Noise Reduction Using Enhanced Bilateral Filter

Yen-Lan Huang, Chiou-Shann Fuh

Department of Computer Science and Information Engineering, National Taiwan University,
Taipei, Taiwan, 10617, R.O.C
r93092@csie.ntu.edu.tw

Abstract. Noise reduction is an important block in the image pipeline. Noticing noise in an image is unpleasant. A good noise reduction method can reduce the noise level and preserve the detail of the image. In this paper, we introduce some basic noise types and traditional noise reduction methods. Then we create photometric functions and geometric functions based on the concept of the bilateral filter. We use experiments to show our proposed methods are more robust to salt-and-pepper noise. Besides, we show that our methods take less time compared with Gaussian bilateral filter.

1. Introduction

1.1 Motivation

In order to get a clean and sharp image, noise reduction is the main issue in image pipeline. Many filters were used to reduce noises but also blur the whole image because image details and noises are difficult to distinguish by computer. Filtering is the most popular method to reduce noise. In the spatial domain, filtering depends on location and its neighbors. In the frequency domain, filtering multiplies the whole image and the mask. Some filters operate in spatial domain, some filters are mathematically derived from frequency domain to spatial domain, other filters are designed for special noise, combination of two or more filters, or derivation from other filters.

C. Tomasi and R. Manduchi introduce a noise reduction filtering method called bilateral filtering [5]. This paper modifies Tomasi's method to make it suitable for more noises.

1.2 Introduction to Noise Types

Noises are inevitable in devices which contain analog to digital converter. Different noises will occur in each step from original source to digital output.

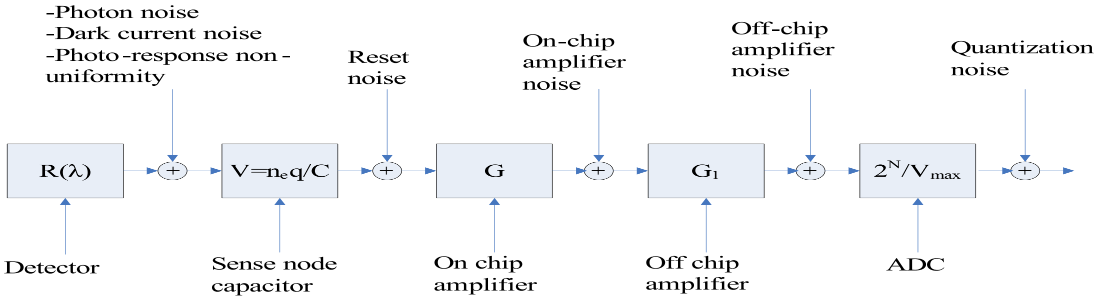


Fig. 1. Noise transfer diagram [4].

In general, noise types can be divided to two general parts. One is fixed pattern noise, the other is temporal noise. Fixed pattern noise remains the same in different shots, and temporal noise differs from different shots.

2. Analysis of Noise

2.1 Noise Level Measurement

In order to measure the level of the noise, signal to noise ratio (SNR) is a good benchmark to calculate the noise level if we have the original image [3]. In our experiment, we add mathematically generated noises on the known image to simulate the noise behavior. That is, original image is known, and we can easily calculate the SNR value before and after the noise reduction method. SNR is defined as follows:

$$\begin{aligned} SNR &= 10 \times \log_{10}\left(\frac{VS}{VN}\right) \\ VS &= \sum_N (I(i,j) - \mu_s)^2 \\ VN &= \sum_N (I'(i,j) - I(i,j) - \mu_n)^2 \\ \mu_s &= \sum_N I(i,j) \\ \mu_n &= \sum_N (I'(i,j) - I(i,j)) \end{aligned} \tag{1}$$

where *VS* is the gray level image variance; *VN* is the noise variance; *N* is the total pixel number of the image; *I*(*i*, *j*) is the original image pixel value at (*i*, *j*); and *I'*(*i*, *j*) is the noise image pixel value at (*i*, *j*).

2.2 Noise Models

In general, noise behavior can be divided into two models: Gaussian model and salt and pepper noise (impulse noise). For example, photon noise, dark current noise, photo response non-uniformity, readout noise, and ADC noise can be modeled as Gaussian noise. The dead pixel, and the stuck pixel can be explained as salt and pepper noise. We use mathematical models to simulate Gaussian noise and the salt and pepper noise.

For Gaussian noise, we use the following equation [3]:

$$I'(i,j) = I(i,j) + amplitude \times N(0,1) \tag{2}$$

where *I'(i, j)* is the output image value; *I(i, j)* is the original image value at the point (*i*, *j*). Variable *amplitude* determines the noise amplitude. Random function *N*(0, 1) has normal distribution with mean=0 and standard deviation=1 to simulate the noise.

For salt and pepper noise, we use another equation to simulate [3]:

$$\begin{aligned} I'(i,j) &= 0 && \text{if } uniform(0,1) < p \\ I'(i,j) &= 255 && \text{if } uniform(0,1) > 1 - p \\ I'(i,j) &= I(i,j) && \text{otherwise} \end{aligned} \tag{3}$$

where *I'(i, j)* and *I(i, j)* are the same definition as above; random function *uniform*(0, 1) has uniform distribution with random numbers in the interval (0, 1); variable *p* determines the level of the salt and pepper noise.

3. Original Bilateral Filter

3.1 Introduction to Bilateral Filter

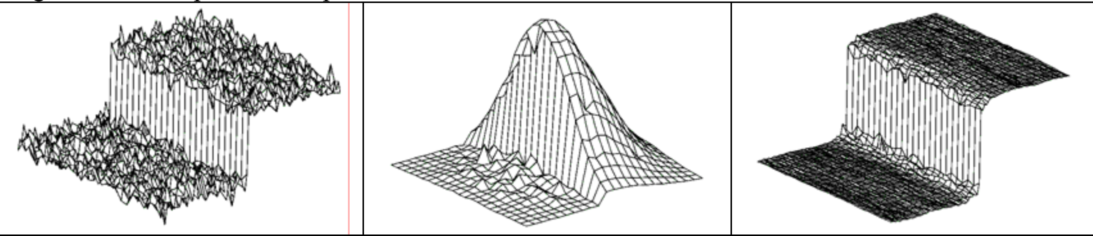
Our method is based on Tomasi's paper [5]. In Tomasi's paper, he introduces a method combining geometric closeness and photometric similarity at the same time. In his method, two elements are combined by a mathematical equation as follows:

$$h(x) = k^{-1}(x) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\xi) c(\xi, x) s(f(\xi), f(x)) d\xi \tag{4}$$

with the normalization

$$k(x) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, x) s(f(\xi), f(x)) d\xi \qquad \textbf{(5)}$$

where $\boldsymbol{h}$ is the output and $\boldsymbol{f}$ is the input, $c(\xi, \boldsymbol{x})$ measures the geometric closeness between the neighborhood center $\boldsymbol{x}$ and a nearby point ξ, $s(\boldsymbol{f}(\xi), \boldsymbol{f}(\boldsymbol{x}))$ is the photometric similarity between the pixel at the neighborhood center $\boldsymbol{x}$ and that of a nearby point ξ. The normalization term $k(\boldsymbol{x})$ ensures that the weights for all the pixels add up to one.



(a) A 100-gray-level step perturbed by Gaussian noise with σ = 10 gray levels	(b) Combined similarity weights for a 23*23 neighborhood	(c) The step in (a) after bilateral filtering.

Fig. 2. The bilateral filter [5].

The bilateral filter is a basic concept for noise cleaning. This concept needs two well-designed functions mentioned in the previous paragraph. Images will become unpredictable if the functions were not designed properly.

3.2 The Gaussian Case Example

A simple case of bilateral filtering is shift-invariant Gaussian filtering. In this case, geometry closeness function and the photometry similarity function are Gaussian functions.

The geometric function uses the Gaussian function as follows:

$$c(\xi, x) = e^{-\frac{1}{2}\left(\frac{d(\xi, x)}{\sigma_d}\right)^2} \qquad \textbf{(6)}$$

where

$$d(\xi, x) = d(\xi - x) = \| \xi - x \| \qquad (7)$$

the $c(\xi, \boldsymbol{x})$ and the σ_d are the same definitions in Section 3.1; and the $d(\xi, \boldsymbol{x})$ is the Euclidean distance between ξ and $\boldsymbol{x}$.

The photometric function can also use a Gaussian equation. The Gaussian photometric function is as follows:

$$s(\xi, x) = e^{-\frac{1}{2}\left(\frac{\delta(f(\xi), f(x))}{\sigma_r}\right)^2} \qquad \textbf{(8)}$$

where

$$\delta(\phi, f) = \delta(\phi - f) = \| \phi - f \| \qquad \textbf{(9)}$$

the $s(\xi, \boldsymbol{x})$ and the σ_r are the same definitions in Section 3.1; and the $\delta(\phi, f)$ is a suitable measure of distance between the two intensity values ϕ and $\boldsymbol{f}$.

3.3 Disadvantages of the Tomasi's Method

Tomasi's method can get a clear image from a noisy image. But impulse noise will still exist after Tomasi's method applied to the image. Figure 3. shows the situation where bilateral filter will give the unpleasant image. This error happens because the photometric function can not handle the impulse value of the noise. The photometry made the weighting of the neighborhood pixels very small but the pixels where the impulse noise lies have large weights. In order to solve this problem, two ways can be selected. The first is to design a robust photometric function where the impulse is well-handled; the second is to combine the ideas of the previous works in noise reduction. In our experiment, both ways were examined to find the most robust photometric filter. The first step of our method is to provide an easily designed geometric function, and then invent the well-designed photometric function which can handle the impulse noise.

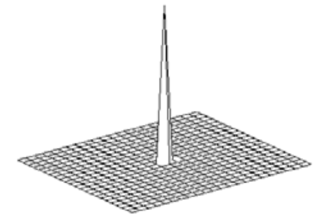
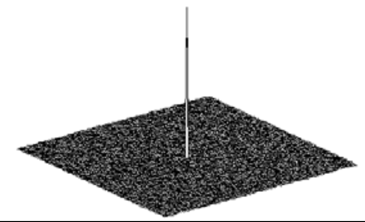
		
(a) Impulse noise in a noisy image	(b) Bilateral filter mask at the impulse point	(c) Impulse noise after the bilateral filtering.

Fig. 3. Disadvantage of the bilateral filter where salt-and-pepper noise remains after bilateral filtering in (c).

The performance of bilateral filter is determined by the two functions: geometry function and the photometry function. If those two functions are well-designed, the result of the bilateral filtering is good. On the other hand, well-designed functions are usually hard to implement or may cost too much time. In our proposed method, we invent an easily-implemented photometry function which can be tuned for different purposes.

4. Our Proposed Method

4.1 Geometric Function

Once we index the mask, we can use a simple equation to generate weights of geometric function. The definition of the Gaussian function uses exponential calculation to get the weighting, which is a time-consuming calculation. In our method, we replace the exponential definition of Gaussian function with the following equations:

$$c(\xi, x) = \frac{1}{d(\xi, x)^2 + 1} \qquad \textbf{(10)}$$

where the $d(\xi, \boldsymbol{x})$ is the Euclidean distance of ξ and $\boldsymbol{x}$.

Using this definition for the geometric function prevents the exponential calculation in each mask element and preserves the characteristic of the Gaussian filter at the same time. To prevent the divide-by-zero error, we add 1 to the denominator. The central weighting can be modified to zero to prevent the impulse noise condition.

4.2 Photometric Function

The photometry similarity function is an important factor to decide the performance of the bilateral filter. Because of the disadvantage of the bilateral filter, the impulse noise effect must be considered when designing a photometric function. In our method, we create three different photometric functions. All of the three functions need less computation compared with the Gaussian function. The first two functions keep the linear property and are more robust to the impulse noise compared with the Gaussian function, and the third function combines does not keep the linear property. In the third function, the impulse noise error is perfectly prevented.

4.2.1 The Difference Summation Function

The first photometric function uses the following equation:

$$s(\xi, x) = \frac{\sum_{\xi} (d(\xi, x) + 1)}{d(\xi, x) + 1} \tag{11}$$

where $d(\xi, \boldsymbol{x})$ is the photometry difference between ξ and $\boldsymbol{x}$. Similar to the denominator in the geometric function, the denominator in this photometric function is also added by 1 to prevent the divide-by-zero error.

Every pixel in the image will go through the whole procedure. First, calculate the geometric weighting of the geometric mask. Second, calculate the photometric weighting using the photometric function. Finally, apply the mask in the certain pixel.

4.2.2 The Single Difference Function

The second photometric function uses σ_r to determine the weighting of the mask. The main concept of the second function is to calculate the weighting value according to the luminance difference between the central pixel and the current pixel. Weighting value can be calculated without reference to the summation of difference between central pixel and every pixel in the mask. It only refers to the current pixel and the central pixel. The concept illustration is as follows:

Central pixel value

Pixel value difference	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5
Denominator	3	2	2	2	1	1	1	2	2	2	3
Photometry weighting	$\frac{1}{3}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{1}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{3}$

σ_r σ_r σ_r

Fig. 4. Illustration of the concept of the weighting table.

where σ_r determines the combinational range of the pixel luminance. In Figure 4, we take $\sigma_r = 3$ for example.

The photometry weighting values in Figure 4 is our goal. After derivation, we can use mathematical equation to calculate the weighting value in all σ_r values. The mathematical equation is shown as follows:

$$s(\xi, x) = \frac{1}{\left\lfloor \frac{d(\xi, x)}{\sigma_r} + 1.5 \right\rfloor} \tag{12}$$

where σ_r is the combinational range; $d(\xi, \boldsymbol{x})$ is the Euclidean distance of ξ and $\boldsymbol{x}$; and $s(\xi, \boldsymbol{x})$ is the photometry weighting value.Using the single difference function as the photometric function, the bilateral filter becomes a one-pass procedure.

4.3 The Hybrid Function

The third photometric function combines the ideas of previous works in noise reduction. It is also our proposed method. The method we choose is the alpha-trimmed filter, because α value can decide the performance of the alpha-trimmed filter. In the n x n mask, if we set $\alpha = \frac{n^2 - 1}{2}$, the performance of the alpha-trimmed filter is the same as median filter. If we set $\alpha = 1$, the alpha-trimmed filter will discard the maximum and minimum value in the mask.

We can combine the concept of the alpha-trimmed filter into the single difference function to generate the hybrid photometric function. In the hybrid function, the largest and the smallest α pixels of the luminance are remembered. After the whole photometric mask is generated, change the photometric mask weighting to zero if the corresponding pixel is the largest or the smallest α values in the mask.

To determine the alpha value of the hybrid method, we use at least one-third of pixels covered by the mask. That is, the alpha value will not be larger than one-third of the pixel numbers. The bilateral filter using the hybrid function as the photometric function is a two-pass procedure.

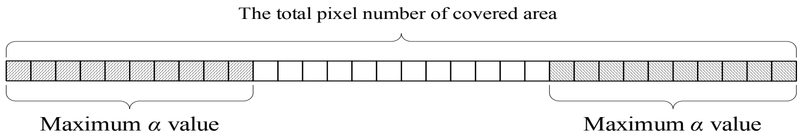


Fig. 5. Determine the maximum α value.

5. Experiments and Results

5.1 Test Conditions

In our experiments, we use two different amplitude values for Gaussian noise, two different p values for salt-and-pepper noise, and combination of Gaussian noise and salt-and-pepper noise. All eight noisy images are gray level with height and width 512 pixels, BMP (Bitmap) format. Our testing environment is a desktop PC (Personal Computer) with an AMD Sempron™ Processor 2800+, 1.61GHz, and 1G RAM (Random Access Memory).

5.2 Comparison of Our Method and Bilateral Filter

In order to show the difference between our method and the Gaussian bilateral filter, we put every noise type into a graph to show the SNR and tickcount value.

Overall, our proposed method gives clearer image and needs less computing time.

The comparison images of our proposed method and the Gaussian case bilateral filter center weighted normally are shown as follows:

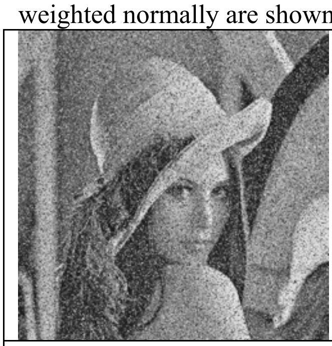
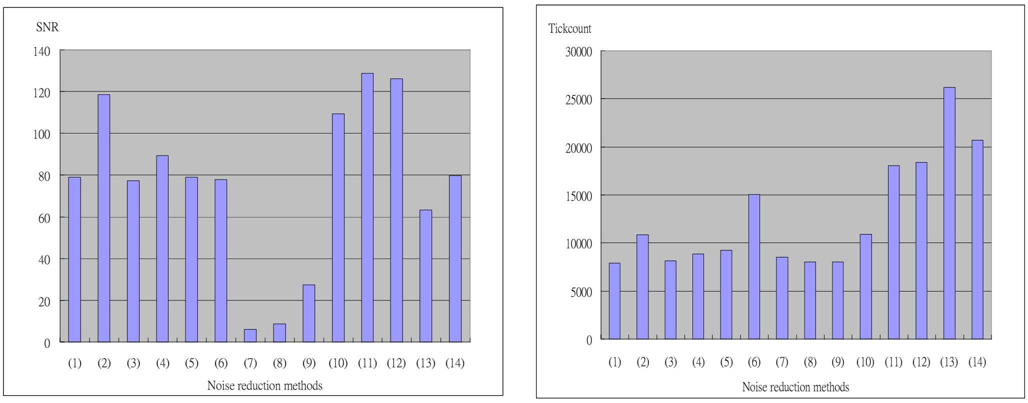
		
(a) Original image with salt-and-pepper noise p = 0.1. SNR=-2.1	(b) Bilateral filter. SNR=3.7	(c) Our proposed method. SNR=16.4

Fig. 6. Sample images for bilateral filter and our proposed method.

5.3 Summary

To examine the total performance of our proposed method of all noisy images, we summarize the SNR values and tickcount values for each noise types. The higher SNR value means the better overall results. The lower tickcount value means the faster calculation.

(1)	Box Filter	(2)	Median Filter
(3)	Outlier	(4)	Contrast Dependent
(5)	Smooth Replacement	(6)	Nearest Neighbor
(7)	Max Filter	(8)	Min Filter
(9)	Midpoint Filter	(10)	Alpha-trimmed
(11)	Hybrid Bilateral Center Weighted	(12)	Hybrid Bilateral Center Weighted Zero
(13)	Gaussian Case Bilateral Center Weighted	(14)	Gaussian Case Bilateral Center Weighted Zero



(a) SNR summation of all noise types. (b) Tickcount summation for all noise types.

Fig. 7. The summation of SNR value and tickcount value.

Our proposed methods (11) and (12) have the highest SNR values although with more computation time.

6. Conclusion

The hybrid bilateral filter can reduce the salt-and-pepper noise and the Gaussian noise and preserve the detail of the image. Although applying median filter and the box filter at the same time can also reduce the salt-and-pepper noise and Gaussian noise. Applying the mask twice will blur the image twice and lose the image detail. As to calculation time, applying median filter and box filter needs twice the computing time. Although our proposed method takes more time than box filter and median filter, the combination of box filter and median filter takes more time than our proposed method.

Median filter is good to salt-and-pepper. In our experiment, our proposed method is more robust to salt-and-pepper noise than median filter. That is, median filter applied to one image with heavy salt-and-pepper noise may leave some salt or pepper pixels. Using our method to the same image will leave less salt-and-pepper noise or even eliminate the salt-and-pepper noise in the whole image.

Acknowledgement

This research was supported by the National Science Council of Taiwan, R.O.C., under Grants NSC 94-2213-E-002-032 and NSC 93-2213-E-002-073, by the EeRise Corporation, EeVision Corporation, Machvision, Tekom Technologies, IAC, ATM Electronic, Primax Electronics, Scance, Lite-on and Liteonit.

Reference

[1] M. C. Gino, “Noise, Noise, Noise,” http://www.astrophys-assist.com/educate/noise/noise.htm, 2002.
[2] R. C. Gonzalez and R. E. Woods, Digital Image Processing, 2nd Ed., Prentice-Hall, Upper Saddle River, NJ, 2002.
[3] R. M. Haralick and L. G. Shapiro, Computer and Robot Vision, Vol. I, Addison Wesley, Reading, MA, 1992.
[4] T. Z. Regier, “Development of a Gamma-Ray Beam Profile Monitor for the High-Intensity Gamma-Ray Source,” Master Thesis, Department of Physics and Engineering Physics, University of Saskatchewan, Saskatoon, 2003.
[5] C. Tomasi, R. Manduchi, "Bilateral Filtering for Gray and Color Images," Proceedings of International Conference on Computer Vision, Bombay, pp. 839-846, 1998.
[6] J. Wilkinson, “Production and Transmission of Light,” http://www.life.umd.edu/faculty/wilkinson/BSCI338/L8ltproduct/, 2002.