SEPARABLE BILATERAL FILTERING FOR FAST VIDEO PREPROCESSING

Tuan Q. Pham   *Lucas J. van Vliet*

Quantitative Imaging Group, Department of Imaging Science and Technology
Delft University of Technology
http://www.qi.tnw.tudelft.nl/~lucas

ABSTRACT

Bilateral filtering is an edge-preserving filtering technique that employs both geometric closeness and photometric similarity of neighboring pixels to construct its filter kernel. Multi-dimensional bilateral filtering is computationally expensive because the adaptive kernel has to be recomputed at every pixel. In this paper, we present a separable implementation of the bilateral filter. The separable implementation offers equivalent adaptive filtering capability at a fraction of execution time compared to the traditional filter. Because of this efficiency, the separable bilateral filter can be used for fast preprocessing of images and videos. Experiments show that better image quality and higher compression efficiency is achievable if the original video is preprocessed with the separable bilateral filter.

1. INTRODUCTION

Bilateral filtering is a term coined by Tomasi and Manduchi [12] in 1998 to refer to an edge-preserving filtering technique that takes both geometric closeness and photometric similarity of neighboring pixels into account. The edge-preserving capability is due to an implicit local mode selection as a result of intensity-selective filtering [3]. In fact, the idea of filtering in both spacial and tonal domains has been around since the sigma filter [8] in 1983. Smith and Brady [11] in 1995 added Gaussian weighting to the sigma filter and produced their own version of bilateral filter. The bilateral filter has also been shown [2, 3] to be equivalent to a number of known techniques such as anisotropic diffusion [10], local-mode finding [3] and mean-shift analysis [5]. Recently, Boomgaard and Weijer [4] extended bilateral filtering to a robust facet model to estimate local image structures under random noise and outliers.

Although the bilateral filter produces excellent filtered results, its application is limited by its speed. The filter's computational complexity is high and it increases exponentially with the number of dimensions. Equivalent filtering techniques such as anisotropic diffusion and mean-shift analysis are not fast either because of their iterative nature. A fast implementation of bilateral filtering is therefore desirable.

One first attempt to speedup the bilateral filter is a piecewise linear approximation proposed by Durrand and Dorsey [6]. The method is similar to the idea of channel filtering by Felsberg and Grandlund [7], in which the image is spatio-tonally filtered with respect to a number of intensity channels. The output image is then a pixel-wise interpolation of the channel responses. Although the computation of each channel response requires only two Gaussian convolutions, many channels are needed to cover the whole dynamic range (up to 17 in [6]). The gain in speed is therefore only apparent for very large kernels (kernel size > 25 pixels per dimension as shown in figure 11 in [6]). The piecewise-linear approximation is also not suitable for large images or videos because of excessive memory needed to store the temporary channel images.

In this paper, we propose a fast separable implementation of the bilateral filter. The image is bilaterally filtered in one dimension and the intermediate result is filtered again in subsequent dimensions. The separable implementation is not only fast but it also approximates the true bilateral filter reasonably well. Little memory overhead is required since the output image can be written direct onto the input's memory. Experiments on various images show that the separable implementation outperforms both the original and the piecewise-linear implementation in speed while achieving similar filtering results. As an application, the separable bilateral filtering is applied to noisy videos to improve image quality and coding efficiency of video compression systems.

2. BILATERAL FILTERING

Bilateral filtering performs a weighted average of local samples, in which higher weights are given to samples that are closer in both space and intensity to the center sample. The weighted average is done over a neighborhood S around the center sample $\mathbf{s}_0 = \{x_0, y_0\}$ whose intensity is $I(\mathbf{s}_0)$:

$$O(\mathbf{s}_0) = \sum_{\mathbf{s} \in S} f(\mathbf{s}, \mathbf{s}_0).I(\mathbf{s}) \bigg/ \sum_{\mathbf{s} \in S} f(\mathbf{s}, \mathbf{s}_0) \qquad (1)$$

(a) noisy edge   (b) bilateral filter   (c) filtered output

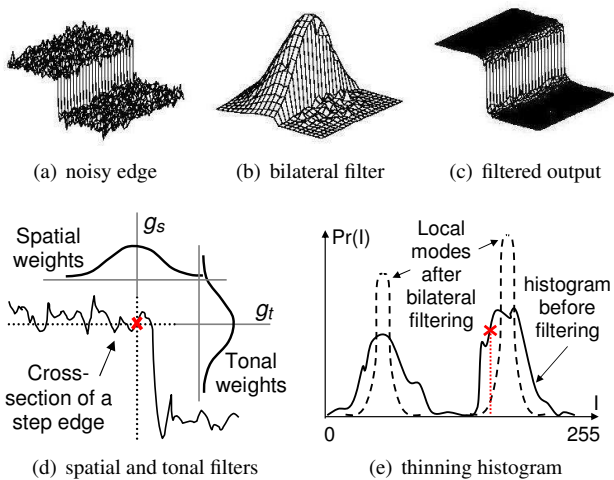(d) spatial and tonal filters   (e) thinning histogram

Fig. 1. Bilateral filtering as a mode seeking process.

where $f(\mathbf{s}, \mathbf{s}_0) = g_s(\mathbf{s} - \mathbf{s}_0) \cdot g_t(I(\mathbf{s}) - I(\mathbf{s}_0))$ is the bilateral filter for the neighborhood around $\mathbf{s}_0$. The constituent spatial and tonal weights g_s and g_t are Gaussian functions:

$$g_s(\mathbf{s}) = g(x, \sigma_s) \cdot g(y, \sigma_s) \qquad g_t(I) = g(I, \sigma_t) \qquad (2)$$

where $g(t, \sigma) = \frac{1}{\sigma \sqrt{2\pi}} e^{-t^2/2\sigma^2}$, σ_s and σ_t are the spatial and tonal scales of bilateral filtering.

Figure 1 depicts the formation of a bilateral filter centered at a pixel on one side of a noisy step edge (fig. 1a). The role of the spatial weight g_s is to limit the spatial extend of the filter operation. It accounts for the bell shape of the bilateral filter. The tonal weight g_t suppresses the contributions of pixels from the other side of the edge. It is responsible for the truncation of approximately half of the Gaussian bell (fig. 1b). Since only pixels sharing similar intensity with the current pixel have significant weights in the local analysis, the edge is not diffused across and noise is effectively suppressed (fig. 1c). A compact bi-modal histogram of the filtered edge compared to that of the noisy edge in figure 1e further confirms this noire reduction and edge sharpening. As pointed out in [3], bilateral filtering is a first iteration of an iterative local mode finding process. If the bilateral filter is applied several times, the filtered result will eventually converge to a sharp edge transition and the local histogram will reduce to two spikes at the intensity levels on either sides of the edge.

3. SEPARABLE BILATERAL FILTERING

Gaussian filtering is very fast because it can be implemented in a separable way (i.e. the image is first filtered in the x-dimension, followed by a filtering in the y-dimension and so on...). A recursive implementation of the separable Gaussian filter, for example, only requires a fixed number of operations per pixel irrespective of the Gaussian scale [14] ($O(Nd)$, where N is the total number of pixels in the image, and d is the image dimensionality). Unfortunately, the bilateral filter $f(\mathbf{s}, \mathbf{s}_0)$ in (1) is not separable due to the intensity-dependent component g_t. However, when implemented in a separable way, the new filter still satisfies all desired requirements: noise reduction and signal preservation.

Similar to separable Gaussian filtering, a one-dimensional bilateral filter is applied to the first dimension and the intermediate result is filtered again in subsequent dimensions. In this way, the computational complexity of the separable implementation is just $O(Nmd)$ compared to $O(Nm^d)$ for a full kernel implementation (where m is the size of the filter in each dimension). With a modest filter size of 7, for example, the speed improvement of the separable filter over the original filter could reach 3.5 times for 2D and 16 times for 3D data.

Not sacrificing performance for speed, the separable bilateral filter is also a good approximation to the original filter. Similar to a separable median filter [9], the separable bilateral filter first seeks the local mode along the x-dimension, followed by a mode seeking in the y-dimension. Because of the non-separability of $f(\mathbf{s}, \mathbf{s}_0)$ in (1), different orderings of filtering could result in a slightly different output. However, the difference is often small for normal Signal-to-Noise Ratios[1] (SNR $\geq 20dB$). Also, due to the separable filtering along the sampling axes, the proposed filter approximates the original filter better for image patches whose dominant orientation aligns with the sampling grid. Nevertheless, even in a worst case scenario of a $45°$-tilt step edge in figure 2, the separable filter still performs noise reduction and edge preservation very well.

3.1. Kernel approximation in 2D

We show how an effective kernel of the separable bilateral filtering looks like on the verge of a $45°$-tilt step edge. Images on the top row of figure 2 show an input patch and its separable filter kernels as gray-scale images (brighter pixel mean higher filter weight and vice versa). The center pixel lies just on the white side of the edge in the middle of the image in figure 2a. Since we are interested in the effective kernel centered at this pixel, intermediate 1D kernels in the x- and y-dimensions are shown in figure 2b-d.

In the first step, a horizontal bilateral filtering is applied to all rows of the input image. The x-kernels centered at pixels in the middle column are shown in figure 2b. As can be seen in figure 2b, bilateral filtering assigns low weights to the pixels far away from the center position and pixels with a large intensity difference from the center intensity (which changes from high to low as the center pixel moves from top of the column to the bottom).

---

[1]SNR=$10 \log_{10}(\sigma_I^2/\sigma_n^2)$; where σ_I^2, σ_n^2 are variances of signal, noise

(a) input   (b, c, d) x-, y- & xy-kernels   (e) 2D kernel
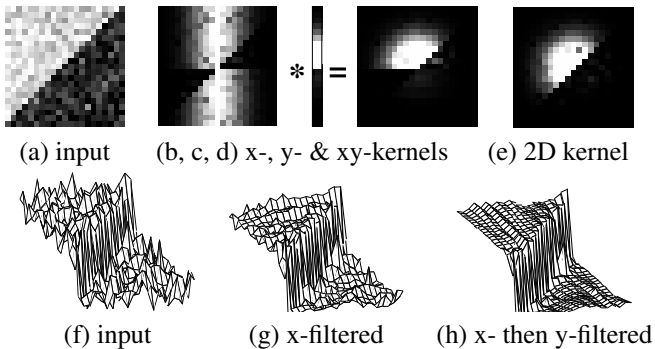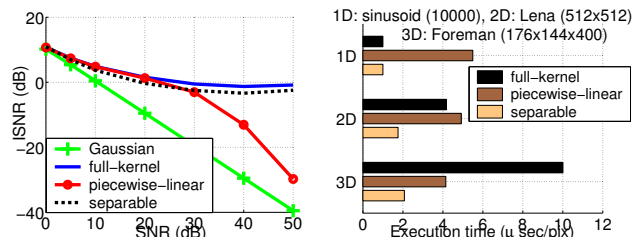
(f) input   (g) x-filtered   (h) x- then y-filtered

Fig. 2. Separable bilateral filtering as an approximation of the original filter [12]. Top: kernels. Bottom: filtered outputs

(a) Improvement in SNR of Gaussian and bilateral filters on Foreman

(b) Execution time in Matlab on AMD 1.47 GHz with 1 GB of RAM

Fig. 3. Performance of separable bilateral filtering.

In the second separable step, a vertical bilateral filtering is applied to the previously filtered result. The y-kernel for the center pixel is shown as a single image column in figure 2c. Again, note that bilateral filtering prevents the low intensity pixels to contribute to the high intensity output by giving them low weights. Multiplying the columns in the x-kernels image and y-kernel image, we obtain the effective kernel in figure 2d.

Compared to the full 2D bilateral kernel in figure 2e, the effective separable kernel in figure 2d gathers fewer samples for the local weighted average. This is due to a truncation of the lower half of the y-kernel. However, the effective kernel still gathers enough pixels of the correct intensity mode for the filtering. The separable bilateral filter therefore still performs noise reduction and edge preservation well. The filtered results at the bottom of figure 2 confirm this. After a bilateral filtering in the x-dimension, figure 2g is already less noisy than the input (fig. 2f). A subsequent bilateral filter in the y-dimension reduces noise even further while still keeping the edge sharp (fig.2h).

3.2. Reduction of edge jaggedness

One undesirable effect of bilateral filtering as a mode seeking algorithm is that edge transitions are often too abrupt after filtering. This results in a cartoon-like appearance of the filtered images [12]. It also hampers later processing tasks such as iso-surface extraction and image compression. Moreover, high noise could bias pixels on the edge towards a wrong local mode, causing noticeable edge jaggedness. To avoid these unwanted effects, we limit the edge sharpening by imposing a minimum smoothing in the filter kernel. In the 1D separable filter, this is done by forcing the values of two immediate middle taps to be greater than or equal to α-times that of the center tap (we choose $\alpha = 0.25$ for a minimum smoothing equivalent to a low-pass filtering by $\frac{1}{6}[1\ 4\ 1]$). Note that this trick only smoothes out very sharp edges, whereas image patches with gradual intensity varia-

tions are not affected.

4. EXPERIMENTS

4.1. Performance of separable bilateral filter

To quantitatively show that the separable implementation can approximate the original bilateral filter well, we compare the results of all implementations (full-kernel, piecewise-linear, separable) on three images: an 1D sinusoid pattern (10000), a 2D Lena image (512×512) and a 3D Foreman video ($176 \times 144 \times 400$, see fig.5) at different SNRs. Filter sizes of 9 for 1D, 9×9 for 2D and $9 \times 9 \times 5$ for 3D with spatial scale $\sigma_s = 2$, temporal scale = 1, and tonal scale $\sigma_t = 3\sigma_n$ are used for all implementations. The Improvement in SNR of a filtered image $\hat{f}$ over a noisy image g with respect to a noise-free original f is defined as:

$$ISNR = 10\log_{10}\left(\sum(f-g)^2 \Big/ \sum(f-\hat{f})^2\right) \quad (3)$$

As can be seen in figure 3a, the ISNR of separable bilateral filtering closely follows that of the full-kernel implementation. The piecewise-linear implementation does not perform very well at high SNRs due to errors in channel interpolation. Finally, the ISNR of Gaussian filtering is worse than any implementations of the bilateral filter.

The execution time chart in figure 3b shows that the separable filter is the fastest amongst all three implementations: full-kernel, piecewise-linear and separable. At the processing speed of 2 μsec per pixel for 3D images, the separable method is two-times faster than the piecewise-linear method. Note that we have speeded up the piecewise-linear filter in [6] with several enhancements: tile processing to reduce memory consumption (63 tiles for the 3D Foreman sequence), non-uniform channel selection (8 channels on average). Two-times downsampling of the channel responses is used because the spatial scale is small ($\sigma_s = 2$). The execution time of separable bilateral filtering is linearly proportional to the number of dimensions, whereas the relationship is exponential for the full-kernel implementation.
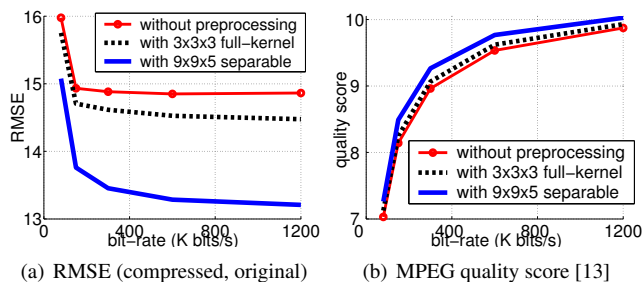
(a) RMSE (compressed, original) (b) MPEG quality score [13]

Fig. 4. Quality of compressed Foreman with bilateral filter

(a) without preprocessing (b) with bilateral pre-filtering

Fig. 5. Frame 31 of Foreman MPEG1 video at 150 Kbits/s

4.2. Preprocessing for efficient compression

Preprocessing is often used to improve visual quality and coding efficiency of video compression systems. Due to timing constraints, however, only fast filters such as window averaging filters are used. The problem with this type of low-pass filters is that they blur out small details and edges. Averaging along the temporal axis also results in ghost effects around moving objects. Bilateral filtering resolves all these problems by an intensity-selective averaging. The separable implementation is fast enough to be incorporated into any video compression schemes. Similar to [12], the separable filter is applicable to color images if the tonal weight g_t in (2) is computed from an Euclidean distance in the CIE-Lab color space.

We show that better compressed video is achievable if the separable bilateral filter is used in the preprocessing step. In this experiment, the original QCIF Foreman sequence[2] ($176 \times 144 \times 400$) is compressed using an MPEG-1 encoder [1]. The compressed videos with and without preprocessing are compared in terms of Root Mean Squared Error ($RMSE = \sqrt{\sum(f - \hat{f})^2}$) with the original video and an MPEG quality score. The MPEG quality score, being an averaged JPEG quality score [13] of all frames, ranges from 0 to 10 with higher values mean less blocking artifacts. As can be seen from figure 4, the MPEG sequence with separable bilateral preprocessing has a smallest RMSE and a highest MPEG quality score. Full-kernel bilateral preprocessing with the same CPU requirements does improve the video quality. However, the improvement is minimal due to a small kernel size ($3 \times 3 \times 3$ full-kernel compared to $9 \times 9 \times 5$ separable). The 31st frame of a Foreman sequence compressed at 150 Kbits/s can be seen in figure 5 where the video without preprocessing on the left is clearly more blocky than the video with separable bilateral pre-filtering on the right. This once again confirms a higher quality curve of the video with preprocessing in figure 4b.

5. CONCLUSIONS
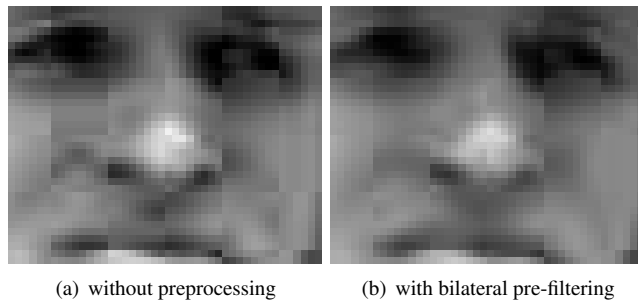
In conclusion, we have presented a separable implementation of the bilateral filter. The separable implementation is fast and is a good approximation of the original bilateral filter. When used as a preprocessing step, the separable bilateral filter helps increasing the coding efficiency and visual quality of MPEG video compression systems.

6. REFERENCES

[1] avi2mpg1. http://home.cogeco.ca/~avi2vcd.

[2] D. Barash. A fundamental relationship between bilateral filtering, adaptive smoothing and the non-linear diffusion equation. *PAMI*, 24(6):844–847, 2002.

[3] R. v.d. Boomgaard and J. v.d. Weijer. On the equivalence of local-mode finding, robust estimation and mean-shift analysis as used in early vision tasks. In *Proc. of ICPR*, pages 927–930, Quebec, Canada, 2002.

[4] R. v.d. Boomgaard and J. v.d. Weijer. Least squares and robust estimation of local image structure. In *Proc. of Scale-Space*, pages 237–254, 2003.

[5] D. Comaniciu and P. Meer. Mean shift analysis and applications. In *ICCV*, pages 1197–1203, 1999.

[6] F. Durrand and J. Dorsey. Fast bilateral filtering for the display of high dynamic range images. In *Proc. of SIGGRAPH'02*, pages 844–847, 2002.

[7] M. Felsberg and G. Granlund. Anisotropic channel filtering. In *Proc. of SCIA'03, LNCS 2749*, pages 755–762, 2003.

[8] J.-S. Lee. Digital image smoothing and the sigma filter. *CVGIP*, 24:255–269, 1983.

[9] P. M. Narendra. A separable median filter for image noise smoothing. *PAMI*, 3(1):20–29, 1981.

[10] P.Perona and J.Malik. Scale-space filtering and edge detection using anisotropic diffusion. *PAMI* 12(7):629–639, 1990.

[11] S. Smith and J. Brady. SUSAN-a new approach to low level image processing. Tech. Rep. TR95SMS1c, Oxford, 1995.

[12] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. of ICCV*, pages 839–846, USA, 1998.

[13] Z. Wang, H. Sheikh, and A. C. Bovik. No-reference perceptual quality assessment of JPEG compressed images. In *Proc. of ICIP*, pages 477–480, 2002.

[14] I. Young, L. van Vliet, and M. van Ginkel. Recursive Gabor filtering. *Signal Processing*, 50(11):2798–2805, 2002.

[2]available at http://trace.eas.asu.edu/yuv/qcif.html