, especially when spatial quantization is employed. Note that Perreault and Hebert (2007) is designed for 8-bit images and cannot be directly extended for weighted median filtering. Our method does not have this limitation.

Table 1 Computational complexity measured in the number of $O(1)$ filters

Method	Dim	Number of $O(1)$ Filters:	
		Box or Recursive	Adaptive weighted
Median	1	16	0
	3	48 (=16 × 3)	0
Bilateral	1	16 (=8 × (1 + 1))	0
	3	256 (=4^3 × (3 + 1))	0
Weighted Median	1	0	16
	3	0	48(=16 × 3)

6 Conclusion

A uniform framework for median filtering and bilateral filtering is presented. It enables the median filter and bilateral filter to be computed either exactly or approximately in constant time per image pixel. Unlike previous methods, the resulting bilateral filter can have arbitrary spatial and arbitrary range filter kernels. The proposed framework is also extended for constant time weighted median filtering.

Table 2 Exact runtime for processing a one megapixel image

Method	Dim	Runtime(ms/Mp)
Our box filter	1	**4**
Our exponential filter	1	**8**
Our gaussian filter	1	**12**
Our Median Filter ($N = 16$)	1	**83**
Perreault's Median Filter Perreault and Hebert (2007)	1	81
Our Box Bilateral ($N = 8$)	1	**125**
Our Box Bilateral+Spatial Quantization	1	**33**
Porikli Porikli (2008) (32 bins)	1	155
Porikli Porikli (2008)+Perreault and Hebert (2007)+*SSE* (32 bins)	1	40
Guided Filter He et al. (2010)	1	40
Our Box Bilateral ($N = 8$)	3	**2,432**
Our Box Bilateral+Spatial Quantization	3	**307**
Gaussian KD-Tree Adams et al. (2009) (invColorStdev=8)	3	>10,000
Permutohedral LatticeAdams et al. (2010) (SpatialStdev=4)	3	>1,000
Bilateral GridParis and Durand (2009) ($\sigma_s = 16, \sigma_r = 0.1$)	3	>500
Guided Filter He et al. (2010)	3	150
Guided Filter+Additional Guidance Image He et al. (2010)	3	300
Domain Transform Filter (NC)Gastal and Oliveira (2011)	3	160
Domain Transform Filter (RF)Gastal and Oliveira (2011)	3	60
Recursive Bilateral FilterYang (2012b)	3	43

Except for the proposed methods (*in bold*), the runtime reported by the author(s) or obtained from the code published by the author(s) is used. The default parameters or those suggested by the authors were used. The runtime of our bilateral filter reported in this table was computed with a small downsampling factor of 4 ($\frac{1}{4} \times \frac{1}{4}$ of the original spatial resolution) when spatial quantization is employed

Acknowledgments This work was supported in part by a GRF Grant from the Research Grants Council of Hong Kong (Project No. CityU 122212) and a grant from HP lab.

References

Adams, A., Baek, J., & Davis, A. (2010). Fast high-dimensional filtering using the permutohedral lattice. *Comput. Graph. Forum*, *29*(2), 753–762.

Adams, A., Gelfand, N., Dolson, J., & Levoy, M. (2009). Gaussian kd-trees for fast high-dimensional filtering. *ACM Trans. Graph.*, *28*, 21:1–21:12.

Ahuja, N. (1996). A transform for multiscale image segmentation by integrated edge and region detection. *PAMI*, *18*, 1211–1235.

Chen, J., Paris, S., & Durand, F. (2007). Real-time edge-aware image processing with the bilateral grid. In *Siggraph*, vol. 26.

Davis, L. S., & Rosenfeld, A. (1978). Noise cleaning by iterated local averaging. *IEEE Trans. Syst. Man, Cybernet. SMC*, *8*, 703–710.

Deriche, R. (1992). Recursively implementing the gaussian and its derivatives. In *ICIP* (pp. 263–267).

Durand, F., & Dorsey, J. (2002). Fast bilateral filtering for the display of high-dynamic-range images. In *Siggraph* vol. 21.

Eisemann, E., & Durand, F. (2004). Flash photography enhancement via intrinsic relighting. *Siggraph*, *23*(3), 673–678.

Elad, M. (2002). On the bilateral filter and ways to improve it. *IEEE Trans. Image Process.*, *11*(10), 1141–1151.

Fattal, R. (2009). Edge-avoiding wavelets and their applications. *ToG*, *28*(3), 1–10.

Gastal, E., & Oliveira, M. (2011). Domain transform for edge-aware image and video processing. *ToG*, *30*(4), 69:1–69:12.

He, K., Sun, J., & Tang, X. (2010). Guided image filtering. In *ECCV* (pp. 1–14).

Huber, P. J., & Wiley, J. (1981) *Robust statistics*. New York: Wiley InterScience

Kass, M., & Solomon, J. (2010). Smoothed local histogram filters. *ToG*, *29*(4), 100:1–100:10.

Paris, S., & Durand, F. (2009). A fast approximation of the bilateral filter using a signal processing approach. *IJCV*, *81*, 24–52.

Paris, S., Kornprobst, P., Tumblin, J., & Durand, F. (2009). Bilateral filtering: Theory and applications. *Found. Trends Comput. Graph. Vision*, *4*(1), 1–73.

Perreault, S., & Hebert, P. (2007). Median filtering in constant time. *Trans. Image Process.*, *16*, 2389–2394.

Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., & Toyama, K. (2004). Digital photography with flash and no-flash image pairs. *Siggraph*, *23*(3), 664–672.

Pham, T. Q., & van Vliet, L. J. (2005). Separable bilateral filtering for fast video preprocessing. In *Proceedings of the International Conference on Multimedia and Expo*.

Porikli, F. (2008). Constant time o(1) bilateral filtering. In *CVPR*.

Smith, J. O. (2007). *Introduction to digital filters with audio applications*. W3K Publishing. http://www.w3k.org/books/.

Tabb, M., & Ahuja, N. (1997). Unsupervised multiscale image segmentation by integrated edge and region detection. *TIP*, *6*, 642–655.

Tomasi, C., & Manduchi, R. (1998). Bilateral filtering for gray and color images. In *ICCV* (pp. 839–846).

Viola, P., & Jones, M. (2001). Robust real-time face detection. In *ICCV* (pp. 747–750).

Weiss, B. (2006). Fast median and bilateral filtering. In *Siggraph* (vol. 25, pp. 519–526).

Yang, Q. (2012). A non-local cost aggregation method for stereo matching. In *CVPR* (pp. 1402–1409).

Yang, Q. (2012). Recursive bilateral filtering. In *ECCV* (pp. 399–413).

Yang, Q., Wang, S., & Ahuja, N. (2010). Real-time specular highlight removal using bilateral filtering. In *ECCV* (pp. 87–100).

Yin, L., Yang, R., Gabbouj, M., & Neuvo, Y. (1996). Weighted median filters: A tutorial. *IEEE Trans. Circuits Syst. II*, *43*(3), 157–192.

Zucker, A., Lev, S., & Rosenfeld, A. (1977). Iterative enhancement of noisy images. *IEEE Trans. Syst. Man, Cybernet. SMC*, *7*, 435–441.