

NEW HIERARCHICAL NOISE REDUCTION

¹ Hou-Yo Shen (沈顥祐), ¹ Chou-Shann Fuh (傅楸善)

¹ Graduate Institute of Computer Science and Information Engineering, National Taiwan University
E-mail: kalababygi@gmail.com

ABSTRACT

In this thesis, we propose a new hierarchical noise reduction. Y. C. Wang's method [11] is very powerful for noise reduction, but very slow in computation and may be difficult to use. Therefore, we modify some mechanisms to keep more detail and speed up.

1. INTRODUCTION

Noise reduction is important in image processing. For example, we may have pictures with much noise while taking the pictures in dark. If we can decrease the noise, we can get higher quality images.

2. NOISE TYPES

Fixed pattern noise includes "hot pixels" and "cold pixels." They are pixels with fixed value. "Hot" means the pixel value is always high, and oppositely "cold" means the pixel value is always low. Defect sensor causes these types of bad pixels. Another reason of fixed pattern noise is long time exposure, and especially with high temperatures. In short, fixed pattern noise always appears in the same position.

Random noise is a noise which intensity and color fluctuations above and below the actual image intensity. They are always random at any exposure length and most influenced by ISO (International Standards Organization) speed.

Banding noise is characterized by the straight band in frequency on the image and highly camera-dependent. Unstable voltage power causes the noise. It is most visible at high ISO speed and in dark image. Brightening the image or white balance can increase the problem.

3. 1.2.1 DIGITAL IMAGE AND COLOR SPACE TRANSFORMATION

A digital image is a representation of a two-dimensional image using ones and zeros (binary). Depending on whether or not the image resolution is fixed, it may be of vector or raster type. Without

qualifications, the term "digital image" usually refers to raster images. [12]

A color space is a method by which we can specify, create and visualize color. As humans, we may define a color by its attributes of brightness, hue and colorfulness. A computer may describe a color using the amounts of red, green, and blue to match a color. [9]

Digital camera sensors often record images with Bayer filter, so they work on the RGB model. In a Bayer filter arrangement, the ratio of red, green, and blue is 1:2:1. Green is more than other colors because human eyes are sensitive to green. The sensor has a grid of red, green, and blue detectors arranged and the first row is RGRGRGRG, the next is GBGBGBGB, then the sequence is repeated in subsequent rows. [14]

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B \quad (1)$$

$$Cb = 0.564 \times (B - Y) = -0.169 \times R - 0.331 \times G + 0.500 \times B$$

$$Cr = 0.713 \times (R - Y) = 0.500 \times R - 0.419 \times G - 0.081 \times B$$

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 noisy image pixel value at (i, j) .

4. NOISE LEVEL MEASUREMENT

SNR (Signal to Noise Ratio) is a popular criterion formula for noise reduction software. The way to test the effect of the noise reduction algorithms is to add noise model mathematically on a digital image and then calculate the SNR value before and after the noise reduction algorithm. SNR is defined as follows [2]:

$$SNR = 10 \times \log_{10} \left(\frac{VS}{VN} \right) \quad (2)$$

$$VS = \sum_N (I(i, j) - \mu_s)^2$$

$$VN = \sum_N (I'(i, j) - I(i, j) - \mu_n)^2$$

$$\mu_s = \sum_N I(i, j)$$

$$\mu_n = \sum_N (I'(i, j) - I(i, j))$$

5. MAJOR HEADINGS

Gaussian noise is another noise model and can simulate the random noise caused by temperatures, defined as below:

$$I'(i, j) = I(i, j) + \text{amplitude} \times N(0, 1) \quad (3)$$

where $I'(i, j)$ and $I(i, j)$ are the same definition as above; and 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.

6. ORIGINAL HIERARCHICAL METHOD

6.1. Introduction to Hierarchical Method

Y. C. Wang's method [11] is based on the patent published by Imagenomic Limited Liability Company, the producer of Noiseware [8]. For this reason, we think that Noiseware is implemented with this patent.

6.2. Detailed Description of Hierarchical Method [8]

In this section, we describe the steps of Hierarchical Method.

6.2.1. Decomposing the Image into Multi-Scaled Frequency Images

Hierarchical method decomposes image by downscaling image. The $m \times n$ pixel original image is called layer1 image, and we create layer2 image by downscaling the original image to quarter (half the width and length of image, $m/2 \times n/2$ pixels). Repeating the steps, thus we will have layer3 image (the quarter of layer2, $m/4 \times n/4$ pixels) and layer4 image ($m/8 \times n/8$ pixels). In this step, it also decomposes original image to three different channels, Y, Cb, and Cr for processing the channels individually in the following steps.



Fig. 1: Illustrations of multi-scaled frequency images.

6.2.2. Determining Edge Pixels

Keeping detail is based on finding edge. For determining edge pixels, it defines a mask such as Figure 3.2 and a threshold T ($T < 300$). In Figure 3.2, we are processing the middle pixel now and comparing with others black pixels.

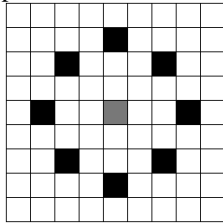


Fig. 1: Illustration of determining edge pixels

If the difference between the middle pixel and one of the black pixels is over $T/2$, we label it as E1. Label it as E2 when it is less than $T/2$ and more than $T/4$. At last, we label all the pixels which are non-labeled as N. Labels E1, E2, E3, E4 and N express the intensity of edge; E1 is the strongest; E4 is the weakest; and N is non-edge. We will use the other labels later.

6.2.3. Fixing Broken Edge Pixels and Eliminating Singular Edge Pixels

It is impossible that an edge pixel with no neighbor which is also an edge pixel. Therefore, use a 3×3 mask to correct mislabeled pixels. In the mask, the middle one is the pixel we check now such as the last step. The middle pixel will be relabeled as N when it is an edge pixel which was labeled as E1 or E2 without neighboring any edge pixels. On the other hand, the middle pixel will be relabeled as E2 when it is a non-edge pixel which was labeled as N with neighboring more than 3 edge pixels.

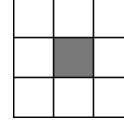


Fig. 1: Mask for correcting edge labels.

6.2.4. Eliminating Mislabeled Edge Pixel Clusters

An image with much noise may have some noise cluster, which is some noise pixel gathering together. A cluster totally contained in a square is thought a gathering noise here. For avoiding the situation, the method defined some squares from 5×5 to 15×15 pixels. Then label all pixels in clusters N if the clusters are all contained in testing square.

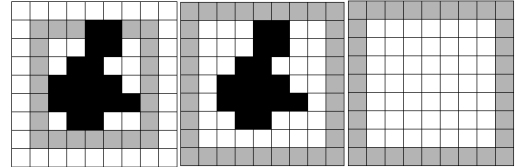


Fig. 1: Illustration of eliminating mislabeled edge pixel clusters.

6.2.5. Edge Cushioning

We believe that human eyes always like to see the smooth pictures. In this case, no one likes the picture which is with sharp edges and other blur parts. The method solves the problem by adding lower intensity label around edge pixels. For example, add E2 label to pixels around E1 pixels, then add E3 label to pixels around E2 pixels. In Cb and Cr channels, add E4 label to pixels around E3 pixels.

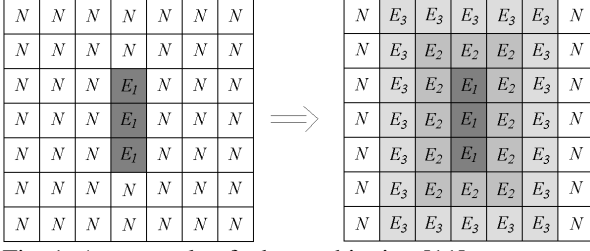


Fig. 1: An example of edge cushioning [11].

6.2.6. Determining Initial Edge Pixel Direction

Edge is a series of pixels with similar colors on a line or a curve. Here we only consider line edges as follows.

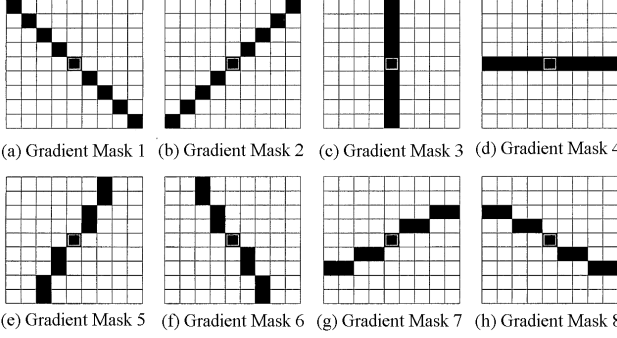


Fig. 1: Luminance gradient masks [11].

Middle one is the current pixel. After calculating the summation of the difference between the middle pixel and other black pixels in each mask, the direction of the edge of the middle pixel has the minimum value among other directions. The calculating formula is as follows:

$$G = \text{Min} \sum_{k=1 \dots 8} (P_0 - P_{ik})^2 \quad (3)$$

where G is the gradient; P_0 is the value of current pixel; and P_{ik} is the value of the i -th pixel of P_0 's reference pixel in the k mask.

6.2.7. Correcting the Edge Pixel Gradient

Use masks in Figure 3.3 again and test the directions of the nine grids. Re-label the direction gradient of middle pixel if the middle one is an edge and differs from general direction. For example, if 4 neighbors of middle one are assigned gradient direction 2; while 2 neighbors are assigned gradient direction 3, and 2 neighbors are assigned gradient direction 1, the middle one will be assigned gradient direction 2.

6.2.8. Smoothing the Luminance and Chroma Values of the Edge Pixels

Smooth edge pixels with their own direction gradient mask:

$$D_k = |Y_o - Y_k| \quad (4)$$

where Y_o is the luminance value of the middle pixel in Figure 3.5; and Y_k is the luminance value of k -pixel in the mask of Y_o . In addition, another threshold value denoted by T_{lum} is assigned. If D_k is lower than T_{lum} then the luminance value is multiplied by the luminance value of the mask pixel and the difference of T_{lum} and D_k to calculate the weighted value of the mask pixel according to:

$$W_k = Y_k \times (T_{lum} - D_k) \quad (5)$$

where W_k is the weighted value of the mask pixel; Y_k is the luminance value of k -pixel in the mask; T_{lum} is the threshold value of the calculation; and D_k is defined in Equation (3.3).

$$L'_o = \frac{\sum_k W_k}{\sum_k (T_{lum} - D_k)} \quad (6)$$

When we have the weighted value of all mask pixels we calculate the "smoothened" value of the current edge pixel, by summing together all of the weighted values calculated in Equation (3.3) according to:

where L'_o is the smoothed luminance value; and W_k , Y_k , and D_k are defined above.

Moreover we process chroma pixels in similar steps.

6.2.9. Combining Different Frequency Images to Produce a Processed Image

We need to repeat this step 3 times; at first, to combine layer3 and layer4, then, to combine layer2 and layer3, and finally, to combine layer1 and layer2. For example, layer3 is high frequency layer and layer4 is low one when we combine layer3 and layer4 images. The formulas are used to combine as follows:

Luminance:

$$E1: H = 7/8H + 1/8L;$$

$$E2: H = 1/2H + 1/2L;$$

$$E3: H = 1/4H + 3/4L;$$

$$N: H = L;$$

Chroma:;

$$E1: H = H;$$

$$E2: H = 3/4H + 1/4L;$$

$$E3: H = 1/2H + 1/2L;$$

$$E4: H = 1/8H + 7/8L;$$

$$N: H = L;$$

(7)

where H is the pixel value in higher frequency image; L is the pixel value in upscaled image; H and L are in the same position; $E1$, $E2$, $E3$, and N are the same definition as "Determining Edge Pixels."

7. Improvements of Y. C. Wang's Method

7.1. Multi-threshold

In Figure 1.3, there is a higher noise in the lower luminance part in an image. For the reason, Y. C. Wang

added more thresholds for determining edge pixels as below:

Intensity	0~31	32~63	64~95	96~127	128~159	160~191	192~223	224~255
Layer1	T ₁₁	T ₁₂	T ₁₃	T ₁₄	T ₁₅	T ₁₆	T ₁₇	T ₁₈
Layer2	T ₂₁	T ₂₂	T ₂₃	T ₂₄	T ₂₅	T ₂₆	T ₂₇	T ₂₈
Layer3	T ₃₁	T ₃₂	T ₃₃	T ₃₄	T ₃₅	T ₃₆	T ₃₇	T ₃₈
Layer4	T ₄₁	T ₄₂	T ₄₃	T ₄₄	T ₄₅	T ₄₆	T ₄₇	T ₄₈

7.2. Different Mask Sizes

The mask for determining edge pixels is unsuitable in different image sizes, especially in the small image. Y. C. Wang supported extra masks for different image sizes as follows.

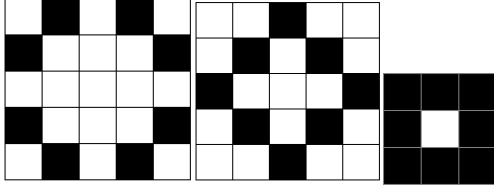


Fig. 1: Different masks for judging edge pixel [11].

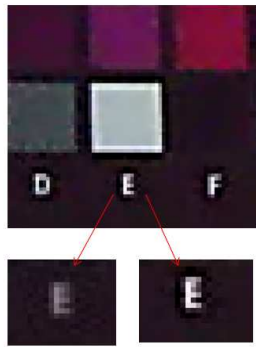
7.3. Pixel distance

Square operation takes higher computing power in determining edge pixels direction. To speed up, absolute operation replaces square operation.

$$G = \text{Min}_{k=1...8} \sum_{i=1...8} |P_0 - P_{ik}| \quad (6)$$

8. OUR PROPOSED METHOD

Y. C. Wang's method and Noiseware support good quality but are weak in detail. Y. C. Wang's method is slow. We propose the modification for keeping more details and speeding up in this chapter.



Noiseware Our Method

Fig. 1: Compare the detail between Noiseware and our method.

8.1. Disable Clusters

Eliminating cluster is harmful to keep image detail. In Figure 4.1, the size of the letter 'E' is less than 15×15 pixels. After eliminating cluster, the labels of the pixels

compose letter 'E' are labeled as N. Therefore 'E' is getting blurred. We propose to disable eliminating cluster function to keep details. In addition, we can reduce reading pixels 225×m×n times where m and n are the width and length of an image.

8.2. Keep Ambiguous Pixels

The gray pixel in Figure 4.2 will be determined to belong to a horizontal line, but it is better to belong to a vertical line. Therefore we will keep the gray pixel if it is at the intersection of white and black edges.

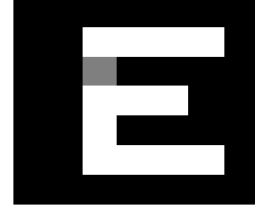
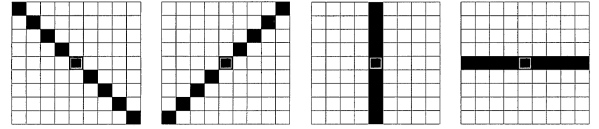


Fig. 1: Illustration of an ambiguous pixel.

8.3. Reduced Edge Pixel Directions

Determining edge pixel direction takes much more time than other steps, so we reduce direction test to 4 directions.



(a) Gradient Mask 1 (b) Gradient Mask 2 (c) Gradient Mask 3 (d) Gradient Mask 4

Fig. 1: The 4 directions of reduced edge directions.

While we take a picture of a ball, it will become an octagon if we take a picture with only 4 direction masks. For this reason, we need a compensation function as below:

if $D_i < \text{Threshold}$:

$$L'_o = \frac{\sum_{kp} W_{kp}}{\sum_{kp} (T_{lum} - D_{kp})}$$

Else:

$$L'_o = \frac{E_p}{E_p + E_s} \times \frac{\sum_{kp} W_{kp}}{\sum_{kp} (T_{lum} - D_{kp})} + \frac{E_s}{E_p + E_s} \times \frac{\sum_{ks} W_{ks}}{\sum_{ks} (T_{lum} - D_{ks})}$$

where W_{kp} is the weighted value of the mask pixel of prime direction; W_{ks} is the weighted value of the mask pixel of secondary direction; T_{lum} is threshold parameter; E_p is the entropy of the primary direction; and E_s is the entropy of the secondary direction.

9. EXPERIMENTS AND RESULTS

9.1. Experimental Environment

CPU: AMD Turion(TM) X2 Ultra Dual- Core Mobile
ZM-82 2.20 GHz
Memory: 4 GB
OS: Windows Vista™
Programming Language: Dev-C++ Version 4.9.9.2

9.2. Images without SNR

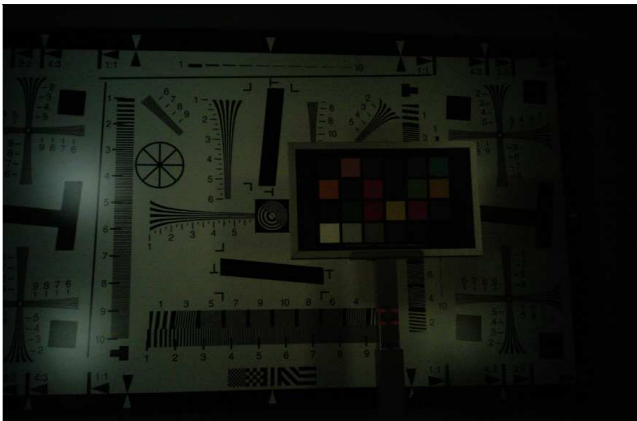


ORIGINAL IMAGE

Luminance threshold	55,55,55,15,15,15,55,55,55,55,55,15,15,15,55,55,35,35,35,10,10,10,35,35,35,35,35,10,10,10,35,35
Chroma threshold	75,25,25,25
Luminance smoothing	15, 15, 15, 5
Chroma threshold	15

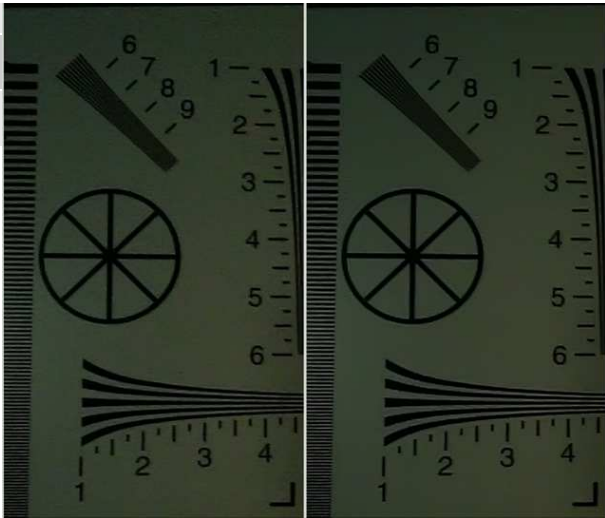


NOISEWARE: 0 VOTES OUR METHOD :21 VOTES



ORIGINAL IMAGE

Luminance threshold	5,55,55,55,55,55,55,55,55,55,55,55,55,55,55,55,25,35,35,35,35,35,35,35,15,15,15,15,15,15,15,15
Chroma threshold	75,25,25,25
Luminance smoothing	5, 5, 5, 5
Chroma threshold	15



NOISEWARE: 0 VOTES OUR METHOD :21 VOTES



ORIGINAL IMAGE	
Luminance threshold	25,20,15,15,5,5,10,10,20,16,12,12,4,4,8,8,15,12,9,9,3,3,6,6,10,8,6,6,2,2,6,6
Chroma threshold	10,6,2,6
Luminance smoothing	15, 15, 15, 5
Chroma threshold	30



NOISEWARE: 3 VOTES OUR METHOD :18 VOTES

9.2. Images with SNR



ORIGINAL IMAGE	
Luminance threshold	130,130,130,130,130,130,130,130,50,50,50,50,50,50,35,35,35,35,35,35,35,25,

	25,25,25,25,25,25,25,15
Chroma threshold	120,50,20,10
Luminance smoothing	55, 30, 5, 5
Chroma threshold	50



NOISEWARE: 0 VOTES OUR METHOD :21 VOTES
SNR: 16.898 SNR: 18.388



ORIGINAL IMAGE

Luminance threshold	140,140,140,140,140,140,140,140,70,70,70,70,70,70,70,70,35,35,35,35,35,35,35,35,25,25,25,25,25,25,25,15
Chroma threshold	60,70,40,20
Luminance smoothing	140, 80, 35, 15
Chroma threshold	40











NOISEWARE: 4 VOTES OUR METHOD :17 VOTES
SNR: 15.669 SNR: 16.124

	Noiseware	Our Method	Vote of Noise-ware	Vote of Our Method
01			0/21	21/21
02			0/21	21/21
03			3/21	18/21
04			6/21	15/21
05			5/21	16/21

06			4/21	17/21
07			0/21	21/21
08			10/21	11/21
09			3/21	18/21
10			6/21	15/21

	Noiseware	Our Method	Vote of Noise-ware (SNR)	Vote of Our Method (SNR)
01			0/21 16.398	21/21 18.388
02			7/21 16.998	14/21 18.121
03			2/21 11.853	19/21 12.875
04			6/21 16.872	15/21 17.517
05			4/21 15.669	17/21 16.124
06			2/21 15.067	19/21 16.074

07			8/21 15.231	13/21 15.727
08			0/2114. 397	21/211 5.546
09			10/21 23.179	11/21 24.188
10			1/21 15.503	20/21 15.839

10. CONCLUSION AND FUTURE WORK

10.1. Conclusion

Our method supports high quality images but the parameters are manually given. That is hard to use for general users.

Image quality is difficult to define. For example, a picture gets many votes, but its SNR may be low. For example the pixels all shifting a pixel to the left reduces the SNR but has the same image quality. No matter what, human eyes like to see the smooth edge even if the SNR of the image is low.

10.2. Future Work

1. In the future, we wish the parameters can be created automatically. It is very useful to general user to use our program.
2. SNR is not a good criterion to de-noise software, therefore maybe we can develop a new function to replace SNR.
3. In 3 million pixel picture, our method takes 15 seconds, and Noiseaware takes 4 seconds. We can try to find some other algorithms to reduce more computing time.

REFERENCES

- [1] A. Buades, "Image and Movie Denoising by Non-Local Means," Ph. D. Thesis of Doctor in Mathematics Universitat de les Illes Balears, 2005.
- [2] R. M. Haralick and L. G. Shapiro, Computer and Robot Vision, Vol. I, Addison Wesley, Reading, MA, 1992.
- [3] Imagenomic, "Noiseaware: The Better Way to Remove Noise," <http://www.imagenomic.com/nwpg.aspx>, 2009.

- [4] N. O. Krahnstoever, K. Z. Tang, and C. W. Yu, "Image Filtering," <http://vision.cse.psu.edu/krahnsto/coursework/cse585/project1/report.html>, 1999.
- [5] Y. C. Lee, "Noise Reduction with Non-Local Mean and Hierarchical Edge Analysis," Master Thesis, Department of Computer Science and Information Engineering, National Taiwan University, 2008.
- [6] S. T. McHugh, "Digital Camera Image Noise" <http://www.cambridgeincolour.com/tutorials/noise.htm>, 2009.
- [7] M. S. Nixon and A. S. Aguado, Feature Extraction and Image Processing, Academic Press, New York, 2008.
- [8] A. Petrosyan and A. Ghazaryan, "Method and System for Digital Image Enhancement," US Application#11/116,408, 2006.
- [9] C. Poynton, "Colour Space Conversions," <http://www.poynton.com/PDFs/coloureq.pdf>, 2009.
- [10] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," Proceedings of IEEE International Conference on Computer Vision, Bombay, India, pp. 839-846, 1998.
- [11] Y. C. Wang, "Hierarchical Noise Reduction," Master Thesis, Department of Computer Science and Information Engineering, National Taiwan University, 2008.
- [12] Wikipedia, "Digital Image," http://en.wikipedia.org/wiki/Digital_image, 2009.
- [13] Wikipedia, "Gaussian Filter," http://en.wikipedia.org/wiki/Gaussian_filter, 2009.
- [14] Wikipedia, "RGB Color Model," http://en.wikipedia.org/wiki/RGB_color_model, 2009.
- [15] Wikipedia, "YCbCr," <http://en.wikipedia.org/wiki/YcbCr>, 2009..