Noise Reduction in Raw Data Domain

Wen-Han Chen(陳文漢), Chiou-Shann Fuh(傅楸善)

Graduate Institute of Networking and Multimedia, National Taiwan University, Taipei, Taiwan

E-mail: r98944034@ntu.edu.tw

***Abstract*—In this paper we propose a noise reduction method in raw data domain. It combined Bayer pattern pixel to determine the three channel values, and effectively downscale image in different sizes. Determine the pixel's property–Edge or NonEdge, and eliminate noise. Finally combine these images and write back corresponding pixel position.**

Keywords-component; noise reduction, raw data

1. INTRODUCTION

Noise reduction is an important factor in image quality. Common noise reduction methods usually work after color interpolation and gamma correction. After these image pipeline functions, noise will be amplified. Noise reduction before image pipeline process will avoid these effects.

We use H. Y. Shen's method [5] to denoise. H. Y. Shen's method [5] is very powerful in noise reduction. We improve the method for raw data.

First we use bilinear interpolation to get the other channel information. Afterwards, we will get the different channel values, and we use these values to get different frequency information in different-size image; use suggested mask to define each pixel whether it is edge or nonedge. We also need to correct all the possible errors, such as broken edge or singular edge. After determining edge pixel, sometimes artifact edges appear. We use adjust function–edge cushioning to correct artificial edge. Find edge pixel direction in minimum gradient. We use eight direction masks to calculate each mask gradient and choose the minimum value represent the pixel corresponding direction. Like above correct edge pixel function, use the same way to correct edge pixel gradient. Set a threshold in luminance and chroma domain. Smooth luminance noise and chroma noise. Finally we combine the four images in different coefficients and write each channel value in corresponding Bayer pattern position.

We try to keep more detail but still denoise. Thus we enhance the edge by unsharp mask, multi-threshold, and more masks to determine edge pixel.

2. RELATED WORK

2.1 Noise Source of Image Sensor

There are three common noise types in digital image.

(1) Fixed-pattern noise

We always classify into hot or dead pixels. Some defective sensors have bad pixels. We can observe these noises in high exposure time or low ISO (International Standards Organization) Speed with effective sensor.

(2) Read Noise

When CCD (Charge-Coupled Device) or CMOS (Complementary Metal-Oxide-Semiconductor) catches a photon to transform electrons, it always has some statistic error in this process. Moreover, read noise is affected by the temperature largely.

(3) Dark current Noise

Even in the dark environment, sensors still produce electrons and electric current. Generally, sensor always generates electric current when the temperature is over -273°C. Moreover, this electric current produces noise in digital image.

2.2 Data Format in Digital Camera

Initially, sensors catch the different channel values in digital image, called ‘raw data’. It arranges different color channel values on this image. In the same position, we just get one channel value. If we want to get the other channel information, we need to use a function to transform these data: color interpolation.

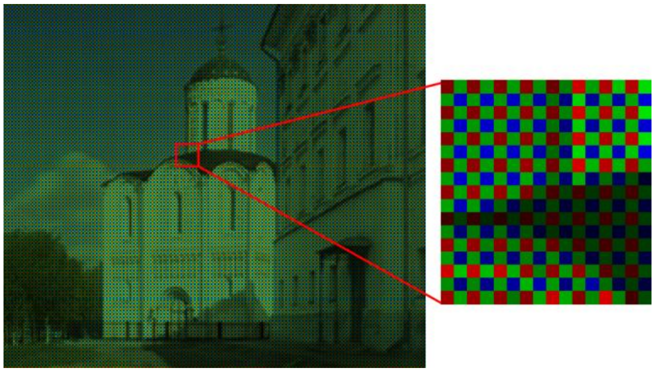


Figure 1: This figure shows that the real world image in raw data format. The image information in arranged color channel value with different luminance intensity.

2.3 Unsharp Mask

It is a common technique to sharpen image. The basic concept is to get entire image difference between blur image and original image. First, we blur entire image with small mask, such as 3*3 mask. Adopt weight average method to blur entire image. Second, get a difference between original image and blur image. Finally, set a threshold and add this difference to the original image.

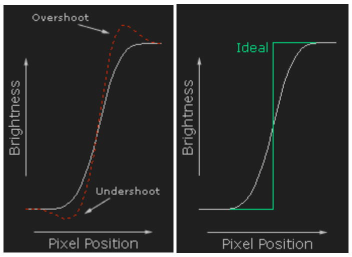


Figure 2: The left image is real world image. The right image is ideal case.

Figure 2 shows that adding difference back to original image raises contrast successfully. But sometimes we still need to avoid halo artifact.

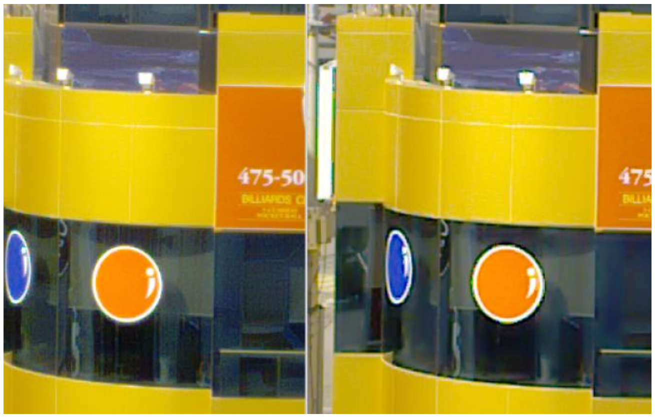


Figure 3: The left picture is original image; the right picture is after unsharp mask function with halo artifact.

2.4 Noise Level Measurement

SNR (Signal-to-Noise Ratio) is a common way to calculate the noise in a known image. In general, high quality image has high SNR.

$$\mathrm{SNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{VS}}{\mathrm{VN}}\right)$$

$$\mathrm{VS} = \left(\frac{1}{\mathrm{N}}\right) \sum_{\mathrm{N}} (\mathrm{I}(\mathrm{i},\mathrm{j}) \ - \ \mu_{\mathrm{s}} \ )^2$$

$$\mathrm{VN} = \left(\frac{1}{\mathrm{N}}\right) \sum_{\mathrm{N}} (\mathrm{I}'^{(\mathrm{i},\mathrm{j})} - \mathrm{I}(\mathrm{i},\mathrm{j}) - \ \mu_{\mathrm{n}} \ )^2$$

$$\mu_s = (\frac{1}{N})\sum_{N} I(i,j)$$

$$\mu_n = (\frac{1}{N})\sum_{N} (I(i,j)' - I(i,j))$$

VS: the gray-level image variance

VN: the noise variance

N: the total pixel number of the image

I(i, j): the original image pixel value at *(i, j)*

I'(i, j): the noise image pixel value at *(i, j)*

3. ORIGINAL HIERARCHICAL METHOD

The hierarchical method decomposes the image into multi-scaled images first. We want to get the frequency information about image and reduce the image size by half in four levels.

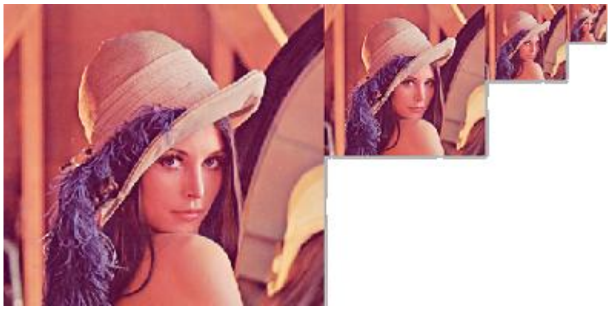


Figure 4: This figure shows that how to reduce the image size by half in four levels.

Next, we want to preserve edge and reduce noise, so we use different masks to determine which pixel is edge or not. These masks have many choices in H.Y. Shen's method [5]. Use the absolute difference between neighbors and middle pixel to determine middle pixel in edge label.

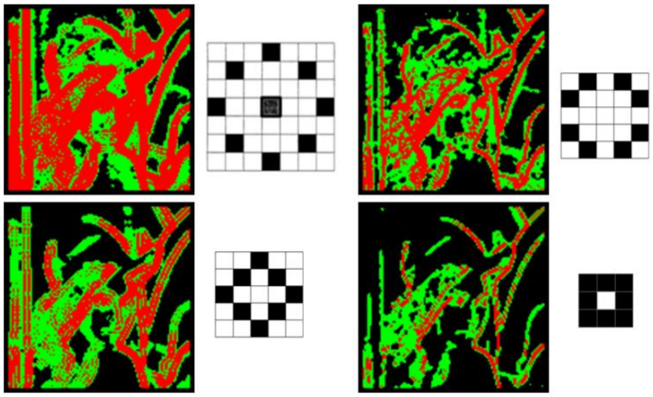


Figure 5: This figure shows that different masks catch edge pixel ability. Where the mask much smaller, it can catch edge more accurately.

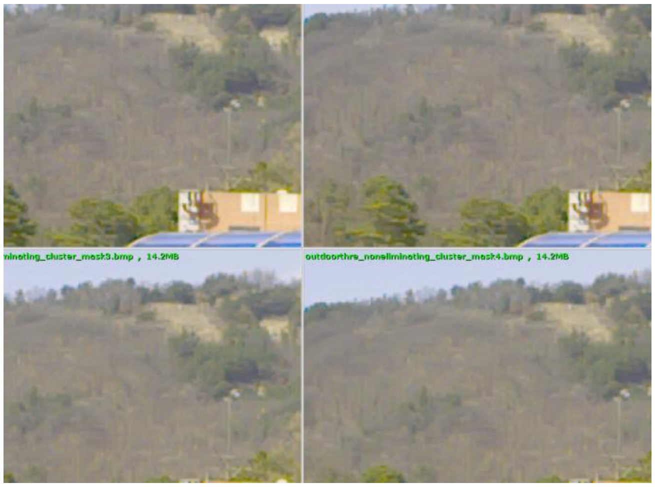
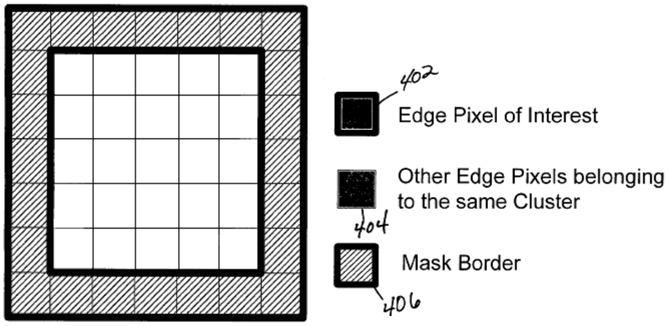

minating_cluster_mask3.bmp , 14.2MB
outdoorthre_noneliminating_cluster_mask4.bmp , 14.2MB

Figure 6: This figure shows that different masks will affect how to catch edge pixels in test image. The left upper picture is the larger mask; the right lower picture is the smaller mask.

Sometimes the edge may be broken or mislabeled, the adjust function will fix broken edge pixels and eliminate singular edge pixel. According to edge property, if middle pixel is an edge, its neighbors are edges, too. Obviously, setting a threshold to judge neighbor's label will correct edge successfully. In this function we also have more masks to fit different requests. Generally, noise cluster is always identified by mask to pixels around the middle pixel. Via this property, we can set a mask with border, if we find many edge pixels around middle pixel, but we can not find any edge pixel in the border of mask. By this judgment, this mask presumes it is a noise cluster, and eliminates it. In H.Y. Shen’s method [5], this function can be canceled if we want to keep more detail. Usually, the mask size is 5*5~15*15 pixels.



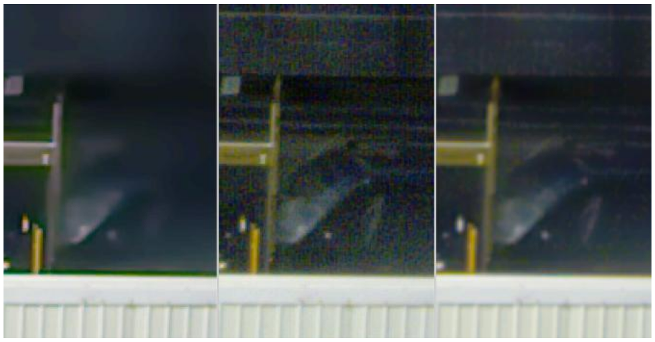
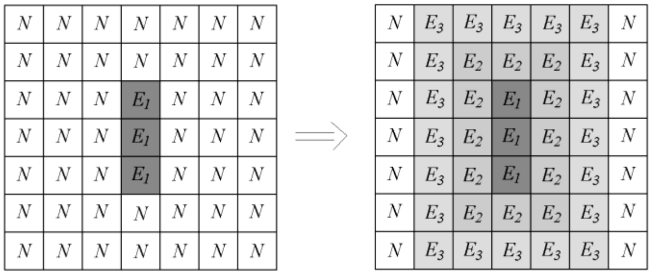


Figure 7: The left picture is eliminating cluster; the middle is original picture; the right picture is disable eliminating cluster.

In Figure 7, our experiment shows that this function effect detail of image apparently. In previous step we can get a map used to label which pixel is edge or not in different edge labels. Sometimes around these strongest edges are nonedges. Human eyes like gradient edge, so we need to make edge cushioning.



N	N	N	N	N	N	N
N	N	N	N	N	N	N
N	N	N	E_1	N	N	N
N	N	N	E_1	N	N	N
N	N	N	E_1	N	N	N
N	N	N	N	N	N	N
N	N	N	N	N	N	N

N	E_3	E_3	E_3	E_3	E_3	N
N	E_3	E_2	E_2	E_2	E_3	N
N	E_3	E_2	E_1	E_2	E_3	N
N	E_3	E_2	E_1	E_2	E_3	N
N	E_3	E_2	E_1	E_2	E_3	N
N	E_3	E_2	E_2	E_2	E_3	N
N	E_3	E_3	E_3	E_3	E_3	N

Figure 8: The left picture is original case. The right picture is after edge cushioning.

After edge cushioning, calculate edge gradient by different masks in luminance and color space. Each mask will produce a gradient value. Choose the minimum to represent middle pixel gradient. But it will produce the same problem like previous step, mislabeled gradient or singular gradient. Thus we use the same way to correct it. In this function, we use a 3*3 mask to correct the edge pixel gradient, start from the upper left of image and finish in the lower right of image.

After previous work, smooth the luminance and chromatic values of the edge pixels.

$$D_k = \left|Y_o - Y_k\right|$$

If $\mathrm{D_k} < \mathrm{T}_{\mathrm{lum}}$

$$W_k = Y_k \times \left(T_{lum} - D_k\right)$$

where Y_{o} is the luminance value of the edge pixel of interest; Y_k is the luminance value of k-pixel in the mask; T_{lum} is the threshold parameter; and W_k is the weighted value of the mask pixel.

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

where L'_o is the smoothed luminance value.

Finally, combine different frequency images to generate final image. Combine each layer by different coefficients.

4. PROPOSED METHOD

Our principal goal is to preserve more detail, but smooth noise within limits. To approach this goal, first we get the raw data for color interpolation or downsampling.

Downsampling is that we use 2*2 block represent a pixel's three channel's value. Not only downscale entire image but get the other channel's values.

Each position will get the 3 channel values. Use this information to do color space transformation in YSbSr [2] color space. In this step we transform raw data successfully. Then we use H. Y. Shen's method [5] for noise reduction, but sometimes edge will be blurred too much. To solve this problem, we try to use unsharp mask [3] and get good result. Finally, inverse color space from YSbSr [2] to raw data format. Write back each value in corresponding position and get the raw data. Raw data input and raw data output are our requirement.

YSbSr [2] is a new color space for our experiment.

$$\begin{bmatrix} Y \\ Sb \\ Sr \end{bmatrix} = \begin{bmatrix} 0.6460 & 0.6880 & 0.6660 \\ -1.0 & 0.2120 & 0.7880 \\ -0.3220 & 1.0 & -0.6780 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 0.5 & -0.6077 & -0.2152 \\ 0.5 & 0.12 & 0.6306 \\ 0.5 & 0.4655 & -0.4427 \end{bmatrix} \begin{bmatrix} Y \\ Sb \\ Sr \end{bmatrix} .$$

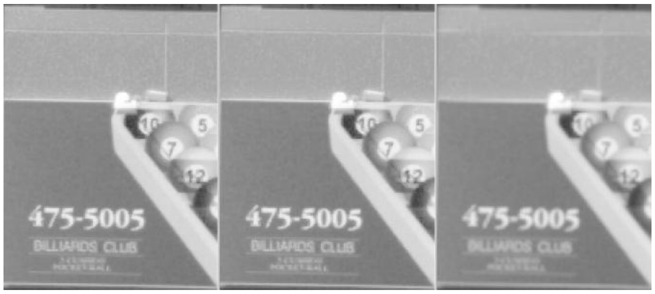


Figure 9: The left picture is original image; middle picture is YSbSr color space; and right picture is YCbCr.

In Figure 9, the other parameters and smoothing algorithm are the same. We can find that the middle picture keeps more detail but smooth noise within limits in luminance space.

Because raw data can not be compared directly, we use the same image pipeline process and parameters. Apical simulator will produce raw data image with its noise reduction algorithm and use the same image pipeline process. Apical's image is our comparison image.

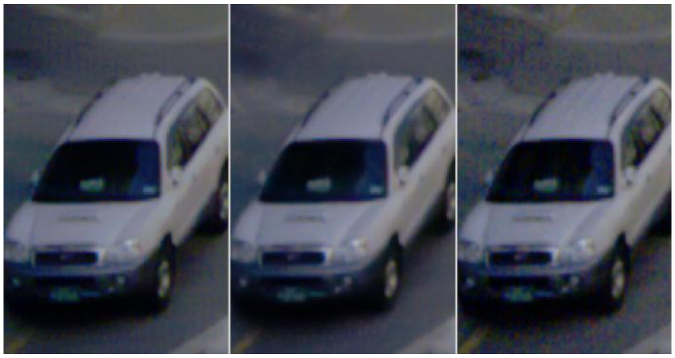
5. EXPERIMENT RESULTS

Original Image 1

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

Multi-Thresholds



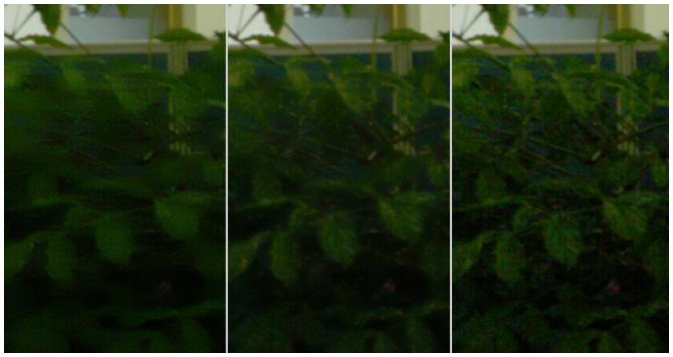
Apical: 5 votes our method: 16 votes original

Original Image 2

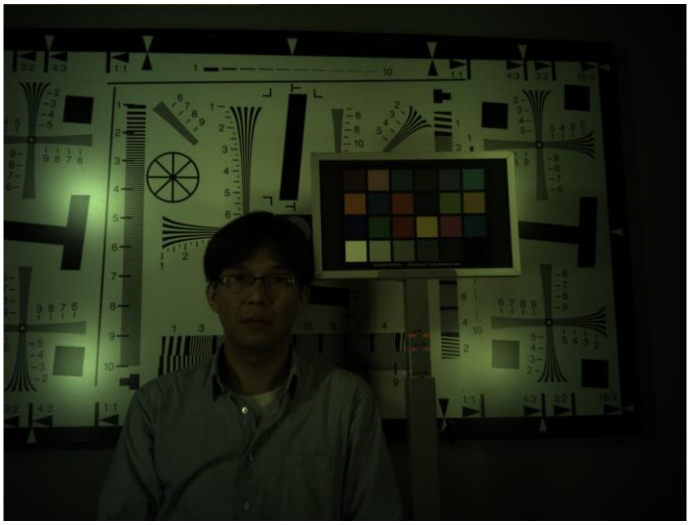
Luminance threshold	25,20,15,15,5,5,10,10, 20,16,12,12,4,4,8,8, 15,12,9,9,3,3,6,6, 10,8,6,6,2,2,6,6
Chroma threshold	10,6,2,6

Luminance smoothing	15,15,15, 5
Chroma threshold	30

Multi-Thresholds



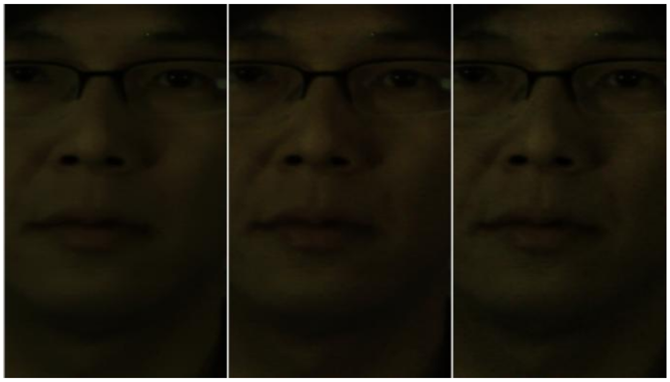
Apical: 3 votes our method: 18 votes original



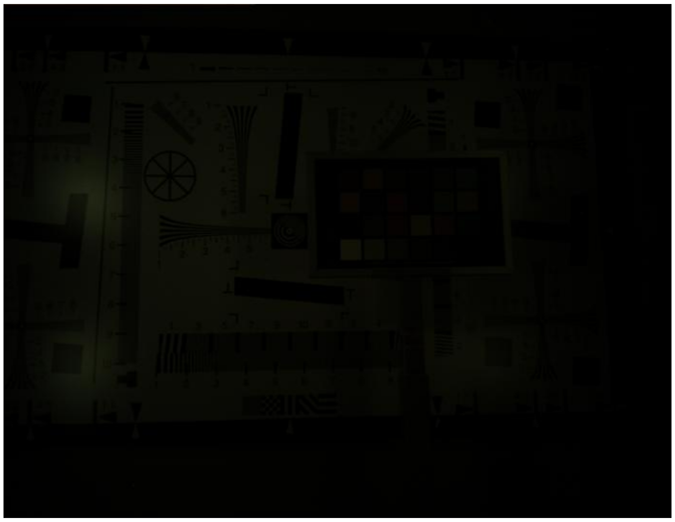
Original Image 3

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

Multi-Thresholds



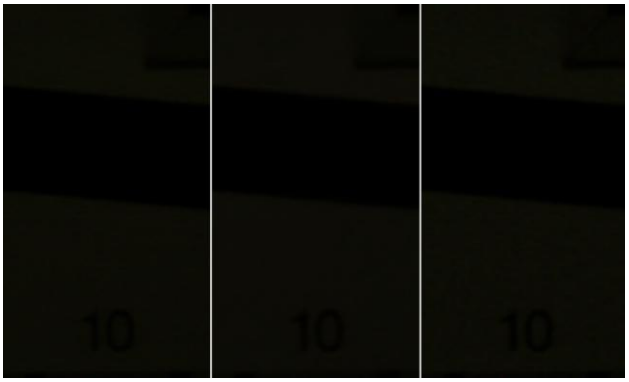
Apical: 8 votes our method: 13 votes original



Original Image 4

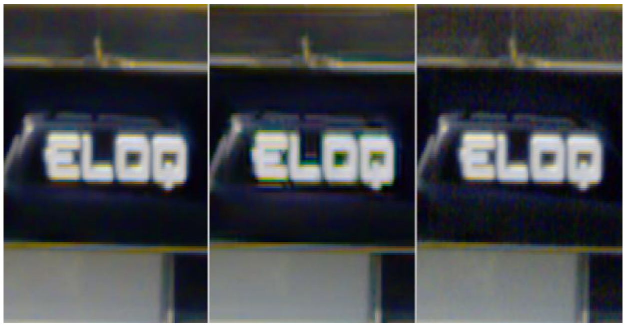
Luminance threshold	5,55,55,55,55,55,55,55, 55,55,55,55,55,55,55,55, 25,35,35,35,35,35,35,35, 15,15,15,15,15,15,15,15
Chroma threshold	75,25,25,25
Luminance smoothing	5,5,5, 5
Chroma threshold	5

Multi-Thresholds



Apical: 2 votes our method: 19 votes original

In comparison with downsample methods.

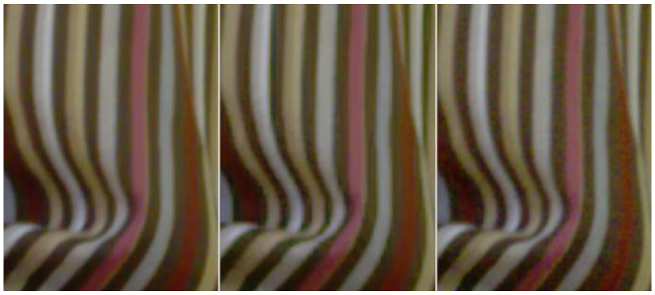


Bi-linear: 20votes
Time: 71 sec
SNR: 23.1197 dB

Downsample: 1vote
Time: 15 sec
SNR: 16.6108 dB

original

Bi-linear: 2 votes downsample: 19 vote original



Bi-linear: 2 votes
Time: 72 sec
SNR: 19.7787 dB

Downsample: 19 votes
Time: 15 sec
SNR: 17.3505 dB

original

6. Conclusion and Future Work

We can denoise well by our proposed method in raw data domain. Noise reduction is an important factor, but we still need to keep more detail. Creating a clear image is our goal. In the future, we try to access raw data directly by downscale image size which will not only speed up our proposed method, but access raw data directly. We still face some problems about this direct access, such as boundary and zigzag effect.

References

[1] Y. Y. Chuang, "Cameras," http://www.csie.ntu.edu.tw/~cyy/courses/vfx/10spring/lectures/, 2010.

[2]H. M. Kim, W. S. Kim and D. S.Cho, "A New Color Transform for RGB Coding," *Proceedings of International Conference on Image Processing*, Singapore, pp.107-110, 2004.

[3]S. McHugh, "Unsharp Mask," http://www.cambridgeincolour.com/tutorials/unsharp-mask.htm,2010.

[4]R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. Sander. "Demosaicking Methods for Bayer Color Arrays," *Jounal of Electronic Imaging*, Vol. 11, No. 3, pp. 306-315, 2002.

[5]H. Y. Shen, "New Hierarchical Noise Reduction," Master Thesis, Department of Computer Science and Information Engineering, National Taiwan University, 2009.

[6]Wikipedia,"singal-to-noise ratio," http://en.wikipedia.org/wiki/Signal-to-noise_ratio, 2010.