Texture Filtering of Image

Professor: Hong Qin

Yishuo Wang

108533945

Abstract

In computer graphics, image texture filtering technology or texture smoothing technology is the method used to determine the texture color for a texture mapped pixel, using the colors (RGB elements of a pixel) of neighbor pixels. Due to this topic is not a brand new one, with the development of the technology in computer graphic area, people try and build many methods to filter the texture of the image. Some of them are died out because of their un-efficient, but these methods seriously built a foundation for future better idea. In this paper, we present three new and popular texture filtering method, Bilateral Filter*[1],* Guided Image Filtering*[2], and* Texture Filter with Interval Gradient*[3]. Finally, we will do the comparison to show the strong and weak points of each one.*

I. Introduction

Most applications in computer vision and computer graphics involve image filtering to suppress and/or extract content in images. Simple linear translation-invariant (LTI) filters with explicit kernels, such as the mean, Gaussian, Laplacian, and Sobel filters [4],have been widely used in image restoration, blurring/sharpening, edge detection, feature extraction, etc. On the other hand, some complicated method(complexity is not linear) also stand a important position in the computer graphics filtering area, like Bilateral Filter[1] due to their speciality.

While we are doing structure-texture decomposition, an image is decomposed as $I = S+T$, where I, S, and T represent the input image, structure elements and texture details, respectively. In general, the structure-texture decomposition problem is formulated as finding an appropriate (or latent) structure S by suppressing texture details T in the input image I.[3] Due to its usefulness in a broad range of image processing applications, many ideas have been proposed by many professional experts.

In paper "*Bilateral Filtering for Gray and Color Images*", Tomasi and Manduchi first time showed to public that smoothing images while preserving edges by means of a nonlinear combination if nearby image value.[1] And in paper "*Guided image filtering*", author proposes a novel explicit image filter called guided filter using a guided image. As for the newest paper "*Decomposition of Images with Interval Gradient*", author was motivated by "*Guided image filtering*", and created an "interval gradient" to filter the image. In this paper, I will show the theories of 3 method and the connection of them.

II. Bilateral Filter

Compared with other two method, *Bilateral Filter* was proposed earliest, which was in 1998. However, because of its advantage, many new methods have done the comparisons with it. The idea is following.

A low-pass domain filter applied to image $\mathbf{f}(x)$ produces an output image defined as follows, which is also the main function of the *Bilateral Filter*:

$$\mathbf{h}(x) = k_d^{-1}(\mathbf{x}) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{f}(\xi) e(\xi, \mathbf{x}) d\xi \quad (1)$$

where $e(\xi, \mathbf{x})$ measures the *geometric* closeness between the neighborhood center $\mathbf{x}$ and a nearby point ξ. The bold font for $\mathbf{f}$ and $\mathbf{h}$ emphasizes the fact that both input and output images may be multiband[1]. If low-pass filtering is to preserve the dc component of low-pass signals we obtain the spatial kernel

$$k_d(\mathbf{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e(\xi, \mathbf{x}) d\xi \quad (2)$$

In the equation (1), $\mathbf{h}$ is the output image, $\mathbf{f}$ is the input image. $\mathbf{h}$(x) and $\mathbf{f}$(x) means the data of the that pixel(Red, Green, Blue,$0 - 255$). This equation can be used in both of the gray image case(1 channel) and color image case(3 channels). $\mathbf{x}$ is each pixel, and ξ represents the neighbor pixel of the $\mathbf{x}$.

If the filter is shift-invariant, $e(\xi, (x))$ is only a function of the vector difference $\xi - \mathbf{x}$, and k_d is constant.

Then, range filtering kernel is similarly defined:

$$\mathbf{h}(x) = k_r^{-1}(\mathbf{x}) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{f}(\xi) s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x})) d\xi \quad (3)$$

Thus, the similarity function s operates in the range of the image function $\mathbf{f}$. The equation will change to

$$k_r(\mathbf{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x})) d\xi \quad (4)$$

The $e(\xi, \mathbf{x})$ in equation (1), (2) is a function of the vector difference $\xi - \mathbf{x}$. Corresponding to this explanation, $s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))$ means $\mathbf{f}(\xi) - \mathbf{f}(\mathbf{x})$.

Due to both of spatial kernel and range filtering kernel have their limitation, author try to combine them, and get the following equation.

$$\mathbf{h}(x) = k^{-1}(\mathbf{x}) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{f}(\xi) e(\xi, \mathbf{x}) s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x})) d\xi \qquad (5)$$

with the following normalization k,

$$k(\mathbf{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e(\xi, \mathbf{x}) s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x})) d\xi \qquad (6)$$

Therefore, combined domain(spatial) and range filtering kernel together will be demoted as *bilateral filtering*. Intuitively, we can show the relation on a picture[5].(Figure 1)

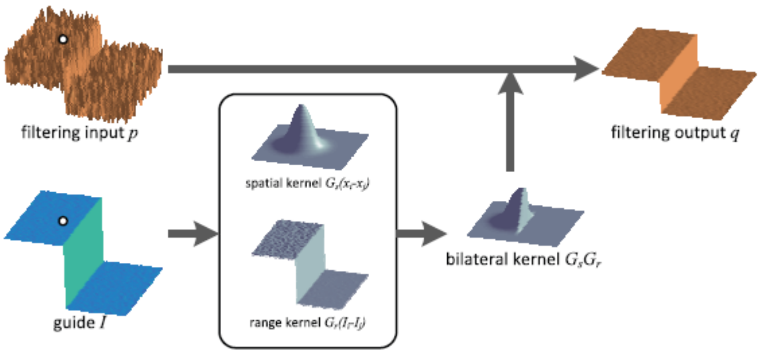

Figure 1: Bilateral Filter

Also, Figure 2 shows us the result of the *Bilateral Filter*.

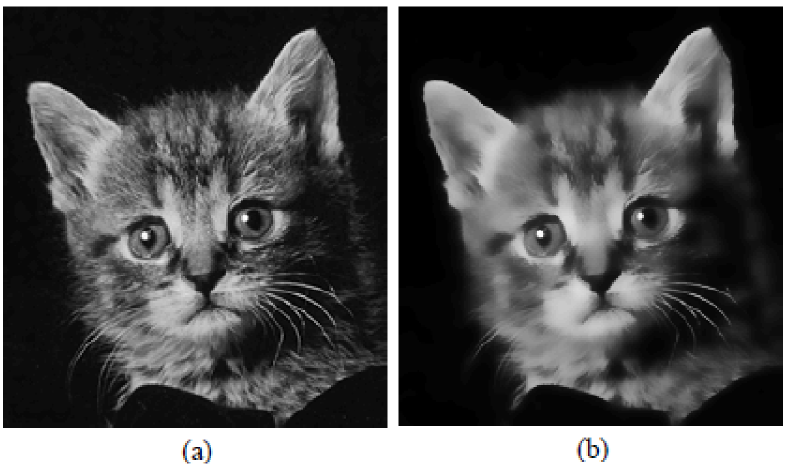

Figure 2: A picture before (a) and after (b) bilateral filtering.

III. Guided Image Filtering

The goal of the *Guided Image Filter* certainly is for filtering the texture of the image. However, the new thing in this method is that we have two input. Paper performs edge-preserving smoothing on an input image **p**, using the content of guided image **I** (second input data), to influence the filtering. The guided image **I** can be the input image **p** itself, a different version of the image, or a completely different image. *Guided image filtering* is a neighborhood operation, and the local window size is 3 by 3.
The main function of the *Guided Image Filter* is

$$\begin{cases} q_i &= p_i - n_i \quad (1) \\ q_i &= aI_i + b \quad (2) \end{cases}$$

Here, q_i is the i^{th} output pixel, p_i is the i^{th} input pixel. n_i is the noise part of the i^{th} pixel, so it is the part of the p_i, and I_i is the i^{th} pixel of the guide image. Figure 3 show its relation[2].

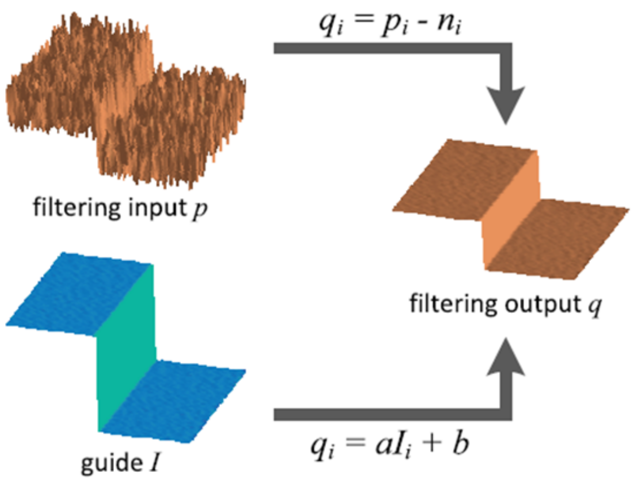

Figure 3: Guided Image Filtering

From the above, we have known that the guided image **I** can be input image **p** itself, and also other different image. If it is same as the input image **p**, the structures are the same, which means an edge in original image **p** is the same in the guided image **I**. On the other hand, if guided image **I** is different from the input image **p**, the structures in the **I** will impact the output image **q**, in effect, imprinting these structures on the original image **P**. This effect is called *structure transference*[6].
Then, we use equation $(1) - (2)$, to get

$$n_i = p_i - aI_i - b \quad (3)$$

And we can extend this equation to define a cost function

$$E(a_k, b_k) = \sum_{i \in \omega_k} ((a_k I_i + b_k - p_i)^2 + \epsilon a_k^2) \quad (4)$$

Equation (4) is *linear ridge regression* model[7][8]. What we want to get is a output image with less noise, which means we want n_i to be minimized. Corresponding to equation (4), we can know what we need to do is minimized the equation (4). So our target change to find a linear coefficients a_k and b_k to let equation (4) to be minimized. Author uses linear regression to get this[8][9].

$$\begin{cases} a_k = \dfrac{\frac{1}{\omega}\sum_{i \in \omega_k} I_i p_i - \mu_k \bar{p}_k}{\sigma_k^2 + \epsilon} & (5) \\ b_k = \bar{p}_k - a_k \mu_k & (6) \end{cases}$$

And the last step, we assign equation(5)(6) back to (2). We can get the filtered image. Here is a sample output and the comparison with *Bilateral Filter*.

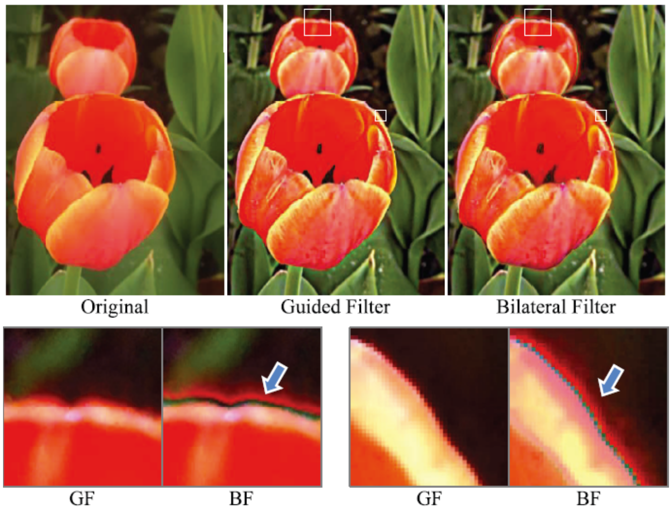

Figure 4: Guided Image Filtering and Bilateral Filter

IV Texture Filter with Interval Gradient

This method creates a new gradient named *Interval Gradient*. Unlike previous filtering algorithms, this method adaptively smooths image gradients to filter out textures from the input images. Using interval gradients, textures can be distinguished from structure edges and smoothly varying shadings[3]. In addition, the proposed method is composed of 1D local filters, similarly to the method in [9].

The method is two-fold. Initially, it is the data preparing. First, we need to get the normal gradient, defined as

$$(\nabla I)_p = I_{p+1} - I_p \quad (1)$$

It measures the difference between two adjacent signal values. In contrast, our interval gradient for pixel p is defined as

$$(\nabla_\Omega I)_p = \mathbf{g}_\sigma^r(I_p) - \mathbf{g}_\sigma^l(I_p) \quad (2)$$

where $\mathbf{g}_\sigma^r and \mathbf{g}_\sigma^l$, respectively, represent left and right clipped 1D Gaussian filter functions defined by

$$\begin{aligned} \mathbf{g}_\sigma^r(I_p) &= \tfrac{1}{k_r}\textstyle\sum_{n\in\Omega(p)} \omega_\sigma(n-p-1)I_n \\ \mathbf{g}_\sigma^l(I_p) &= \tfrac{1}{k_l}\textstyle\sum_{n\in\Omega(p)} \omega_\sigma(p-n)I_n \qquad (3) \end{aligned}$$

ω_σ is the clipped exponential weighting function with a scale parameter σ:

$$\omega_\sigma(x) = \begin{cases} \exp(-\frac{x^2}{2\sigma^2}) & if\ x \geq 0 \\ 0 & otherwise \quad (4) \end{cases}$$

and k_r and k_l are normalizing coefficients defined as

$$\begin{aligned} k_r &= \textstyle\sum_{n\in\Omega(p)} \omega_\sigma(n-p-1) \\ k_l &= \textstyle\sum_{n\in\Omega(p)} \omega_\sigma(p-n) \qquad (5) \end{aligned}$$

After we get the interval gradient, we need to compare its sign with the normal gradient to compute the rescaled gradient

$$(\nabla' I)_p = \begin{cases} (\nabla I)_p \cdot w_p & if\ sign((\nabla I)_p) = sign((\nabla_\Omega I)_p) \\ 0 & otherwise \end{cases} \quad (6)$$

$(\nabla' I)_p$ is the rescaled gradient, and w_p is the rescaling weight:

$$w_p = \min\left(1, \frac{|(\nabla_\Omega I)_p| + \epsilon_s}{|(\nabla I)_p| + \epsilon_s}\right) \quad (7)$$

After that, we need to get the temporary signal R based on the rescaled gradient

$$R_p = \sum_{k=0}^{p-1} I_0 + (\nabla' I)_k, p \in \{0, 1, \ldots, N_p\} \quad (8)$$

Based on what we have talked about in the section III, we can connect section III and IV to see together. Then we can get a result that what R_p doing in the section IV is same as the guided Image in the section III. However, the way they computer and filter is not same because *interval gradient filtering* method is go through 1D, and *guided image filter* is 2D case. Same as the before, we get the *linear ridge regression* model and the linear coefficients a and b[7][8]

$$a_p = \frac{\mathbf{g}_\sigma((RI)_p) - \mathbf{g}_\sigma(R_p)\mathbf{g}_\sigma(I_p)}{\mathbf{g}_\sigma(R_p^2) - \mathbf{g}_\sigma(R_p)^2 + \epsilon}$$

$$b_p = \mathbf{g}_\sigma(I_p) - a_p \cdot \mathbf{g}_\sigma(R_p) \quad (9)$$

And also same as the *guided Image filter*, we can use the linear coefficients to build the new image

$$S_p = \mathbf{g}_\sigma(a_p)R_p + \mathbf{g}_\sigma(b_p)$$

and the $\mathbf{g}$ function is the *Gaussian smoothed method.*

V Comparison

At first, in the figure 4, we can see for the square area, the *Guided Image Filter* is better than *Bilateral Filtering* in detail part[2]. Also, depends on the mathematic, *Guided Image Filter* is linear while *Bilateral Filtering* is non-linear, so we can say that for the most cases *Guided Image Filter* is better than the *Bilateral Filtering.* And I will show another picture to double prove this. (Figure 5)

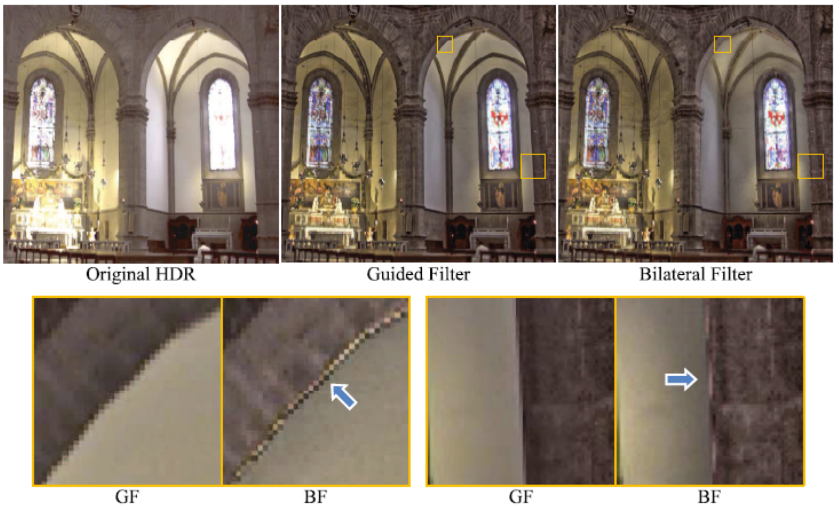

Figure 5: Guided Image Filtering and Bilateral Filter

And for the *Interval Gradient Filter*, because some of the parts are based on the *Guided Image Filter*, the result is not a lot of differences. However, due to this method adaptively smooths image gradients to filter out textures from images instead of from the images directly, it can deal with more complicated input images, which means it will have more opportunities to be used by industry.

VI Conclusion

In this report, we introduce several image texture filter method including *Bilateral Filtering*, *Guided image filtering*, and *Structure-Texture Decomposition of Images with Interval Gradient*. *Bilateral Filtering* is non-linear, the other two are linear. All of three method can get a good result for filtering an image. However, compared with *Bilateral Filtering*, the other two have more advantages, including the filter range of the images and the running time complexity. However, due to *Interval Gradient filtering* smooths image by smoothing image's gradients, it can figure out bigger range of the input and it has more potential in the future's progress.

REFERENCES

[1] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images", Proc. IEEE Int¡¯l Computer Vision Conf., 1998

[2] HE K., SUN J., TANG X.: "Guided image filtering". pami 35, 6 (2010), 1397¨C1409.

[3] L. Hyunjoon, J. Junho, K. Junho, L. Seungyong, "Structure-Texture Decomposition of Images with Interval Gradient", Computer Graphics forum, V.00(2016), N.0 pp.1-13

[4] R.C. Gonzalez and R.E. Woods, "Digital Image Processing", second ed. Prentice Hall, 2002.

[5] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama, "Digital Photography with Flash and No-Flash Image Pairs," Proc. ACM Siggraph, 2004.

[6] E.S.L. Gastal and M.M. Oliveira, "Domain Transform for Edge-Aware Image and Video Processing", ACM Trans. Graphics, vol. 30, no. 4, pp. 69:1-69:12, 2011.

[7] N. Draper and H. Smith, Applied Regression Analysis, second ed. John Wiley, 1981.

[8] T. Hastie, R. Tibshirani, and J.H. Friedman, The Elements of Statistical Learning. Springer, 2003.

[9] GASTAL E. S. L., OLIVEIRA M. M.: Domain transform for edge-aware image and video processing. ACM Transactions on Graphics 30 (2011), 69:1¨C69:12.