Fast Bilateral Filtering Using Recursive Gaussian Filter for Tone Mapping Algorithm

Primastuti Dewi* · Jin-woo Nam* · Eui-Young Cha*

*Pusan National University

E-mail : {primastuti, junimnjw, eycha}@pusan.ac.kr

ABSTRACT

In this paper, we propose a fast implementation of Bilateral filter for tone mapping algorithm. Bilateral filter is able to preserve detail while at the same time prevent halo-ing artifacts because of improper scale selection by ensuring image smoothed that not only depend on pixel closeness, but also similarity. We accelerate Bilateral filter by using a piecewise linear approximation and recursive Gaussian filter as its domain filter. Recursive Gaussian filter is scale independent filter that combines low cost 1D filter which makes this filter much faster than conventional convolution filter and filtering in frequency domain. The experiment results show that proposed method is simpler and faster than previous method without mortgaging the quality.

Keyword

Bilateral filter, recursive Gaussian filter, tone mapping algorithm

Ⅰ. Introduction

Human visual system (HVS) has a feature that adaptively processes the scene independent from surround illumination which helps us to identify object and color. However, camera or other low-dynamic-range (LDR) devices do not have this feature, so what the digital camera captures may look different from what we see directly with our eyes. Specially for high-dynamic-range (HDR) scene, as in figure 1, where the dark and bright area appear at the same time in the scene. One of that regions must be either under-exposed or over-exposed. A technique to map colors from HDR scene into LDR device by reducing the contrast of scene value is called *tone mapping*.

Here, we present a tone mapping algorithm using a non-linear edge-preserving smoothing algorithm, Bilateral filter. Using edge-preserving smoothing filter in tone mapping process helps reducing the contrast while preserve the detail part, prevent halo-ing artifacts along the edge or high gradient region which usually appear if we use general smoothing filter, also produce more natural image [1, 2].

Unfortunately, the cost of this filter is computationally high because of its spatial convolution which use both spatial information and pixels value differences. We propose fast Bilateral filtering using linear piecewise approximation and recursive Gaussian filter as its domain filter to overcome this problem.

Figure 1. Example of HDR image

Section 2 and 3 will cover more detail description about Bilateral filter and recursive Gaussian filter respectively. We will describe our proposed method that has both advantages of Bilateral filter and recursive Gaussian filter in section 4. Section 5 will include explanation and experiment results of tone mapping algorithm using our proposed method.

Ⅱ. Bilateral Filter

Bilateral filter, developed by Tomasi and Maduchi, is a filter that smooths image not only by its geometric closeness but also by its photometric similarity as in equation (1) [1].

$$I_{out}(x)=\frac{1}{k(x)}\int_{-\infty}^{\infty}c(\zeta)\,s(I_{in}(x+\zeta)-I_{in}(x))\,I_{in}(x) \quad (1)$$

where

$$k(x)=\int_{-\infty}^{\infty}c(\zeta)\,s(I_{in}(x+\zeta)-I_{in}(x)) \quad (2)$$

Output value for pixel x, $I_{out}(x)$, is an integral of input pixel $I_{in}(x)$ multiply domain filter value and range filter value. Domain filter $c()$ and range filter $s()$ can use any smoothing filter, but Gaussian filter is widely used.

The domain filter input value is distance of the current neighbor pixel to center pixel. This means as the pixel gets farther from the center pixel it gives less contribution. Range filter input value is the difference of current neighbor pixel intensity and center pixel intensity, which means as the intensity is similar to center pixel intensity, the value get higher, but if the intensity is greatly different, the value is almost zero. This filter has a role to preserve the edge. Edge region's intensity usually change greatly, so the range filter value is almost zero, which means less blurring in this region.

Normalize function $k()$ is to ensure that the output value keep in the range.

Durand and Dorsey, in their paper, show that Bilateral filter is simpler but as robust as previous well-known edge-preserving smoothing filter, such as Anisotropic diffusion [1].

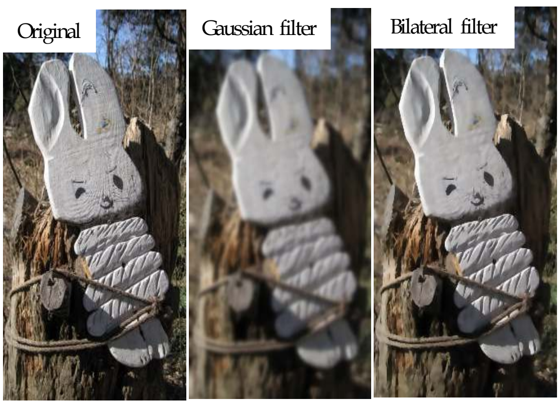

Figure 2. Result of smoothing filter

Ⅲ. Recursive Gaussian Filter

Recursive implementation of Gaussian filter takes $O(N^2)$ for N^2-size image, much faster than direct convolution which complexity rises to $O(N^4)$ for large file and convolution using the fast fourier transform (FFT) which has complexity time approximately $O(N^2 Log_2 N)$ as showed in table 1 [4, 5]. This method consisted of 6 MADDs per dimension and the complexity is independent of the scale value in Gaussian filter.

Table 1. Filter complexity comparison

Method (Size : N^2)	Computation cost
Direct convolution (Filter size : L^2, Border : B)	$(N+B)^2L^2(A+M)$
Fast Fourier Transform (FFT)	$7N^2LogN(A+2M)$
Recursive filter	$(N+B)^2+(16M+12A+4D)$

This method using two low cost 1D filter than high cost 2D filter for processing 2D signal such as image. In each 1D filter process, the input data is filtered in forward direction then filtered backward direction as in equation (3) and (4).

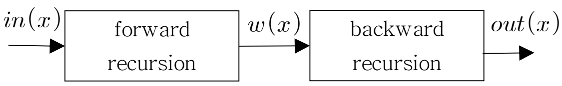

Figure 3. 1D filter in recursive Gaussian filter

$$w(x)=B.in(x)+(b_1.w(n-1))+(b_2.w(n-2))+(b_3.w(n-3))/b_0 \quad (3)$$

$$out(x)=B.w(x)+(b_1.out(n-1))+(b_2.out(n-2))+(b_3.out(n-3))/b_0 \quad (4)$$

$w()$ is output of forward recursion and $out()$ output of filter. B, b_0, b_1, b_2, b_3 is value derived from the scale of Gaussian scale, which can be obtained using equation (5) and (6).

$$q=\begin{cases}0.98711\times\sigma-0.96330 & \sigma\geq 2.5\\ 3.97156-4.14554\times\sqrt{1-0.26891\times\sigma} & 0.5\leq\sigma\leq 2.5\\ 0.1147705182035522469375 & otherwise\end{cases} \quad (5)$$

$$\begin{aligned}b_0&=1.57825+(2.44413\times q)+(1.4281\times q^2)+(0.422205\times q^3)\\ b_1&=(2.44413\times q)+(2.85619\times q^2)+(1.26661\times q^3)\\ b_2&=-((1.4281\times q^2)+(1.26661\times q^3))\\ b_3&=(0.422205\times q^3)\\ B&=1.0-((b_1+b_2+b_3)/b_0)\end{aligned} \quad (6)$$

Ⅳ. Fast Linear Piecewise Bilateral Filter Using Recursive Gaussian

Here, we proposed fast implementation of Bilateral filter using piecewise linear and recursive Gaussian filter.

Different from general convolution which can be accelerated using multiplication in frequency domain, Bilateral filter cannot be accelerated using this method because this filter is not simple convolution since range filter is a signal dependent filter.

Similar to method proposed in [1], we use piecewise linear approximation to accelerate the Bilateral filter. First, possible signal intensity is discretized into NB_SEGMENT, $\{i^j\}$. Then we compute the linear filter using equation (7) and (8).

$$J^j(x+\zeta) = \frac{1}{k^j(x+\zeta)} \int_{-\infty}^{\infty} c(x+\zeta)\, s(I(x+\zeta) - i^j) I(x+\zeta)$$
$$= \frac{1}{k^j(x+\zeta)} \int_{-\infty}^{\infty} c(x+\zeta)\, R^{\,j}_{x+\zeta} \qquad (7)$$

when

$$k^j(x+\zeta) = \int_{-\infty}^{\infty} c(x+\zeta)\, S^j(x) \qquad (8)$$

as we consider the output value is convolution $R^{I(x)} : (x+\zeta) \rightarrow s(I(x+\zeta) - I(x)) I(x+\zeta)$ by $c()$ and the normalization function is also convolution $S^{I(x)} : (x+\zeta) \rightarrow s(I(x+\zeta) - I(x))$ by $c()$. For $c()$ we use recursive Gaussian filter (explained in section 3), which is faster than multiplying in frequency domain, to perform Gaussian filter.

The output value is linear interpolation of $J^j(\zeta)$ of the two closest value i^j of $I(s)$.

```
Piecewise Bilateral filter
(Image I, domain filter c_σd, range filter r_σr)
J=0                          /*set output to zero*/
for j=0 ... NB_SEGMENT
    i^j=min(I)+j×(max(I)-min(I))/NB_SEGMENT
    S^j=r_σr(I-i^j)/*r_σr for each pixel*/
    K^j=S^j ⊗ c_σd           /*normalization factor*/
                             /*smoothing using*/
                             /*recursive Gaussian*/
    R^j=S^j×I  /*H for each pixel*/
    R*^j=R^j ⊗ c_σd
    J^j=R*^j/K^j  /*normalize*/
    J=J+J^j×InterpolationWeight(I,i^j)
end of 'for' statement
```

Figure 4. Pseudo-code of proposed method

The pseudo-code of the method is presented in figure 6. Symbol ⊗ means convolution and × means pixel to pixel multiplication.

Ⅴ. Tone Mapping Algorithm Using Proposed Method

Tone mapping algorithm that will be used here is presented in [1], but we enhance the processing speed using our proposed method that already described in previous section. The tone mapping algorithm can be presented by the diagram in figure 7.

The input is image intensity, and for color image we separate the illuminance and the chromatic element first and recombine it after tone mapping process. We use the log of pixel intensity because it correspond directly to contrast.

In this method, the image is decomposed into 2 layers, *base* layer which represents image illuminance and *detai*l layer which represents reflectance or detail part of the image.

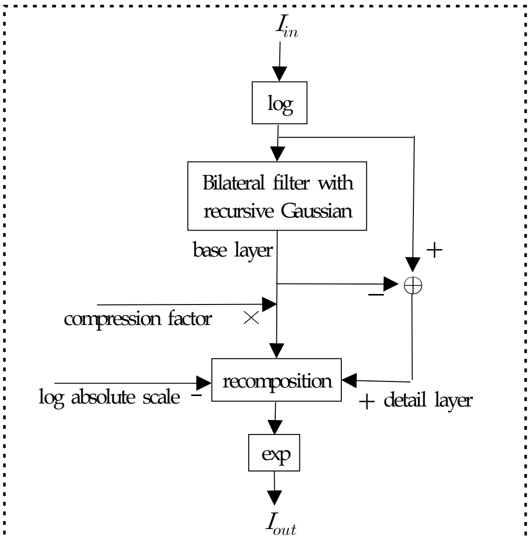

Figure 5. Proposed tone mapping diagram

Base layer is calculated using Bilateral filter and detail layer is image intensity in log domain subtracted by base layer. After compressing the base layer we recombine it with detail layer. This technique is to compress the contrast while preserves the detail. Exponential function is to set the output value back to the image value range.

We use 2% of image size for σ_d, σ_r = 0.4, compression factor = target contrast/(max(log(base))-min(log(base))), target contrast = log(5), log absolute scale = max(log(base))× compression factor, same as in [1] and [7] since the experiment results show satisfy output image with shorter processing time.

Figure 6. Result image using proposed method

Figure 7. Result image using proposed method

Figure 8. Result image using proposed method

Figure 9. Result image using proposed method

We tested our method to several images on a 3 GHz Intel Core Duo computer and the processing time is showed in table 2.

Table 2. Processing time comparison

Image	Image Size	Timing (s)
Figure 1	1600 x 1200	6.78
Figure 6	640 x 480	1.09
Figure 7	1024 x 682	2.59
Figure 8	1120 x 840	6.79
Figure 9	2048 x 1536	13.26

VI. Conclusion

In our proposed method, we combine both the advantages of edge-preserving smoothing filter, Bilateral filter, which fit to reduce image contrast while save the detail part, and fast scale-independent recursive Gaussian filter. We combine this to filter by applying recursive Gaussian filter as domain filter in piecewise linear approximation of Bilateral filter.

Tone mapping experiments of some HDR images using our proposed show good result and fast processing time.

References

[1] F. Durand, J. Dorsey, "Fast Bilateral Filtering for the Display of High-Dynamic-Range Images", ACM Transactions on Graphics, special issue on Proc. of ACM SIGGRAPH 2002, San Antonio, Texas, vol.21(3), pp. 249-256, 2002.

[2] P. Perona, J. Malik, "Scale-space and edge detection using anisotropic diffusion", IEEE PAMI 12, 7, pp. 629-639, 1990.

[3] C.Tomasi, R. Manduchi, "Bilateral filtering of gray and colored images", Porc. IEEE Intl. Conference and Applications, vol. 13(6), pp. 42-48, 1993.

[4] I. T. Young, L. J. van Vliet, "Recursive Implementation of the Gaussian Filter", Signal Processing, Vol. 44, pp. 139-151, 1955.

[5] Qiang Zhou, John P. Oakley, "Low cost implementation for the Multi-scale Retinex filter," Visual Media Production, 2006. CVMP 2006. 3rd European Conference on, pp. 65-70, 2006.

[6] http://people.csail.mit.edu/fredo/PUBLI/Siggraph2002, accessed on April 26th 2010.