國立交通大學

電控工程研究所

碩士論文

基於邊緣偵測之短波紅外線影像壞點偵測與校正

Edge Detection Based SWIR Image Bad Pixel Detection and Correction

研究生：吳泳發

指導教授：張志永

中華民國一百年七月

基於邊緣偵測之短波紅外線影像壞點偵測與校正

Edge Detection Based SWIR Image Bad Pixel Detection and Correction

學　　生：吳泳發　　　　Student : Yung-Fa Wu

指導教授：張志永　　　　Advisor : Jyh-Yeong Chang

國立交通大學
電機工程學系
碩士論文

A Thesis
Submitted to Department of Electrical Engineering
College of Electrical and Computer Engineering
National Chiao-Tung University
in Partial Fulfillment of the Requirements
for the Degree of Master in
Electrical and Control Engineering
July 2011
Hsinchu, Taiwan, Republic of China

中華民國一百年七月

基於邊緣偵測之短波紅外線影像壞點偵測與校正

學生:吳泳發　　指導教授: 張志永博士

國立交通大學電機與控制工程研究所

摘要

本論文使用邊緣偵測和模糊規則偵測短波紅外線影像壞點像素並藉由兩次中值濾波器校正壞點像素。此外，對於短波紅外線影像像素非均勻現象，我們使用兩點式校正方法。

為了提高壞點校正效能，我們也提出改進基於向量次序統計之彩色邊緣偵測技術的方法，我們的邊緣偵測方法包含兩個部份，第一部份，我們利用模糊梯度的概念來估測每個處理像素的梯度方向，並且根據此方向來調整相對應的視窗方位；第二部部分依向量次序統計計算向量平均距離(VMD)，如此一來，整合了向量次序統計與模糊梯度的偵測方法，因此能夠產生更為穩健的邊緣偵測響應。更進一步，我們將此技術整合到一個新的門檻偵測方法，此方法依據影像內容自動作最佳化調整門檻，而不需要手動選取。由測試彩色合成影像與實際影像的數據顯示，我們的自動彩色邊緣偵測是非常方便與可靠的，期許能夠更進一步提高紅外線影像壞點偵測效能。

Edge Detection Based SWIR Image Bad Pixel Detection and Correction

STUDENT: Yung-Fa Wu ADVISOR: Dr. Jyh-Yeong Chang

Institute of Electrical and Control Engineering
National Chiao-Tung University

ABSTRACT

In this thesis, we first use edge detection and fuzzy rules to find bad pixel map of a SWIR sensor. Then we employ two median filters sequentially to correct them. Moreover, we apply two-point correction method to correct non-uniformity among pixels of SWIR sensor.

To enhance the tools for bad pixel correction, we have also proposed a new color edge detector based on vector order statistics. The proposed detector consists of two stages. In the first stage, we use fuzzy gradient to estimate the direction of the gradient for every processing pixel in the image and adjust the corresponding processing window according to this detected direction for reliable edge detection setup. The second stage computes the vector mean distance (VMD) based on vector order statistics. Hence, the proposed detector, which integrates vector order statistics and fuzzy gradient, can provide more robust response for edge detection. Furthermore, we also combine the edge detector to our proposed thresholding method, which can automatically determine an optimal threshold and be adaptive to different image contents without manual intervention. Thus, the excellent results by our proposed edge detection scheme demonstrate that it is very user friendly and confident. This edge detection scheme could also be promising for better detecting bad pixels of a SWIR image sensor.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor, Dr. Jyh-Yeong Chang for valuable suggestions, guidance, support and inspiration he provided. Without his advice, it is impossible to complete this research. Thanks are also given to all of my lab members for their suggestion and discussion.

Finally, I would like to express my deepest gratitude to my family for their concern, supports and encouragements.

Contents

摘要 i

ABSTRACT ii

ACKNOWLEDGEMENTS iii

Contents iv

List of Figures vii

List of Tables xii

Chapter 1 Introduction 1

1.1 Motivation 1

1.2 NUC and Bad Pixel correction 3

1.3 Color edge detection 4

1.4 Automatic Thresholding Technique 5

1.5 Thesis Outline 6

Chapter 2 The Improvement of NUC and Bad Pixel Correction 7

2.1 NUC 7

2.2 Bad Pixel correction 8

Chapter 3 Results of NUC and Bad Pixel Correction 15

3.1 Results of NUC 15

3.2 Results of Bad Pixel Correction 18

Chapter 4 Edge Detection Techniques for Bad Pixel Detection 27

4.1 Vector Order Statistics 28

4.1.1 Vector Order Statistics Review 28

4.1.2 Characteristics of Vector Order Statistics 29

4.2 VMD Detector 31

4.3 Automatic Threshold Selection 39

4.3.1 Determine Parameter Set 39

4.3.2 The Best Threshold Selection 42

4.4 Experimental Result of Comparison with Other Color Edge Detect 44

4.4.1 Quantitative Evaluation 44

4.4.2 Quantitative Evaluation in Synthetic Color Images 46

(A) Quantitative Performance Comparison 56

4.4.3 Comparison of Nature Color Images 58

(A) Example 1, “papers” image 59

(B) Example 2, “Lena” image 61

(C) Example 3, “Road” image ...…………………..………………….63

(D) Example 4, “Chair” image ...………………….……………….65

Chapter 5 Conclusion ……………………………………………………….……68

References …………………………………………………………………...69

List of Figures

Fig. 1.1 The flowchart of our SWIR sensor. ……………………………….4

Fig. 2.1 (a) The completely off image. (b) The 75% exposure image. ……………...7

Fig. 2.2 The Sobel operators. ……………………………………………….……....9

Fig. 2.3 (a) The "Monitor" image. (b) The "Words" image. (c) The "Two-persons" image. …………………………..…………………………………………10

Fig. 2.4 By Sobel edge detection, (a) "Monitor" image, (b) "Words" image, and (c) "Two-persons" image. ……………………………………………………………11

Fig. 2.5 Four-directional neighborhood operator. ……………………………………12

Fig. 2.6 The result of four-directional neighborhood operator and thresholding. ...12

Fig. 2.7 The resulting image of at least two out of three images which is three Sobel edge map images. …………………………………………………………....13

Fig. 2.8 The bad pixel map by using fuzzy rule 2. ……………………………....14

Fig. 3.1 The resulting image of 75% exposure image after NUC. ………………. 15

Fig. 3.2 The resulting images after NUC. (a) "Monitor" image, (b) "Words" image, and (c) "Two-persons" image. ……………………………………….…………. 16

Fig. 3.3 Sobel edge map images of (a) "Monitor" raw image, (b) "Monitor" image after NUC, (c) "Words" raw image, (d) "Words" image after NUC, (e) "Two-persons" raw image, and (f) "Two-persons" image after NUC. ……...……….…………. 17

Fig. 3.4 (a) “Monitor” raw image. (b) Four boundaries replaced of “Monitor” image after NUC. (c) “Words” raw image. (d) Four boundaries replaced of “Words” image after NUC. (e) “Two-persons” raw image. (f) Four boundaries replaced of “Two-persons” image after NUC. 19

Fig. 3.5 The bad pixel map after correcting four boundaries bad pixels. 20

Fig. 3.6 The resulting of 5×5 median filter corrected (a) “Monitor” image. (b) “Words” image. (c) “Two-persons” image. 21

Fig. 3.7 The resulting images after four-directional neighborhood operator, on Fig. 3.6. (a) “Monitor” image, (b) “Words” image, and (c) “Two-persons” image. 22

Fig. 3.8 The resulting images of 3×3 median filter corrected images of Fig. 3.6. (a) “Monitor” image, (b) “Words” image, and (c) “Two-persons image. 23

Fig. 3.9 (a) “Monitor” row image. (b) The resulting image of NUC and bad pixel corrected image of “Monitor.” (c) “Words” row image. (d) The resulting image of NUC and bad pixel corrected image of “Words.” (e) “Two-persons” row image. (f) The resulting image of NUC and bad pixel corrected image of “Two-persons.” 25

Fig. 4.1 The membership function corresponds to *large*. 34

Fig. 4.2 (a) The neighborhood around the central pixel $I(i, j)$. (b) Pixel indicated in gray are used to compute the fuzzy gradient value of pixel $I(i, j)$ for *NW* direction. 35

Fig. 4.3 The membership function corresponds to *absolute value.* ……....................38

Fig. 4.4 The symbolic graphic of the choice of parameter set [20]. …....................40

Fig. 4.5 (a) One kind of original image. (b) One kind of original image. 47

Fig. 4.6 Edge detection results of the 256×256 synthetic image Sample 1 detected by different color edge detectors. (a) Original image. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector. …………………………………………………………………....49 – 50

Fig. 4.7 Edge detection results of the 256×256 synthetic image Sample 2 detected by different color edge detectors. (a) Original image. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector. ……………………………………………………………………51 – 52

Fig. 4.8 Edge detection results of the 256×256 synthetic image Sample 3 detected by different color edge detectors. (a) Original image. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector. ....53 – 54

Fig. 4.9 Edge detection results of the 256×256 synthetic image Sample 4 detected by different color edge detectors. (a) Original image. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector. ....55 – 56

Fig. 4.10 Edge detection results of the "Peppers" image detected by different color edge detectors. (a) Original image. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector. ....60 – 61

Fig. 4.11 Edge detection results of the "Lena" image detected by different color edge detectors. (a) Original image. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector. ...62 – 63

Fig. 4.12 Fig. 4.12 Edge detection results of the “Road” image detected by different color edge detectors. (a) Original image. (a’) gray-level histogram equalized of image (a), used for visual aide and comparison only. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector. ………………………………………………………………..64 – 65

Fig. 4.13 Edge detection results of the “Chair” image detected by different color edge detectors. (a) Original image. (a’) gray-level histogram equalized of image (a), used for visual aide and comparison only. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector. ………………………………………………………………...66 – 67

List of Tables

TABLE 4.1 The fuzzy gradient in each direction. ……………………………......36

TABLE 4.2 The average evaluation results of the thirteen 256x256 synthetic color images detected by the following detectors. …..57

TABLE 4.3 The average evaluation results of the eleven 256×256 synthetic color images detected by the following detectors. ………………………………………57

TABLE 4.4 The average evaluation results of the total synthetic color images detected by the following detectors. …………………………………………………58

Chapter 1 Introduction

1.1 Motivation

With the rapid technological development, the wavelength of visible light has become familiar, and infrared sensing has also been widely applied in various fields of low/no light environments. One of infrared wavelength is short-wave infrared (SWIR), choose bandwidth is close to visible red light band, with a higher level of imaging performance. In addition, SWIR band also used in the following application providing high resolution images: military area, it can be used as the enemy recognition systems and surveillance systems, in low/no light or poor visibility (fog or smoke in the environment) cases the infrared image sensor can also be used to identify the surrounding environment. Health care, the 2003 global outbreak of Severe Acute Respiratory Syndrome (SARS) and the 2009 outbreak of new influenza (H1N1) caused global panic. By use of the infrared image sensor to detect body temperature, we not only can reduce the chance of infection but also can substantially increase the efficiency of body temperature measurement. When infrared sensor falls into the infrared spectral band images, it is known as thermal imaging. When the body of a person reflects the infrared band, we need some special infrared sensors to detect it, and then coupled with the infrared lens to compose. It is well known that infrared focal plane array (IRFPA) has non-uniformity and bad pixels in the produced sensor cell. Hence, the infrared image must do non-uniformity correction (NUC) and bad pixel correction. Bad pixel is the pixel that does not respond (non responsive) i.e., dark situation (commonly) or always responsive i.e. In the SWIR sensor bad pixel saturation is usually happened.

In the low light military operations situations, infrared image processing must be fast and high efficiency. Because military SWIR sensor has to be lightweight and easy to used. Therefore in NUC, the most popular reference-based correction method is the so-called "two-point" correction method where two uniform sources of known intensity are sequentially imaged [1], [2]. Edge detection plays an important role in bad pixel detection. In image processing, edge detection is also very useful on tasks such as segmentation, pattern recognition, object tracking, and image coding. The performance of these problems is greatly affected by good edge detection. In gray image edge detection, the Canny edge detector has become a standard [3]. Its non-maximal suppression and thresholding with hysteresis stages produce thin and well-connected edge maps. Edges will not be easily detected in grayscale images when different objects have different hues but have equal intensities. In this case, the color cue is lost during grayscale conversion. They are treated like one big object in the scene when they cannot be distinguished in grayscale. Moreover, edge detection is sometimes difficult in low contrast images but rather sufficient results can be obtained in color images.

In order to produce more reliable edges, color edge detection has become popular. Humans can differentiate thousands of colors compared to about 256 shades of gray; hence, grayscale images do not carry all the edge information that human visual system (HVS) can detect. In [4], they indicate that luminance component makes up 90% of all edge points in a color image but the remaining 10% can be crucial for subsequent techniques that rely on edges in an image; in some cases the additional information provided by color is of utmost importance. Multi-dimensional nature of color makes it more challenging to detect edges in color images, and often increases the computational complexity threefold compared to gray scale edge detection. Hence, color edge detection algorithms accept from the beginning that all of the efforts are to find the remaining 10% of the edges.

In this thesis, we use NUC to correct infrared image that has non-uniformity and correct bad pixel.

In the companion edge detection scheme developed for better bad pixel detection, we propose an improvement of color edge detector based on vector order statistics (VOS) [5], [6]. We use the concept of fuzzy gradient [7], [8] to calculate the direction of the gradient for every pixel in the image. For using an adjustable window according to the direction of the gradient, it is more accurate to calculate the local maximum edge response for every pixel, and an automatic threshold technique is adaptive to threshold the local maximum edge response for the image content.

1.2 NUC and Bad Pixel Correction

Our SWIR sensor has to be fast, easy to use, and lightweight. Consequently, the two-point correction method is simple and thus will be useful. In bad pixel correction, we will find some bad pixel maps. The Sobel edge is easy to use and fast. Because of bad pixels include non-responsive, dark pixels or always responsive pixels. The completely off image and 75% exposure image should be used. We will use images which are completely off image and 75% exposure image to find bad pixel map. Then, we can correct the bad pixels from bad pixel map. Our SWIR sensor flowchart is illustrated in Fig. 1.1 below.

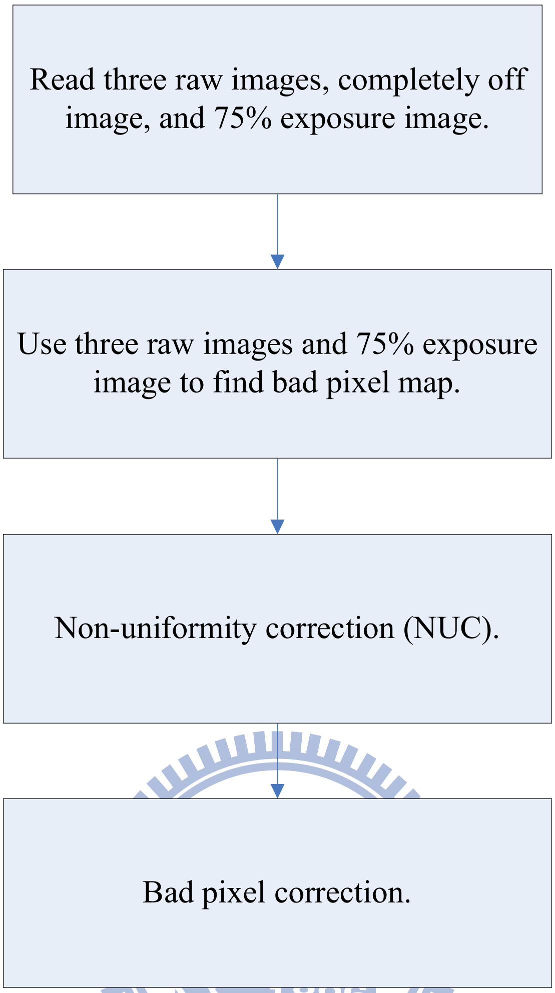

Read three raw images, completely off image, and 75% exposure image.
Use three raw images and 75% exposure image to find bad pixel map.
Non-uniformity correction (NUC).
Bad pixel correction.

Fig 1.1 The flowchart of our SWIR sensor.

1.3 Color Edge Detection

In the review paper on color image segmentation, Ruzon and Tomasi [9] go further and group color edge detection methods into three classifications: output fusion methods, multidimensional gradient methods and vector methods. Output fusion methods apply single-channel edge detection techniques to each color plane and then combine the results.

In multidimensional gradient methods, the gradients from the individual channels are recombined before the edge decision, giving increase to a single edge calculation.

In [10], Scharcanski and Venetsanopoulos have proposed VOS-based approach. Trahanias and Venetsanopoulos used the reduced ordering (R-Ordering) by the VOS edge detectors of [5], [6]. The robust color morphological gradient (RCMG) edge detector [11] recognizes the maximum and minimum pixels in one process, however it does not discriminate between them. It is in contrast to the VOS edge detectors that sort the pixels in ascending order from the vector median to the vector extremum. The matrices are summed over all channels and the edge magnitude and direction given by the principal eigenvalue and the related eigenvector, respectively. Variations of this approach have been used by Cumani [12].

The difficult problem is how to combine the channels to give a final result that is with both output fusion and multidimensional gradient methods. For example, the simplest VOS operator is the vector range edge detector that measures the distance between the lowest and highest ranked vectors, i.e., the vector median and the vector extremum, respectively. The minimum vector dispersion (MVD) was proposed are shown to be the most effective in increasing the robustness to noise. However, the MVD is unable to provide an estimate of edge direction.

1.4 Automatic Thresholding Technique

Thresholding is a fundamental technique applied in many image processing applications. In robust machine vision systems, it would be important to automate the edge thresholding process which is adaptive to different image contents without manual interposition.

There are many thresholding algorithms published in the literature. The Otsu [13] algorithm is based on discriminant analysis and uses the zeroth-order and the first-order cumulative moments of the histogram for calculating the value of the thresholding level. The Rosin algorithm [14] fits a straight line from the peak of the intensity histogram to the last non-empty bin. The point of maximum deviation between the line and the histogram curve will usually be located at a corner which is selected as the threshold value. The new feature image proposed by Rakesh [15] makes it easier to determine hysteresis thresholds.

It is a difficult assignment to selecting an appropriate thresholding. The problem is that different algorithms typically produce different results since they make different assumptions about the image content. Therefore, we will introduce an automatic thresholding method that can find the best hysteresis thresholds from all possible parameters.

1.5 Thesis Outline

The thesis is organized as follows. The basic concepts and technique concerning the NUC and bad pixel correction introduced in Chapter 2. In Chapter 3, the results of our SWIR sensor which is introduced in Chapter 2 are shown and compared. In Chapter 4, we describe our edge detection and automatic thresholding method and compared the experiment results of our automatic color edge detection techniques. At last, we conclude this thesis with a discussion in Chapter 5.

Chapter 2 The Improvement of NUC and Bad Pixel Correction

2.1 NUC

To implement two-point correction method for image sensor, we can assume

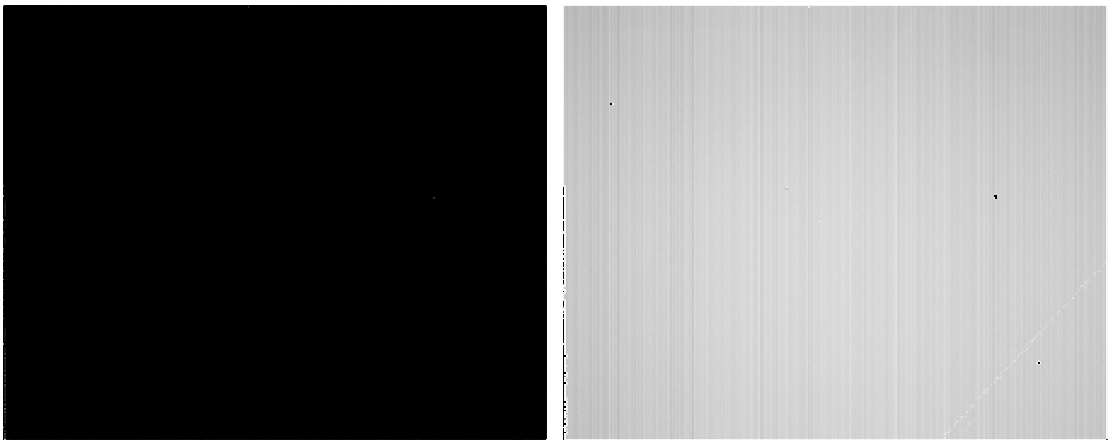
$$\begin{cases} P_2^i = \alpha_i T_2 + \beta_i \\ P_1^i = \alpha_i T_1 + \beta_i \end{cases}, \tag{2.1}$$

where P_1^i and P_2^i represents the i-th pixel value in completely off image and 75% exposure image, respectively. The α_i and β_i represents gain and offset of i-th pixel. The T_1 and T_2 represents illumination level. By using Eq. (2.1), we can assume $T_2 = kT_1$, then

$$\alpha_i = \frac{P_2^i - P_1^i}{(k-1)T_1} \tag{2.2}$$

$$\beta_i = P_1^i - \frac{P_2^i - P_1^i}{(k-1)} \tag{2.3}$$

Calculate averages that completely off image and 75% exposure image which are shown in Figs. 2.1(a)–(b) as $\overline{P_1}$ and $\overline{P_2}$, respectively.



(a) (b)

Fig. 2.1 (a) The completely off image. (b) The 75% exposure image.

Assuming under T_1 and T_2, all pixels output should be the same level. In other word, $\overline{P}_1$ and $\overline{P}_2$ after correction

$$\begin{cases} \overline{P}_2 = \alpha_i A_i T_2 + \beta_i B_i \\ \overline{P}_1 = \alpha_i A_i T_1 + \beta_i B_i \end{cases} \tag{2.4}$$

By using Eq. (2.2), and Eq. (2.3), then

$$A_i = \frac{\left(\overline{P}_2 - \overline{P}_1\right)}{\alpha_i\left(T_2 - T_1\right)} = \frac{\left(\overline{P}_2 - \overline{P}_1\right)}{P_2^i - P_1^i} \tag{2.5}$$

$$B_i = \frac{1}{\beta_i}\left(\overline{P}_1 - \alpha_i A_i T_1\right) \tag{2.6}$$

We want to correct infrared image which is under unknown illumination level T_x, then

$$P_x^i = \alpha_i T_x^i + \beta_i \tag{2.7}$$

Consequently, corrected pixel output signal should be

$$P_c^i = \alpha_i A_i T_x^i + \beta_i B_i = \overline{P}_1 + \frac{\left(P_x^i - P_1^i\right)\left(\overline{P}_2 - \overline{P}_1\right)}{P_2^i - P_1^i} \tag{2.8}$$

Therefore, it can correct directly without any other illumination level that have to using laboratory for reference.

2.2 Bad Pixel Correction

In bad pixel correction, Sobel edge detection is very important which is fast and easy. The masks show in Fig. 2.2, called the Sobel operators.

-1	0	+1
-2	0	+2
-1	0	+1

(a) *x*-direction.

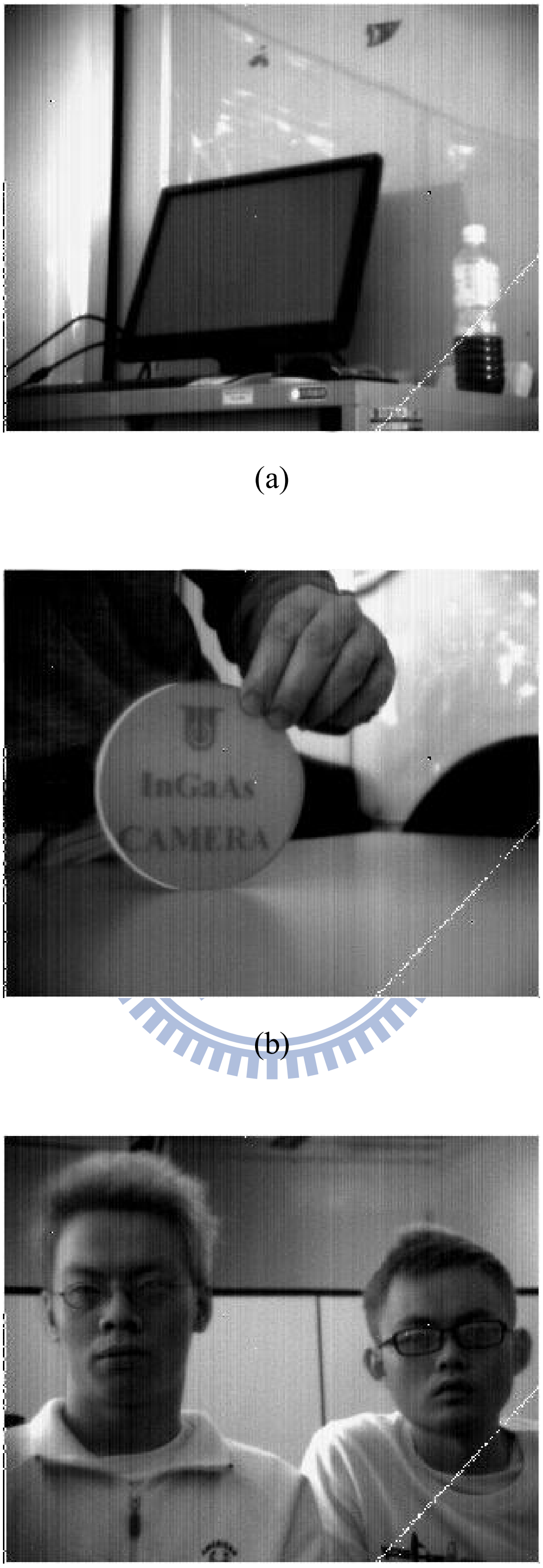
+1	+2	+1
0	0	0
-1	-2	-1

(b) *y*-direction.

Fig. 2.2 The Sobel operators.

The difference between the third and first columns of the 3×3 image region approximates the derivative in the *x*-direction as shown in Fig. 2.2(a); and the difference between the third and first rows approximates the derivative in the *y*-direction as shown in Fig. 2.2(b). The idea behind using a weight value of 2 is to achieve some smoothing by giving more importance to the center point. Note that the coefficients in all the masks shown in Fig. 2.1 sum to 0, indicating that they would give a response of 0 in an area of constant gray level, as expected of a derivative operator.

First, we want to detect bad pixel map by using Sobel edge detection. Consider a grayscale image I with size $m\times n$ be represented by vector $I(i,j)$. By using SWIR sensor, we capture three different images. Therefore, three grayscale images $I_1(i,j)$, $I_2(i,j)$, and $I_3(i,j)$ with size $m\times n$ is shown in Figs. 2.3(a)$-$(c) which is called raw images, which is non-uniform correction is not activated.



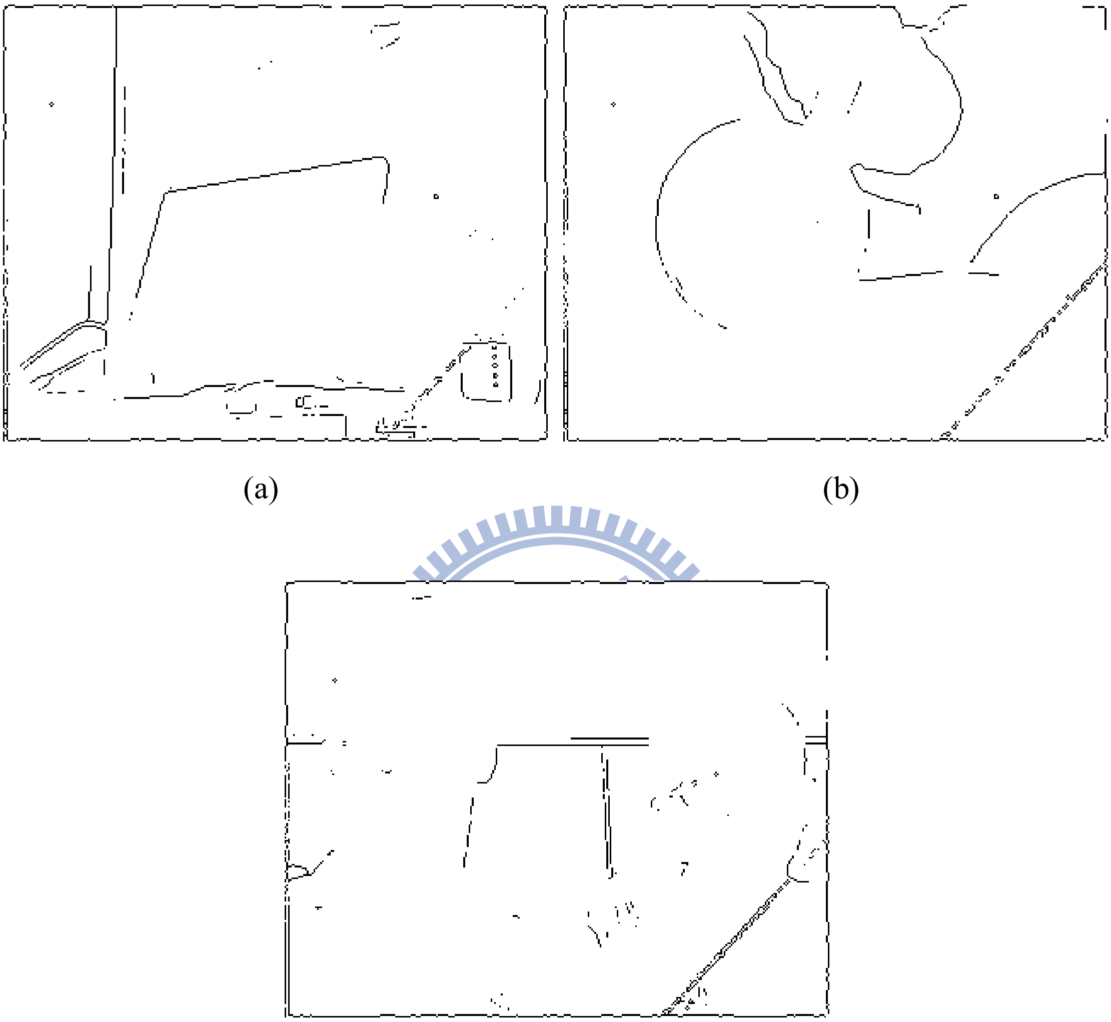
(a)

(b)

(c)

Fig. 2.3 (a) The “Monitor” image. (b) The “Words” image. (c) The “Two-persons” image.

To obtain bad pixels of the image sensor, we first apply Sobel edge operation on these three images. After the operation, we have obtained three edge map images, which is shown in Figs. 2.4(a)$-$(c) and are called S_1, S_2, and S_3, respectively.



(a)

(b)

(c)

Fig. 2.4 By Sobel edge detection, (a) “Monitor” image, (b) “Words” image, and (c) “Two-persons” image.

Then, we calculate four-directional difference $d(i, j)$ in the 3×3 window as shown in Fig. 2.5. The value $d(i, j)$ can be calculated by

$$d(i,j)=\left|I(i,j)-I(i-1,j)\right|+\left|I(i,j)-I(i,j-1)\right|+\left|I(i,j)-I(i+1,j)\right|+\left|I(i,j)-I(i,j+1)\right| , \tag{2.9}$$

where $|\cdot|$ represents an absolute value operator and (i,j) represents the i-th row and j-th column in the grayscale image I.

0	-1	0
-1	4	-1
0	-1	0

Fig. 2.5 Four-directional neighborhood operator.

After calculating $d(i,j)$, we will select a threshold to pick up possible bad pixel candidates. For the 75% exposure image which is shown in Fig. 2.1 (a), we will find bad pixels possibility by using Eq. (2.9) and thresholding, with threshold $value = 40$, which is shown in Fig. 2.6. And it called d_4 image.

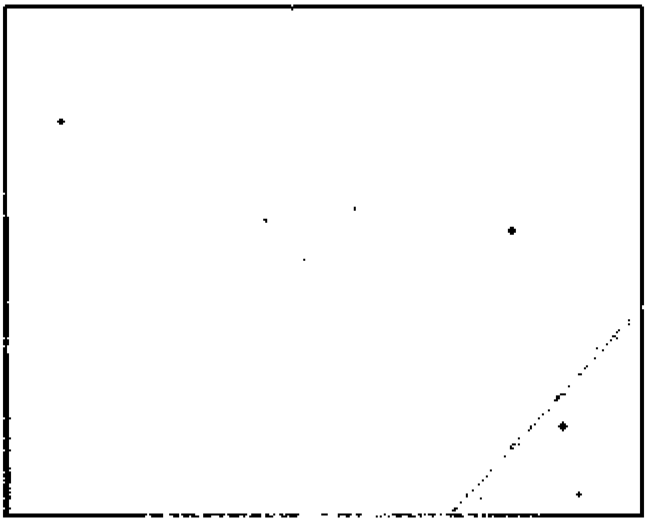


Fig. 2.6 The result of four-directional neighborhood operator and thresholding.

Finally, we calculate the fuzzy derivative donated as $S_{bp}(i,j)$ in each pixel (i,j). This is realized by the following fuzzy rule 1:

Fuzzy Rules 1.

IF $S_1(i,j)$ is an *edge* **AND** $S_2(i,j)$ is an *edge* **AND** $S_3(i,j)$ is an *edge*

OR $S_1(i,j)$ is an *edge* **AND** $S_2(i,j)$ is **not** an *edge* **AND** $S_3(i,j)$ is an *edge*

OR $S_1(i,j)$ is an *edge* **AND** $S_2(i,j)$ is an *edge* **AND** $S_3(i,j)$ is **not** an *edge*

OR $S_1(i,j)$ is **not** an *edge* **AND** $S_2(i,j)$ is an *edge* **AND** $S_3(i,j)$ is an *edge*

THEN $S_{bp}(i,j)$ is an *edge*.

The **AND** operator (**OR** operator) can be the minimum (maximum) that are the well-known triangular norms (together with their dual co-norms) in the fuzzy logic. For the **not** operator, we use the standard negator $N(x)=1-x$ with $x \in [0,1]$. The *edge* is the fuzzy set that was defined the point which is detected by Sobel edge detection. According to fuzzy rule 1, we have S_{bp} edge map is shown in Fig. 2.7.

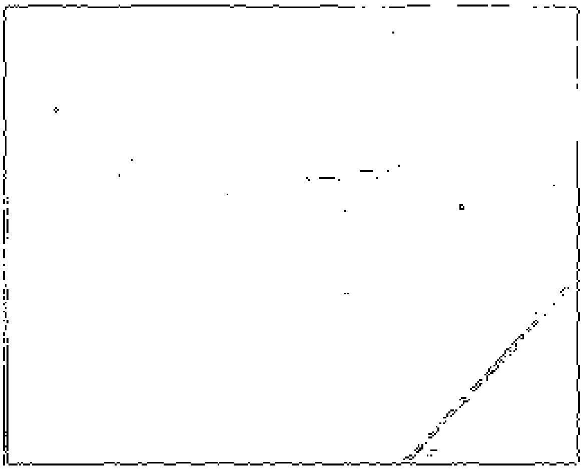


Fig. 2.7 The resulting image of at least two out of three images which is three Sobel edge map images.

Therefore, we use a rule for each edge point. We have two images which are Fig. 2.6 and Fig. 2.7 can estimate BP_{map} bad pixel map following rule 2.

Fuzzy Rules 2.

IF $d_4(i,j)$ is an *edge* **OR** $S_{bp}(i,j)$ is an *edge.*

THEN $BP_{map}(i,j)$ is an *edge.*

The **OR** operator is the maximum in the fuzzy logic. The BP_{map} is shown in Fig. 2.8.

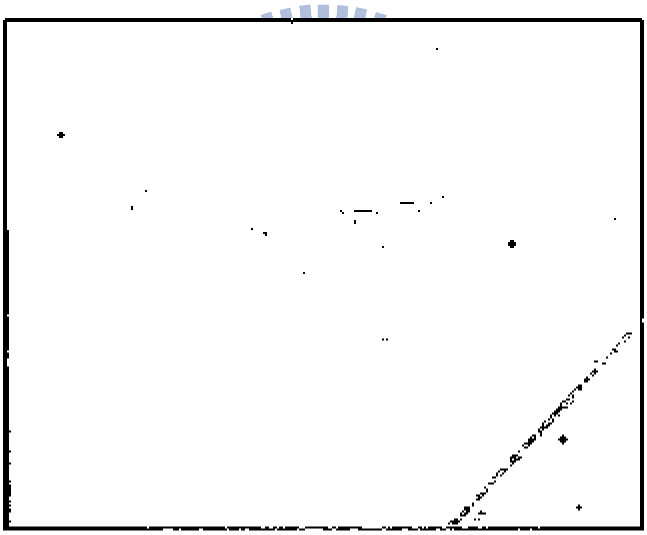


Fig. 2.8 The bad pixel map by using fuzzy rule 2.

Consequently, we can use the bad pixel map which is shown in Fig. 2.8 to do bad pixels correction with median filters shown in Chapter 3.

Chapter 3 Results of NUC and Bad Pixel Correction

3.1 Results of NUC

First, we calculate standard deviation (STD) which is calculated by

$$s = \left(\frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \bar{x} \right)^2 \right)^{\frac{1}{2}}, \tag{3.1}$$

where n is pixels number and $\bar{x}$ is

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \tag{3.2}$$

We will use STD to estimate the performance of the NUC results. Therefore, we use Fig. 2.1(b) as a sample and its STD is 0.1349. It is to be noted that detected bad pixels shown in Fig. 2.8 are not included in calculating STD. By using Eq. (2.8), the result is shown in Fig. 3.1.

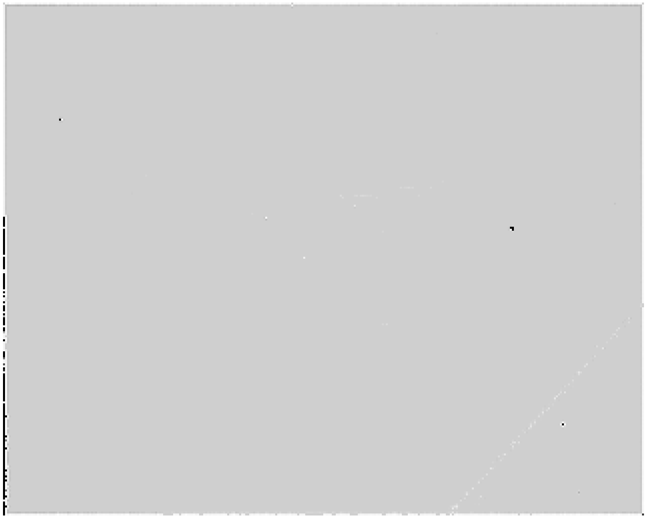
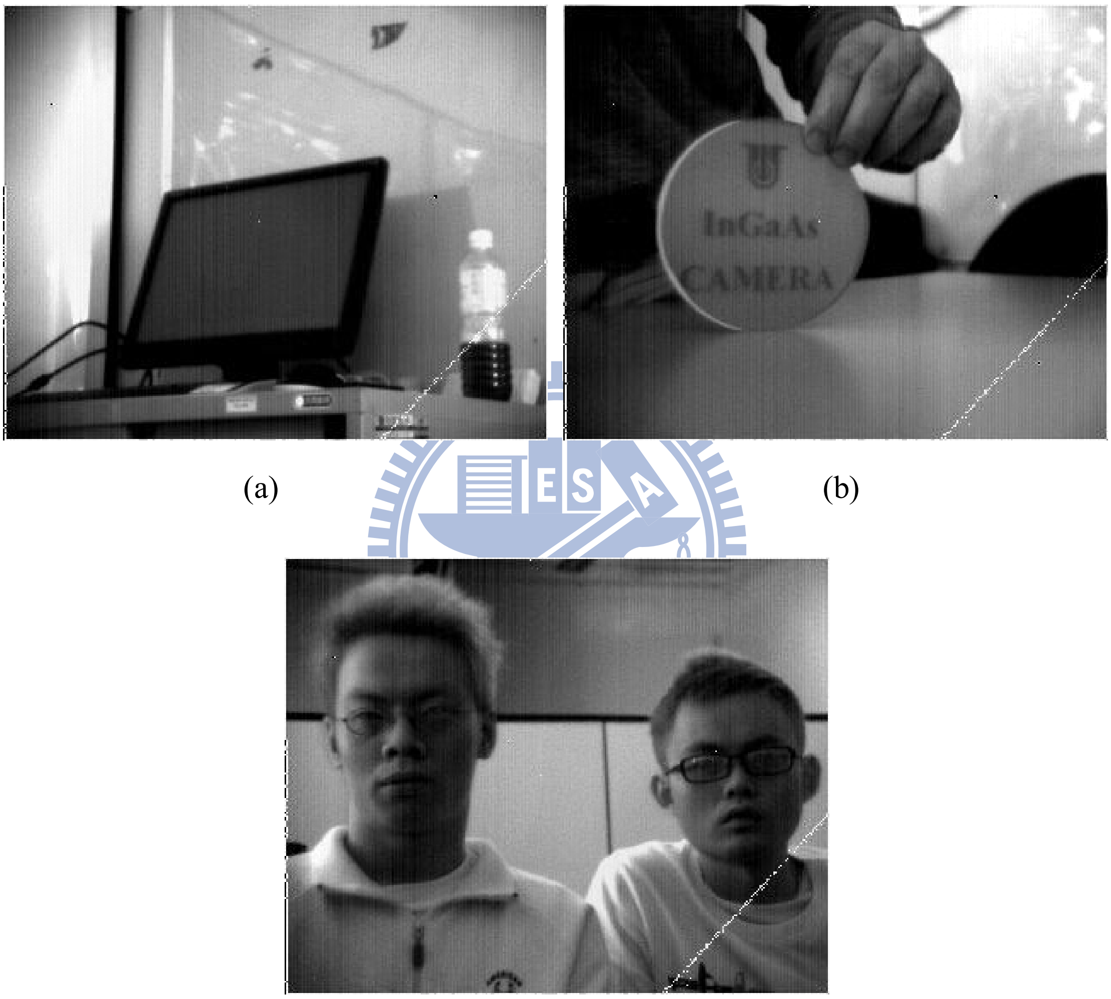


Fig. 3.1 The resulting image of 75% exposure image after NUC.

The STD of Fig. 3.1 becomes 0, in which bad pixels are not included in calculating STD. This great STD reduction after NUC shows its effectiveness in reducing the non-uniformity response among the pixels of image sensors. Applying NUC to "Monitor," "Words," and "Two-persons" images of Figs. 2.3(a) – (c), the corrected images are shown in Figs. 3.2(a) – (c), respectively.



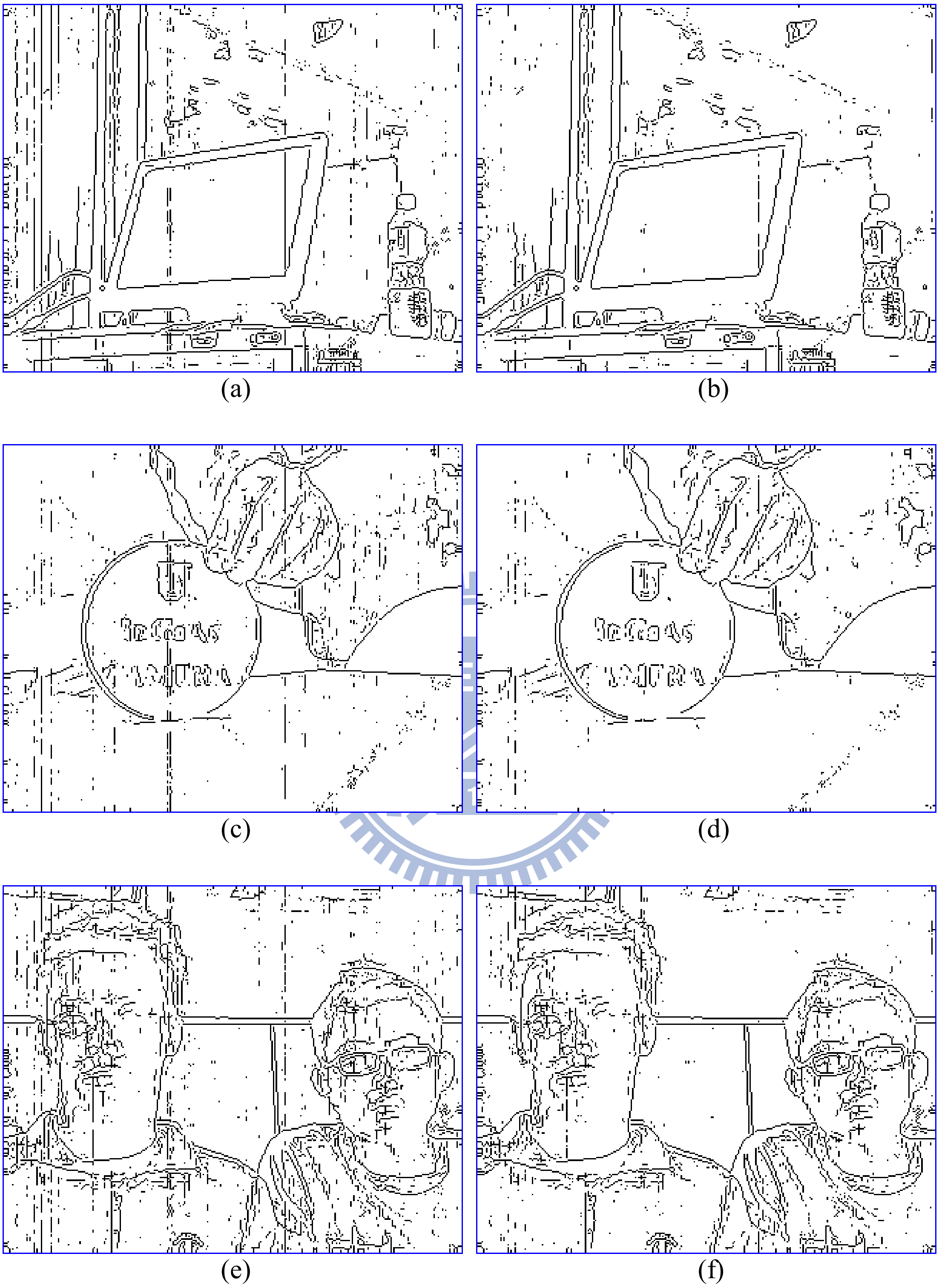
(a)

(b)

(c)

Fig. 3.2 The resulting images after NUC. (a) "Monitor" image, (b) "Words" image, and (c) "Two-persons" image.

To further validate the effectiveness of NUC, we exploit Sobel edge detection with a sensitive threshold to test raw images and NUC corrected images above, leading to Figs. 3.3(a) – (f), respectively.



(a)

(b)

(c)

(d)

(e)

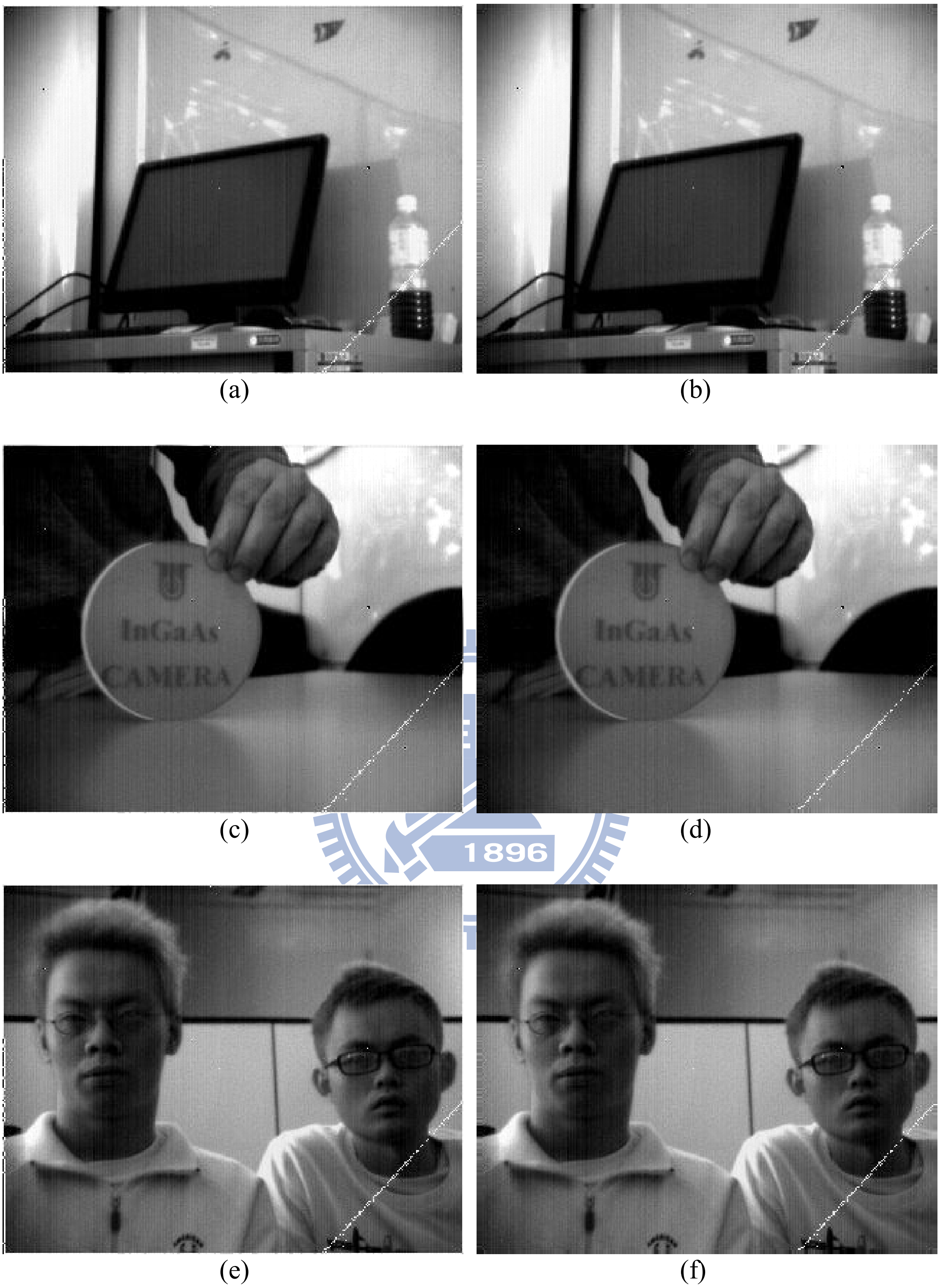
(f)

Fig. 3.3 Sobel edge map images of (a) "Monitor" raw image, (b) "Monitor" image after NUC, (c) "Words" raw image, (d) "Words" image after NUC, (e) "Two-persons" raw image, and (f) "Two-persons" image after NUC.

From these figures in Fig. 3.3, it is easy to see the powerfulness of NUC in reducing the excessive or un-necessary edges due to the non-uniformity of the image sensors. Excessive lines in left-up corner of Fig. 3.3(a) is removed in Fig. 3.3(b), NUC corrected counterpart. From Fig. 3.3(d), it is evident that removed of excessive lines and edges, in tables, wafer, and hand right, of Fig. 3.3(c). Unnecessary lines and edges of left person in Fig. 3.3(e) has been removed in NUC corrected image Fig. 3.3(f).

3.2 Results of Bad Pixel Correction

For the detected bad pixel map as shown in Fig. 2.8, we can use median filters to correct bad pixels. First, you can see in Fig. 2.8 that four boundaries are prone to bad pixels. If a boundary line is almost bad pixels, we employ neighborhood good row or column to replace it. After a boundary line with bad pixel rate exceeding 25% has to be replaced line wise. With this criterion (in effect), the three NUC corrected images are boundary replaced and are shown in Fig. 3.4. Comparison results are as follows in Fig. 3.4.



(a) (b)

(c) (d)

(e) (f)

Fig. 3.4 (a) “Monitor” raw image. (b) Four boundaries replaced of “Monitor” image after NUC. (c) “Words” raw image. (d) Four boundaries replaced of “Words” image after NUC. (e) “Two-persons” raw image. (f) Four boundaries replaced of “Two-persons” image after NUC.

Correcting four boundaries bad pixels, the bad pixel map will be updated as shown in Fig. 3.5.

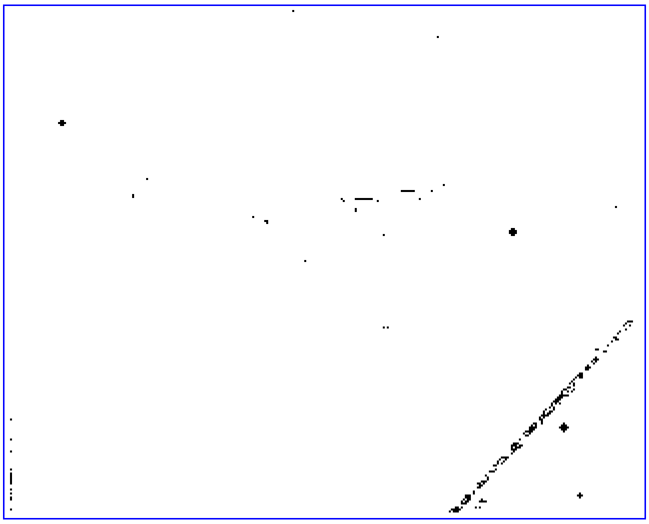
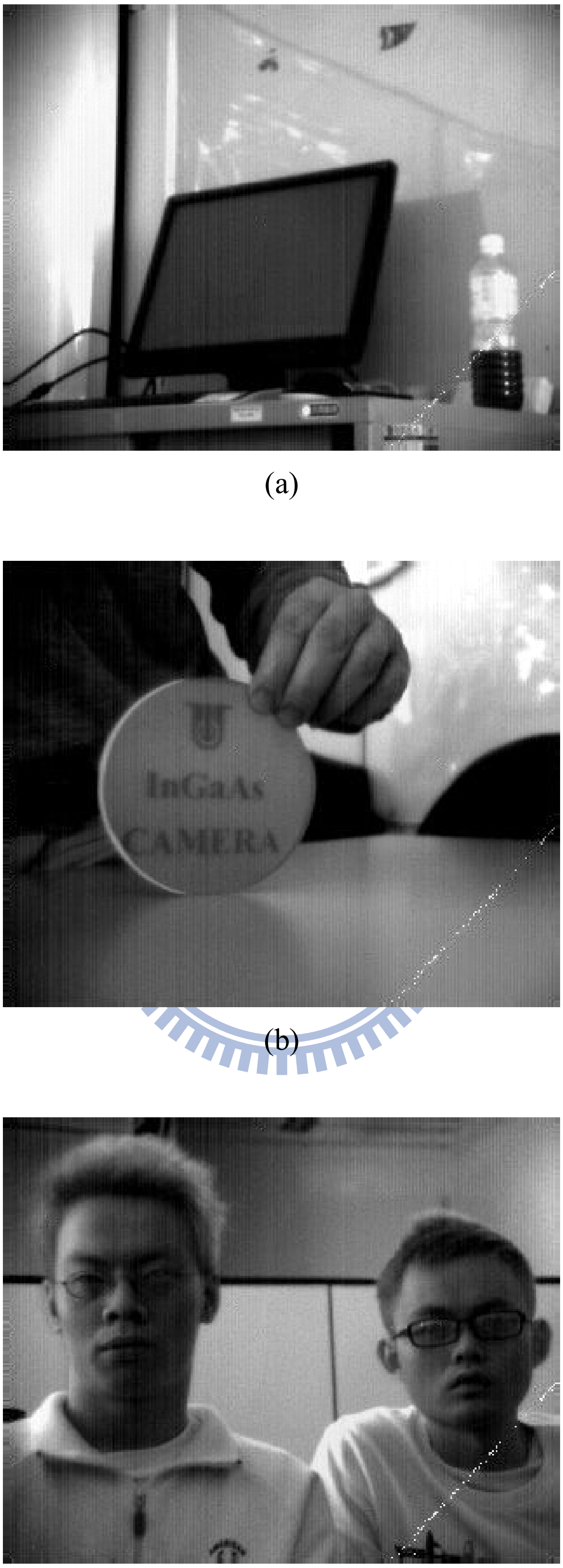


Fig. 3.5 The bad pixel map after correcting four boundaries bad pixels.

Second, we use a 5×5 median filter to correct bad pixels and the results are shown in Figs. 3.6(a) – (c).



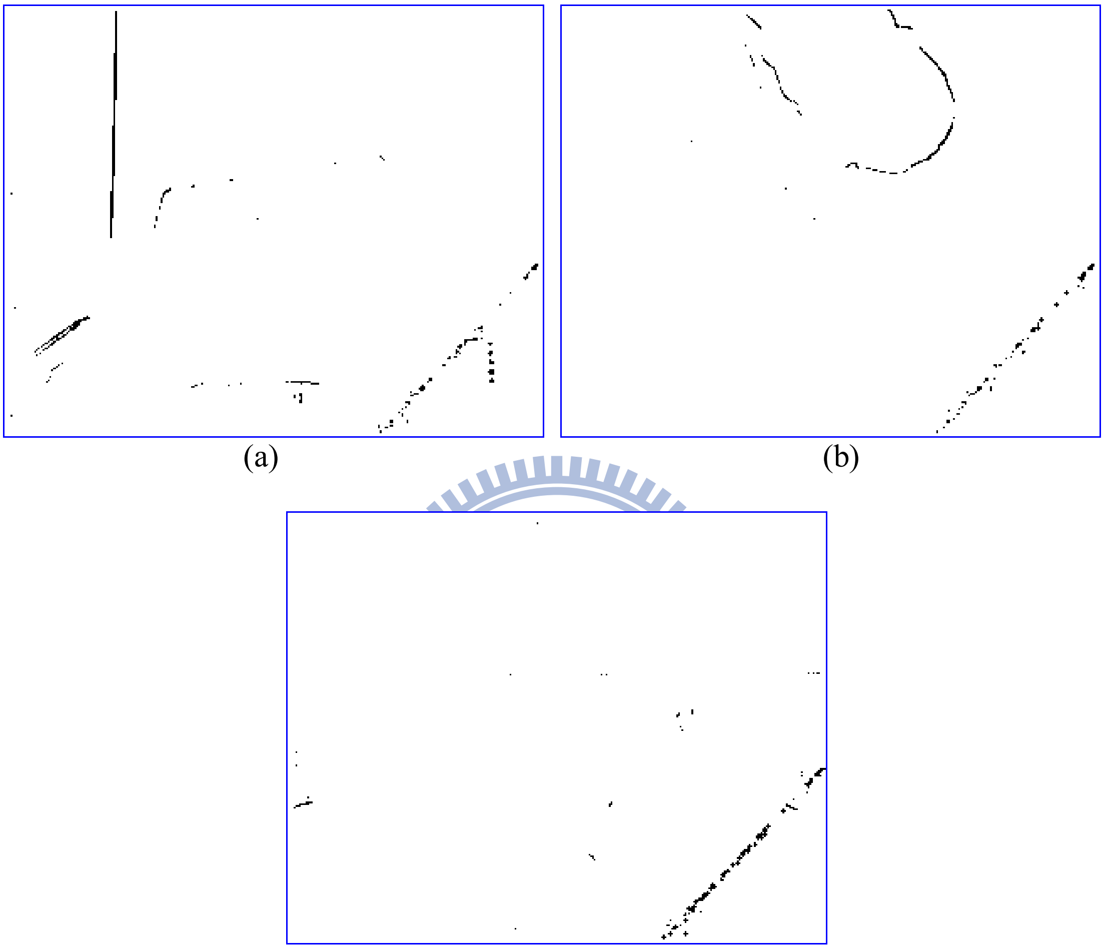
(a)

(b)

(c)

Fig. 3.6 The resulting of 5×5 median filter corrected (a) “Monitor” image. (b) “Words” image. (c) “Two-persons” image.

In Figs. 3.6(a)$-$(c), it can be easily seen that these still are bad pixels on the lower right corner of images. Consequently, we only use Eq. (2.9), four-directional neighborhood operator with threshold $value = 150$, to calculate the bad pixel map again. The bad pixel maps are shown in Figs. 3.7(a)$-$(c).



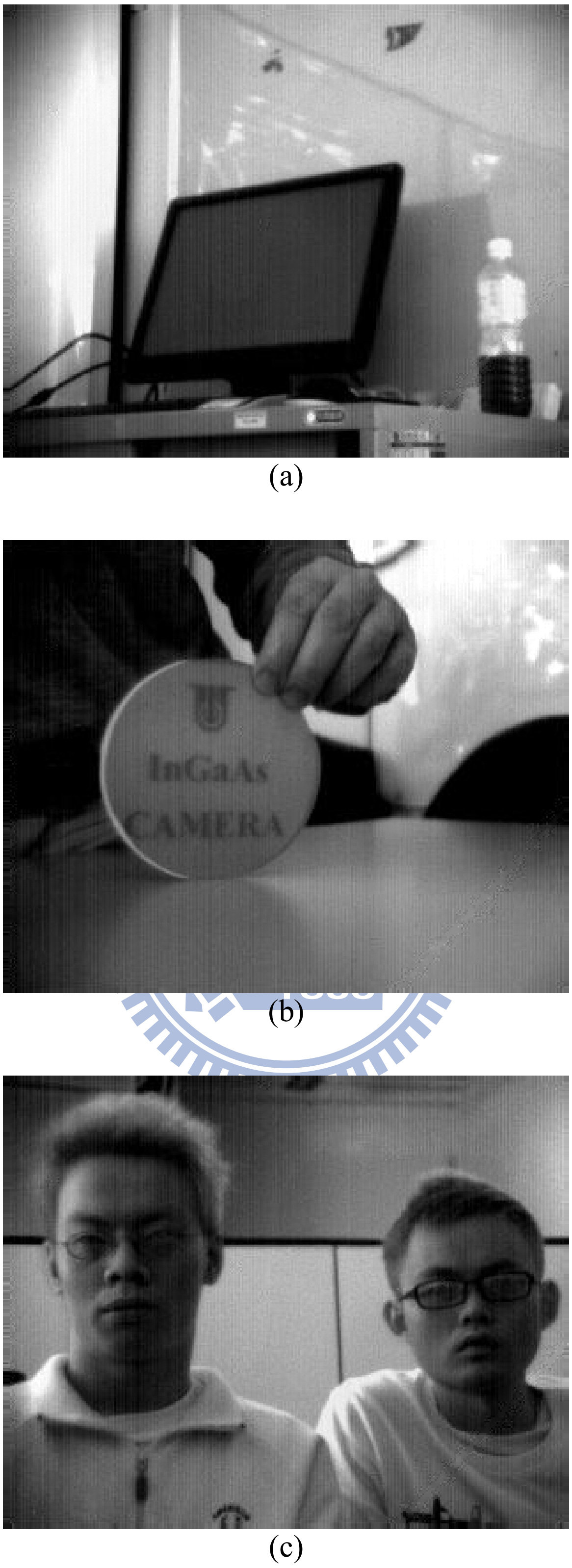
(a)

(b)

(c)

Fig. 3.7 The resulting images after four-directional neighborhood operator, on Fig. 3.6 (a) "Monitor" image, (b) "Words" image, and (c) "Two-persons" image.

In the sequel, we use a 3×3 median filter on Fig. 3.6 to correct bad pixels of Fig. 3.7 again. The resulting images are shown in Figs. 3.8(a)$-$(c).



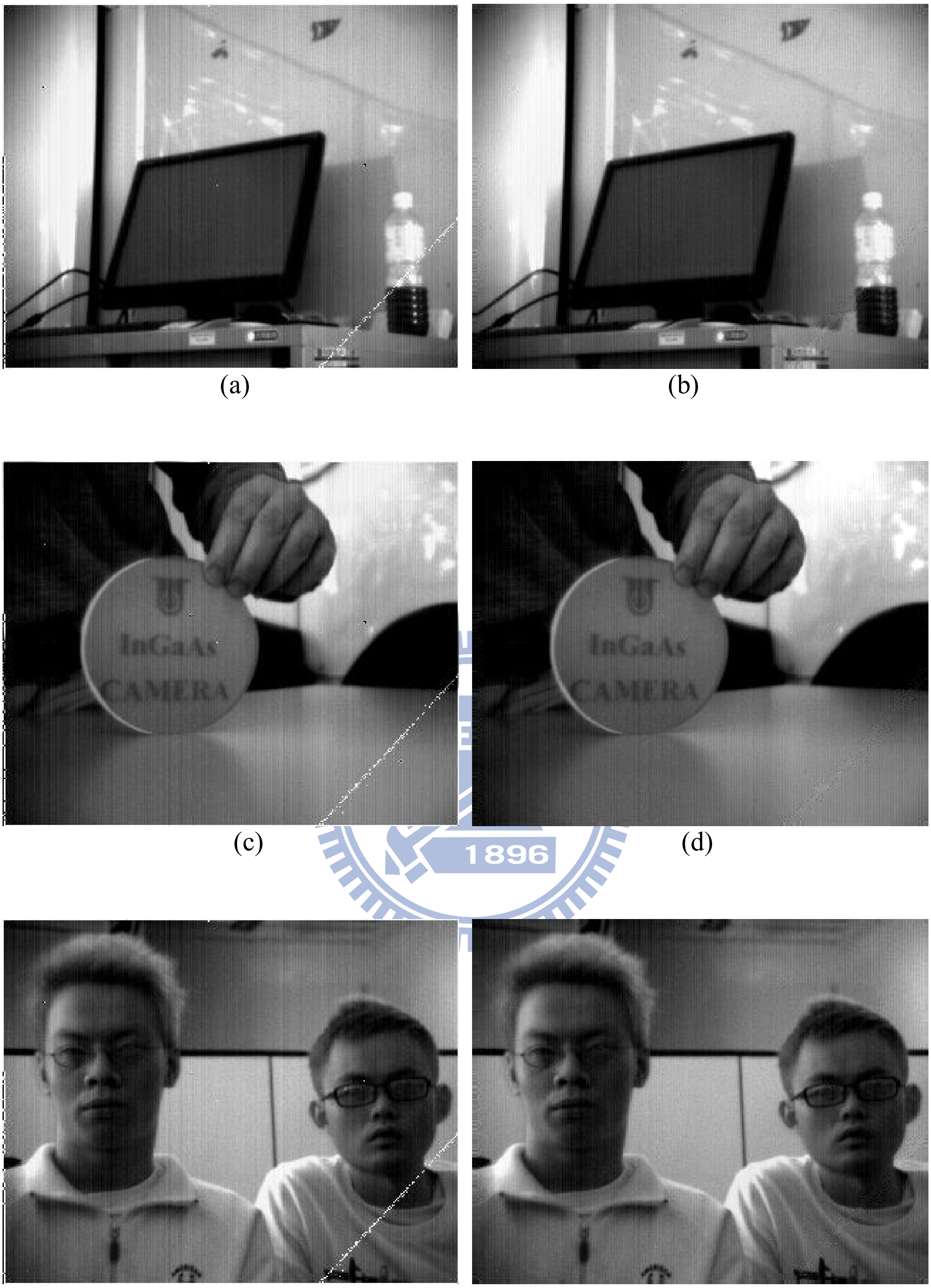
(a)

(b)

(c)

Fig. 3.8 The resulting images of 3×3 median filter corrected images of Fig. 3.6 (a) “Monitor” image, (b) “Words” image, and (c) “Two-persons” image.

Notice that the median filter is useful on bad pixels correction. Moreover, median filter is computational easy and is appropriate for a real-time ROIC implementation. We have also applied bad pixel correction with 3×3 median filter repeatedly for three times, whose results are similar to those obtained by 5×5 median filter and the 3×3 median filter. But it is need three times correction, and it is waste time. Therefore, we try to use 5×5 median filter once and then 3×3 median filter again. The results are good and efficient. Finally, we choose 5×5 median filter once and the 3×3 median filter again. For comparison, the raw images and the corrected images are shown in Figs. 3.9(a) $-$ (f).



(a) (b)

(c) (d)

(e) (f)

Fig. 3.9 (a) “Monitor” row image. (b) The resulting image of NUC and bad pixel corrected image of “Monitor.” (c) “Words” row image. (d) The resulting image of NUC and bad pixel corrected image of “Words.” (e) “Two-persons” row image. (f) The resulting image of NUC and bad pixel corrected image of “Two-persons.”

The difference between the row images and the corrected images are tremendous. We adjust the non-uniformity among image sensor pixels and correct bad pixels by using fast and efficient algorithm. The processed images have been greatly improved by our proposed efficient scheme.

Chapter 4 Edge Detection Techniques for Bad Pixel Detection

Edge detection plays an important role in bad pixel detection of optical sensor chip. In image processing, edge detection is also very useful on tasks such as segmentation, pattern recognition, object tracking, and image coding. The performance of these problems is greatly affected by good edge detection. We use Sobel edge detection and four-directional neighborhood operator to detect bad pixels and test NUC in Chapters 2 and 3. These techniques are basic edge detection methods. In Sobel edge detection and four-directional neighborhood operator, we calculate each pixel’s edge level and select a threshold manually. However, the best edge detection will use an automatic threshold technique to select the best threshold.

Edges will not be detected in grayscale images when neighboring objects have different hues but equal intensities since the color cue is lost during grayscale conversion. Such objects cannot be distinguished in grayscale images. Similar gray tone objects are treated as one big object in the scene. Additionally, edge detection is sometimes difficult in low contrast images but rather sufficient results can be obtained in color images. To obtain more meaningful edges, there has been an increased interest in color edge detection. Humans can differentiate thousands of colors compared to about 256 shades of gray; hence, grayscale images do not carry all the edge information that human visual system (HVS) can detect. In this chapter, we propose an improvement of color edge detector based on vector order statistics. We use the concept of fuzzy gradient to calculate the direction of the gradient for every pixel in the image. For using an adjustable window according to the direction of the gradient, it is more accurate to calculate the local maximum edge response for every pixel, and an automatic threshold technique is adaptive to threshold the local maximum edge response for the image content.

4.1 Vector Order Statistics

4.1.1 Vector Order Statistics Review

Scalar order statistics have played an important role in the design of robust signal analysis techniques. This is due to the fact that any outliers will be located in the extreme ranks in the sorted data. Consequently, these outliers can be isolated and filtered out before the signal is further processed. Ordering of univariate data is well defined and has been extensively studied in order statistics [16]. Let the n random variables X_i, $i = 1, 2, \ldots, n$, be arranged in ascending order of magnitude as

$$X_{(1)} \leq X_{(2)} \leq \ldots \leq X_{(n)} \tag{4.1}$$

Then the i-th random variable $X_{(i)}$ is the so-called ith order statistic. The minimum $X_{(1)}$, the maximum $X_{(n)}$, and the median $X_{(n/2)}$ are among the most important order statistics, resulting the min, the max, and the median filters, respectively.

The concepts are, however, not straightforwardly expanded to multivariate data since there is not any universal way of defining an ordering in multivariate data. There has been a number of ways proposed to perform multivariate data ordering that are generally classified into the ordering of multivariate data [17]: marginal ordering (M-ordering), reduced or aggregate ordering (R-ordering), partial ordering (P-ordering), and conditional ordering (C-ordering).

4.1.2 Characteristics of Vector Order Statistics

Let X represent a p-dimensional multivariate $X=[X_1, X_2, ..., X_p]^T$ where X_l, $l=1, 2, \ldots, p$ are random variables and let X^i, $i=1, 2, \ldots, n$ be an observation of X. Each X^i is a p-dimensional vector $X^i=[X_1^i, X_2^i, ..., X_p^i]^T$.

In M-ordering, the multivariate samples are ordered along each one of the p-dimensions independently. For color signals, this is equivalent to the separable method where each one of the colors is processed independently. The i-th marginal order statistic is the vector $X^{(i)}=[X_1^{(i)}, X_2^{(i)}, ..., X_p^{(i)}]^T$, where $X_r^{(i)}$ is the ith largest element in the r-th channel. The marginal order statistic $X^{(i)}$ may not correspond to any of the original samples $X^1, X^2, ..., X^n$ as it does in one dimension.

In R-ordering, each multivariate observation X^i is reduced to a scalar value d_i according to a distance criterion. A metric that is often used is the generalized distance to some point. The samples are often arranged in ascending order of magnitude of the associated metric value d_i.

In P-ordering, the objective is to partition the data into groups or sets of samples, such that the groups can be distinguished with respect to order, rank, or extremeness. This type of ordering can be accomplished by using the notion of convex hulls. However, the determination of the convex hull is difficult to do in more than two dimensions. Other ways to achieve P-ordering are special partitioning procedures and thus are not preferred. Another drawback associated with P-ordering is that there is no ordering within the groups and thus it is not easily expressed in analytical terms. These properties make P-ordering infeasible for implementation in digital image processing.

In C-ordering, the multivariate samples are ordered conditionally on one of the marginal sets of observations. This has the disadvantage in digital image processing that only the information in one component (channel) is used.

From the above, it is evident that R-ordering is more appropriate for color image processing than the other vector ordering methods. If we employ as a distance metric the aggregate distance of X^i to the set of vectors $X^1, X^2, ..., X^n$, then

$$d_i = \sum_{k=1}^{n} \| X^i - X^k \|, \quad i = 1,\ 2,\ ...,\ n \tag{4.2}$$

where $\| \cdot \|$ represents an appropriate vector norm. The arrangement of the d_is in ascending order $\left(d_{(1)} \leq d_{(2)} \leq ... \leq d_{(n)}\right)$, associates the same ordering to the multivariate X^is.

$$X^{(1)} \leq X^{(2)} \leq ... \leq X^{(n)} \tag{4.3}$$

In the ordered sequence, $X^{(1)}$ is the vector median of the data samples which is introduced by vector median filters [18]. It is defined as the vector contained in the given set whose distance to all other vectors is a minimum. Moreover, vectors appearing in low ranks in the ordered sequence are vectors centrally located in the population, whereas vectors appearing in high ranks are vectors that diverge mostly from the data population. These samples are generally called “outliers.” It follows that this ordering scheme gives a natural definition of the median of a population and of the outliers of a population.

4.2 VMD Detector

For a color image I of size $m \times n$, each pixel location (i, j) is represented by a three-tuple color vector $I(i, j) = \left(I_1(i, j),\ I_2(i, j),\ I_3(i, j)\right)$, in which $I_p(i, j)$ denoting the p-th component of a color space for $i = 1, 2, \dots, m$ and $j = 1, 2, \dots, n$. For each pixel location (i, j), by using a 3×3 window, we compute the local sum of distances to describe the relationship between the current pixel vector $I(i, j)$ and its neighboring pixel vectors. Let $d_l(i, j)$ be the local sum of distances for the current pixel vector $I(i, j)$, then

$$d_l(i, j) = \sum_{k=i-1}^{i+1} \sum_{h=j-1}^{j+1} \left\| \, I(i, j) - I(k, h) \, \right\| \tag{4.4}$$

where $\| \cdot \|$ represents a 2-norm. After we have computed the local sum of distances $d_l(i, j)$ of the current pixel location (i, j), we sort the distance values in the neighboring area in ascending order $d_{l(1)} \le d_{l(2)} \le \dots \le d_{l(9)}$. The distance values $d_{l(1)}$ and $d_{l(9)}$ correspond to the minimum and the maximum of the nine distance values, respectively.

By the concept of R-ordering, the ordering of $d_{l(1)} \le d_{l(2)} \le \dots \le d_{l(9)}$ associates the same ordering to the pixel vectors, $X^{(1)} \le X^{(2)} \le \dots \le X^{(9)}$, which means that $X^{(1)}$ is the pixel vector having the smallest local sum of distances and $X^{(9)}$ is the pixel vector having the largest local sum of distances. Therefore, if the current pixel location (i, j) has an edge, the vector $I(i, j)$ must have a larger response of $d_l(i, j)$.

Although we now obtain the information on the smallest and the largest local sum of distances, the information contained among vectors $X^{(1)}, X^{(2)}, \dots, X^{(9)}$

should also be captured and be useful for edge detection. The maximal variation among vectors is an indication of the distribution of the nine vectors. Since that vectors $X^{(1)}, X^{(2)}, \ldots, X^{(9)}$ correspond to the ordering of the aggregate distances, the confined maximal variation $\mathrm{MV_c}$ among these vectors can be simply defined as

$$MV_c = \max\left(\left\| X^{(i)} - X^{(i+1)} \right\|\right),\ i = 1,\ 2,\ \ldots,\ 8 \tag{4.5}$$

When the value $\mathrm{MV_c}$ is determined, we can also determine the exact two vectors $X^{(i)}$ and $X^{(i+1)}$ which correspond to $\mathrm{MV_c}$. $X^{(i)}$ and $X^{(i+1)}$ further suggest that $X^{(1)}, X^{(2)}, \ldots, X^{(9)}$ can be classified into two clusters: (1) vectors, $X^{(1)}, X^{(2)}, \ldots, X^{(i)}$, from smaller side of the edge, and (2) vectors, $X^{(i+1)}, X^{(i+2)}, \ldots, X^{(9)}$, from larger side of the edge. Let M_s and M_l be the mean vector of the vectors $X^{(1)}, X^{(2)}, \ldots, X^{(i)}$, and the vectors $X^{(i+1)}, X^{(i+2)}, \ldots, X^{(9)}$ respectively. Thus, VMD can be defined as

$$VMD = \left\| M_l - M_s \right\| \tag{4.6}$$

VMD detect the variation between two sides of edge (larger and smaller side) by a distance measure. Consequently, in a uniform area, where all vector values are close to each other, the output of VMD will be small. On the other hand, the output of VMD will be large since M_s and M_l are the mean vectors of two sides of the edge.

The VMD method suffers from the disadvantage of the weak ability for detecting oblique edges due to the fact that its gradient magnitude is derived from the fixed window with the distance between the mean vector of the large side and the mean vector of the small side. In addition, in the presence of noise and for non-ideal edges, the maximal variation that splits the window into the large side and small side may not represent the distribution among the vectors in the fixed window, and then the

VMD may produce an edge response that is not necessarily representative of the real gradient.

To avoid these problems, we use an adjustable window that can rotate its orientation according to the direction of the gradient. In the 3×3 window, we classify the direction of the gradient to four orientations, i.e., $W-E$ direction (0°), $SW-NE$ direction (45°), $S-N$ direction (90°), and $SE-NW$ direction (135°) that can be determined by the fuzzy gradient value which is introduced by the fuzzy image filter [7] and the fuzzy random impulse noise reduction method (FRINR) [8]. Thus, the new proposed method will combine VMD with the fuzzy image filter and FRINR that have the ability to estimate the direction of the gradient for each pixel and adjust the window for more exactly detecting edge response.

In our approach, there are two steps that are used to define the direction of the gradient for each pixel in the color image. First, consider a color image I with size $m \times n$ be represented by color vector $I(i,j) = (I_1(i,j),\ I_2(i,j),\ I_3(i,j))$, in which $I_p(i,j)$ denoting the p-th component of a color space, we calculate $g(i,j)$ and $mg(i,j)$ in the 3×3 window as

$$g(i,j) = \frac{\sum_{k=-1}^{1}\sum_{h=-1}^{1} \| \ I(i+k, j+h) - I(i,j) \ \|}{8} \tag{4.7}$$

$$mg(i,j) = \frac{\sum_{k=-1}^{1}\sum_{h=-1}^{1} \| \ g(i+k, j+h) - g(i,j) \ \|}{8} \tag{4.8}$$

where $\| \cdot \|$ represents a 2-norm and (i,j) represents the *i*-th row and *j*-th column in the color image *I*. Because edge pixels and corrupted impulse noise pixels generally cause large $g(i,j)$ value, we also calculate $mg(i,j)$ that can help us to distinguish edge pixels and noise pixels. To discriminate edge pixels and noise pixels, we can define a fuzzy set denoted as *large*, and it corresponds to the membership function which is shown in Fig. 4.1. We see that we have to determine two important parameters *a* and *b*. The parameters *a* and *b* can be defined as

$$a(i,j) = mg(i,j) \tag{4.9}$$

$$b(i,j) = a(i,j) + 0.2a(i,j) \tag{4.10}$$

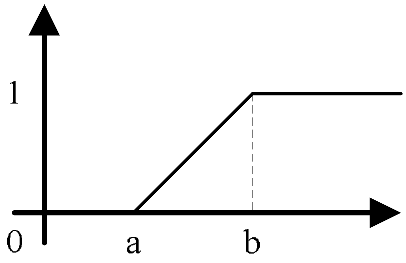


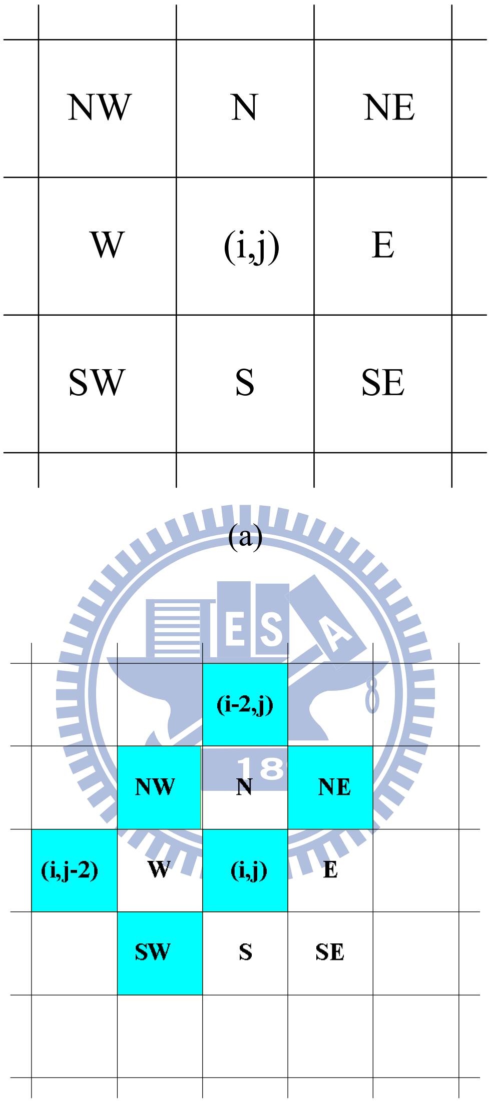
Fig. 4.1 The membership function corresponds to *large*.

Second, we consider a 3×3 neighborhood around the central pixel $I(i,j)$. Each of the eight neighbors of $I(i,j)$ corresponds to one direction {North West (*NW*), North (*N*), North East (*NE*), East (*E*), South East (*SE*), South (*S*), South West (*SW*), West (*W*)} that is displayed in Fig. 4.2(a). We use the concept of the fuzzy gradient value which contains the basic gradient value and the related gradient value. The basic gradient value is donated as $\nabla_D I(i,j)$ of pixel position (i,j) in direction set *D* ($D \in \{NW, N, NE, W, E, SE, S, SW\}$).

For example,

$$\nabla_{NW} I(i,j) = I(i-1, j-1) - I(i,j) \tag{4.11}$$

$$\nabla_{N} I(i,j) = I(i, j-1) - I(i,j) \tag{4.12}$$



(a)

(b)

Fig. 4.2 (a) The neighborhood around the central pixel $I(i,j)$. (b) Pixel indicated in gray are used to compute the fuzzy gradient value of pixel $I(i,j)$ for *NW* direction.

TABLE 4.1

The fuzzy gradient in each direction

D	Basic gradient	Related gradient	Correspond direction
NW	$\nabla_{NW} I(i,j)$	$\nabla_{NW} I(i+1, j-1)$, $\nabla_{NW} I(i-1, j+1)$	$SW-NE\,(45^{\circ})$
N	$\nabla_{N} I(i,j)$	$\nabla_{N} I(i, j-1)$, $\nabla_{N} I(i, j+1)$	$W-E\,(0^{\circ})$
NE	$\nabla_{NE} I(i,j)$	$\nabla_{NE} I(i-1, j-1)$, $\nabla_{NE} I(i+1, j+1)$	$SE-NW\,(135^{\circ})$
W	$\nabla_{W} I(i,j)$	$\nabla_{W} I(i-1, j)$, $\nabla_{W} I(i+1, j)$	$S-N\,(90^{\circ})$
E	$\nabla_{E} I(i,j)$	$\nabla_{E} I(i-1, j)$, $\nabla_{E} I(i+1, j)$	$S-N\,(90^{\circ})$
SW	$\nabla_{SW} I(i,j)$	$\nabla_{SW} I(i-1, j-1)$, $\nabla_{SW} I(i+1, j+1)$	$SE-NW\,(135^{\circ})$
S	$\nabla_{S} I(i,j)$	$\nabla_{S} I(i, j-1)$, $\nabla_{S} I(i, j+1)$	$W-E\,(0^{\circ})$
SE	$\nabla_{SE} I(i,j)$	$\nabla_{SE} I(i-1, j+1)$, $\nabla_{SE} I(i+1, j-1)$	$SW-NE\,(45^{\circ})$

Next, we also calculate the related gradient value which corresponds to each of eight directions. For example, Fig. 4.2(b) shows the related gradient of the *NW* direction and it can be expressed as

$$\nabla'_{NW} I(i,j) = \nabla_{NW} I(i+1, j-1) = I(i, j-2) - I(i+1, j-1) \tag{4.13}$$

$$\nabla''_{NW} I(i,j) = \nabla_{NW} I(i-1, j+1) = I(i-2, j) - I(i-1, j+1) \tag{4.14}$$

In Table 4.1, we show a detail of the eight directions in the column 1, the basic gradient corresponds to each direction in column 2, the two related gradients correspond to each direction in column 3, and the correspond perpendicular direction in column 4. Actually, in the 3×3 window, the direction of the gradient only belong to $W-E$ direction (0°), $SW-NE$ direction (45°), $S-N$ direction (90°), and $SE-NW$ direction (135°). Thus, we can only compute the fuzzy gradient for the direction set ED where $ED \in \{NW, N, NE, W\}$ that contains all the orientations in the 3×3 window. For example, computing the fuzzy gradient for the *NW* direction and *SE* direction are both equivalent to computing the gradient for the $SW-NE$ direction

(45°).

For each direction of the direction set *ED*, we calculate the fuzzy derivative donated as $\gamma_P(i,j)$ in each pixel (i,j) for direction *P*, where $P \in ED$. This is realized by the following fuzzy rule 3.

Fuzzy Rules 3.

IF $|\nabla_P I(i,j)|$ is *large* **AND** $|\nabla'_P I(i,j)|$ is *large* **AND** $|\nabla''_P I(i,j)|$ is *large*

OR $|\nabla_P I(i,j)|$ is *large* **AND** $|\nabla'_P I(i,j)|$ is **not** *large* **AND** $|\nabla''_P I(i,j)|$ is *large*

OR $|\nabla_P I(i,j)|$ is *large* **AND** $|\nabla'_P I(i,j)|$ is *large* **AND** $|\nabla''_P I(i,j)|$ is **not** *large*

THEN $\gamma_P(i,j)$ is *large* in direction *P*.

The **AND** operator (**OR** operator) can be the minimum (maximum) that are the well-known triangular norms (together with their dual co-norms) in the fuzzy logic. For the **not** operator, we use the standard negator $N(x) = 1 - x$ with $x \in [0,1]$. The *large* is the fuzzy set corresponds the membership function LARGE that was defined above. The idea of this rule is to consider an edge passing though the pixel $I(i,j)$ and its neighborhood for the direction, i.e. *SW-NE* direction (45°), not only the basic gradient value $|\nabla_P I(i,j)|$ will be large, but also the related gradient $|\nabla'_P I(i,j)|$ or $|\nabla''_P I(i,j)|$ can expect to be large. Therefore, if two out of three gradient values are small, it is safe to assume that no edge exists in the considered direction.

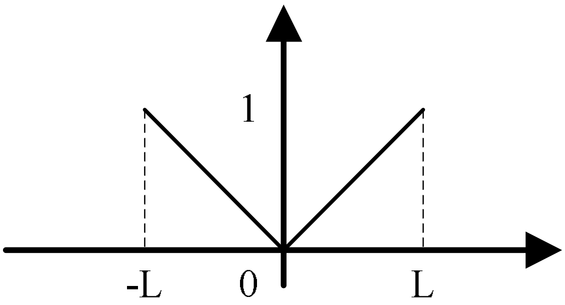


Fig. 4.3 The membership function corresponds to *absolute value.*

Next, we again use a rule for each direction. The idea behind the rule is that if a pixel is assumed to be large in rule 3, then it probably can be consider as an edge for direction *P*, and the derivative value will be used to estimate the gradient direction of this pixel. Thus, we use the following rule 4 to compute $D^{P}_{gradient}$.

Fuzzy Rules 4.

IF $\gamma_P(i,j)$ is *large* **AND** $\nabla_P I(i,j)$ is *absolute value*

THEN $D^{P}_{gradient}$ is *absolute value* in direction *P*.

Where the fuzzy set *absolute value* corresponds to the membership function which is shown in Fig. 4.3 and the parameter $L = 255$ is used in the experiment. The **AND** operator is also the minimum in the fuzzy logic.

The final step in the computation of the fuzzy gradient is the defuzzification. We are interested in obtaining the direction that has maximum value of $D^{P}_{gradient}$. P^*, which is estimated to be the gradient direction in pixel (i,j), is determined by

$$P^* = \arg\max_{P \in ED} D^{P}_{gradient} I(i,j) \tag{4.15}$$

Finally, we rotate the window with the angle that corresponds to P^*. For using the adjustable window, the VMD method will be more robust for detecting the edge response. The experiment will be discussed in Chapter 4.4.

4.3 Automatic Threshold Selection

To automatically obtain the best threshold that is adaptive to the image contents, we propose a new method for hysteresis thresholding method combining the merits of Yitzhaky and Peli [19] and Medina *et al.* [20] methods. Yitzhaky and Peli can find the best thresholds within a set of possible values, but the performance will depend on the set of possible values chosen. On the other hand Medina *et al.* method is similar to Yitzhaky and Peli, but the performance will depend on one's choice of the subset and the overset. In the following, we will introduce how to apply the thresholding methods to VMD method.

4.3.1 Determine Parameter Set

Fig. 4.4 shows the symbolic graphic of the choice of parameter set. For an image *I*, let E_I be the unknown true edge points set of the image *I* with the condition $A_I \subseteq E_I \subseteq B_I$ where A_I and B_I are the subset and the overset of the image *I*. For a possible hysteresis thresholds set *C*, for example

$$C = \left\{ (t_{low}, t_{high}) \;\middle|\; t_{low}, t_{high} \in [0,\ 1],\ t_{low} < t_{high} \right\}$$

we want to find the parameter set *T* in the region between the subset A_I and the overset B_I. Thus, we will get the best edge map $E_{t_{low},t_{high}}$ determined by hysteresis thresholds t_{low} and t_{high} with $(t_{low}, t_{high}) \in T$ in the next section.

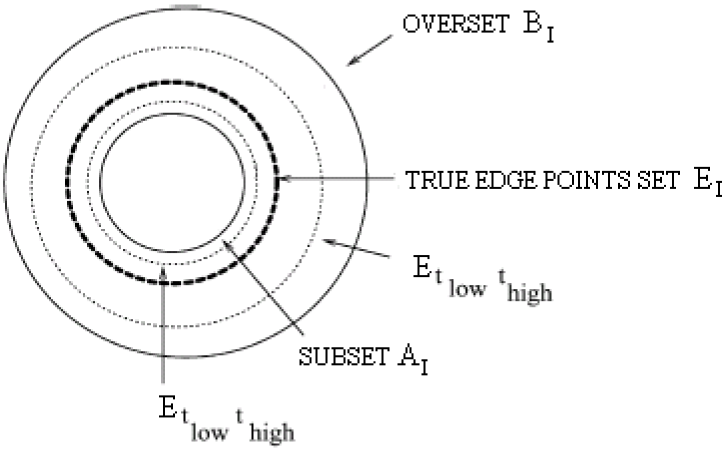


Fig. 4.4 The symbolic graphic of the choice of parameter set [20].

Considering the feature image histogram is usually unimodal, we use the Otsu method [13] and the Rosin method [14] to determine the subset A_I and the overset B_I. The Otsu method is not very sensitive on unimodal histograms and performs rigorously on detecting edge points for edge detection, but edge pixels detected by the Otsu method have a high probability of being true edge points. Thus, the edge map E_{Otsu} can be utilized as the subset of the image. The Rosin method is very sensitive on unimodal histograms and it can be noisy for edge detection, but the Rosin method can usually detect the true edge pixels together with many fakery ones. Thus, the edge map $E_{Ro\sin}$ can be employed as the overset of the image. If the conditions $E_{t_{low},t_{high}} \subseteq E_{Ro\sin}$ and $E_{Ostu} \subseteq E_{t_{low},t_{high}}$ hold, then the following expression can easily be proved

$$FP(E_{t_{low},t_{high}}, E_{Ro\sin}) = 0$$

$$FN(E_{t_{low},t_{high}}, E_{Ostu}) = 0 \qquad (4.16)$$

where *FP* and *FN* are False Positive and False Negative in ROC analysis [21]. Here, *FP* indicates that the points were decided as edges in $E_{t_{low},t_{high}}$ and coincide with non-edge points in $E_{Ro\sin}$. FN indicates that the points were decided as non-edges in $E_{t_{low},t_{high}}$ and coincide with edge points in E_{Ostu}.

Actually, there is some error probability may exist between E_{Otsu} and $E_{Ro\sin}$ such that we can not find any hysteresis thresholds t_{low} and t_{high} with $(t_{low},t_{high}) \in C$ and satisfying $FP(E_{t_{low},t_{high}}, E_{Ro\sin}) = 0$ and $FN(E_{t_{low},t_{high}}, E_{Ostu}) = 0$. For this case, we can consider whether the following expressions are true or not.

$$FN(E_{t_{low},t_{high}}, E_{Ro\sin}) \neq 0$$

$$FP(E_{t_{low},t_{high}}, E_{Ostu}) \neq 0 \qquad (4.17)$$

$FN(E_{t_{low},t_{high}}, E_{Ro\sin}) \neq 0$ implies that there are negatives in the edge map $E_{t_{low},t_{high}}$ that are positives in the set $E_{Ro\sin}$, and $FP(E_{t_{low},t_{high}}, E_{Ostu}) \neq 0$ implies that there are positives in the edge map $E_{t_{low},t_{high}}$ that are negatives in the set E_{Otsu}. Thus, let C_H and C_K be the sets that define as

$$C_H = \left\{ (t_{low},t_{high}) \;\middle|\; t_{low},t_{high} \in C \right\} \text{ with the condition (4.16)}$$

$$C_K = \left\{ (t_{low},t_{high}) \;\middle|\; t_{low},t_{high} \in C \right\} \text{ with the condition (4.17)}$$

The parameter set T can be determined by

```
If $C_H \neq \Phi$
    Then the parameter set $T = C_H$
Else if $C_K \neq \Phi$
    Then the parameter set $T = C_K$
Else
    Then the parameter set $T = C$
```

4.3.2 The Best Threshold Selection

Here, we will find the best hysteresis thresholds within the parameter set T determined above. For a parameter set T, in the first, we can construct an image map PGT (potential ground truth) using all edge maps $E_{t_{low},t_{high}}$ that determined by the hysteresis thresholds with $(t_{low},t_{high}) \in T$. Defining an edge pixel as “1” and a non-edge pixel as “0” for all edge maps $E_{t_{low},t_{high}}$, each pixel of the PGT image will get a value from the sum of the correspond pixel of all edge maps $E_{t_{low},t_{high}}$. For example, consider a pixel $p \in I$, there are q different edge maps $E_{t_{low},t_{high}}$ with different hysteresis thresholds t_{low} and t_{high}, where $(t_{low},t_{high}) \in T$, detect the pixel p as an edge pixel, then we mark the value of q to this pixel in the PGT image. It means that there are q hysteresis threshold sets support the pixel p to be an edge.

Second, we will compute the maximum value of the PGT image. If the maximum value is L, then the PGT_i can be obtained by threshold the PGT image with possible thresholds $i = 0, 1, ..., L-1$. For each PGT_i edge map, we calculate TP, TN, FP and FN that indicate True Positive, True Negative, False Positive and False Negative from comparing with each $E_{t_{low},t_{high}}$, and the average of all the probabilities resulting are computed by

$$\overline{TP_{PGT_i}} = \frac{1}{N} \sum_{(t_{low},t_{high} \in T)} TP_{PGT_i, E_{t_{low},t_{high}}} \quad \text{with } i = 0,\ 1,\ ...,\ L-1 \tag{4.18}$$

$$\overline{TN_{PGT_i}} = \frac{1}{N} \sum_{(t_{low},t_{high} \in T)} TN_{PGT_i, E_{t_{low},t_{high}}} \quad \text{with } i = 0,\ 1,\ ...,\ L-1 \tag{4.19}$$

$$\overline{FP_{PGT_i}} = \frac{1}{N} \sum_{(t_{low}, t_{high} \in T)} FP_{PGT_i, E_{t_{low}, t_{high}}} \quad \text{with} \quad i = 0,\ 1,\ \ldots,\ L-1 \tag{4.20}$$

$$\overline{FN_{PGT_i}} = \frac{1}{N} \sum_{(t_{low}, t_{high} \in T)} FN_{PGT_i, E_{t_{low}, t_{high}}} \quad \text{with} \quad i = 0,\ 1,\ \ldots,\ L-1 \tag{4.21}$$

where $TP_{PGT_i, E_{t_{low}, t_{high}}}$ indicates the pixels are edges in the PGT_i and coincide with edges in the $E_{t_{low}, t_{high}}$. $TN_{PGT_i, E_{t_{low}, t_{high}}}$ indicates the pixels are non-edges in the PGT_i and coincide with non-edges in the $E_{t_{low}, t_{high}}$. $FP_{PGT_i, E_{t_{low}, t_{high}}}$ indicates the pixels are edges in the PGT_i but detected as non-edges in the $E_{t_{low}, t_{high}}$. $FN_{PGT_i, E_{t_{low}, t_{high}}}$ indicates the pixels are non-edges in the PGT_i but detected as edges in the $E_{t_{low}, t_{high}}$. N represents the cardinality of the parameter set T. Next, the Chi-square test of the optimal threshold can be calculated by

$$\chi^2_{PGT_i} = \frac{\overline{Sn_{PGT_i}} - Q_{PGT_i}}{1 - Q_{PGT_i}} \frac{\overline{Sp_{PGT_i}} - \left(1 - Q_{PGT_i}\right)}{Q_{PGT_i}} \tag{4.22}$$

where

$$Q_{PGT_i} = \overline{TP_{PGT_i}} + \overline{FP_{PGT_i}}$$

$$\overline{Sn_{PGT_i}} = \frac{\overline{TP_{PGT_i}}}{\overline{\overline{TP_{PGT_i}} + \overline{FN_{PGT_i}}}}$$

$$\overline{Sp_{PGT_i}} = \frac{\overline{TN_{PGT_i}}}{\overline{\overline{TN_{PGT_i}} + \overline{FP_{PGT_i}}}}$$

Finally, a higher $\chi^2_{PGT_i}$ can obtain a better threshold. The best threshold in parameter set T is correspond to the value of i that maximizes $\chi^2_{PGT_i}$. Thus, we have the following conclusion: If $k = \underset{i \in \{0, 1, \ldots, L-1\}}{\arg\max}\ \chi^2_{PGT_i}$, then the hysteresis thresholds

$\left(t^{*}_{tlow}, t^{*}_{thigh}\right) = \underset{(t_{low}, t_{high}) \in T}{\arg\max}\, \mu\left(E_{t_{low}, t_{high}}\right)$ will be the best choice, where

$$\begin{aligned}
&\mu\left(E_{t_{low}, t_{high}}\right) \\
&= \chi^{2}_{E_{tlow,thigh}, PGT_k} \\
&= \frac{Sn^{*}{}_{PGT_k, E_{tlow,thigh}} - Q^{*}{}_{PGT_k, E_{tlow,thigh}}}{1 - Q^{*}{}_{PGT_k, E_{tlow,thigh}}} \frac{Sp^{*}{}_{PGT_k, E_{tlow,thigh}} - \left(1 - Q^{*}{}_{PGT_k, E_{tlow,thigh}}\right)}{Q^{*}{}_{PGT_k, E_{tlow,thigh}}} \qquad (4.23)
\end{aligned}$$

and

$$Q^{*}{}_{PGT_k, E_{tlow,thigh}} = TP_{PGT_k, E_{tlow,thigh}} + FP_{PGT_k, E_{tlow,thigh}}$$

$$Sn^{*}{}_{PGT_k, E_{tlow,thigh}} = \frac{TP_{PGT_k, E_{tlow,thigh}}}{TP_{PGT_k, E_{tlow,thigh}} + FN_{PGT_k, E_{tlow,thigh}}}$$

$$Sp^{*}{}_{PGT_k, E_{tlow,thigh}} = \frac{TN_{PGT_k, E_{tlow,thigh}}}{TN_{PGT_k, E_{tlow,thigh}} + FP_{PGT_k, E_{tlow,thigh}}}$$

One of the drawbacks of Yitzhaky and Peli method is that the result depends on the parameter set we choose. For different parameter sets, very different results will be obtained. Thus, after the improvement of the choice of the parameter set, a more reliable result will be obtained.

4.4 Experimental Result of Comparison with Other Color Edge Detector

4.4.1 Quantitative Evaluation

The evaluations of edge detectors are usually subjective by observers [23]. Most of the objective evaluation methods assume that the specific features of images are known such as boundaries in simple synthetic images. In such cases, for the known ideal edges considered to be the ground truth (GT), the quantitative of the edge detection can be measured. In nature images, Bowyer [24] manually created specification of the edges to form a GT, and Fernandez [25] used an automatic statistical method to generate GT images. However, for a quantitative evaluation, different criteria will produce different result. The approach adopted here is to use the GT images generated from synthetic image with the widely used performance measures, FOM [22] and ROC analysis [21].

First, The FOM is defined by

$$FOM = \frac{1}{\max\{I_D, I_I\}} \sum_{i=1}^{I_D} \frac{1}{1+\alpha\left(d_i\right)^2} \times 100\% \tag{4.24}$$

where I_D and I_I are the number of detected and number of ideal edge points respectively, α (>0) is a calibration constant, and d_i is the edge deviation for the i-th detected edge pixel. In all cases $0 < \mathrm{FOM} \leq 1$; for a perfect match between the detected and the ideal edges $\mathrm{FOM} = 1$ whereas the detected edges deviate more and more from the ideal ones FOM goes to zero. The scaling constant $\alpha = 0.2$ proposed in [11] has been adopted.

Next, we would like to introduce the True Positive Rate (TPR), True Negative Rate (TNR) and Normalized Accuracy (NACC) of ROC analysis. The TPR is defined by

$$TPR = \frac{TP}{TP+FN} \times 100\% \tag{4.25}$$

where *TP* represents the number of pixels which are detected as an edge pixel and belong to an ideal edge pixel, and *FN* represents the number of pixels which are detected as an edge pixel but belong to an ideal non-edge pixel. On the other hand, the TNR is defined as

$$TNR = \frac{TN}{TN + FP} \times 100\% \tag{4.26}$$

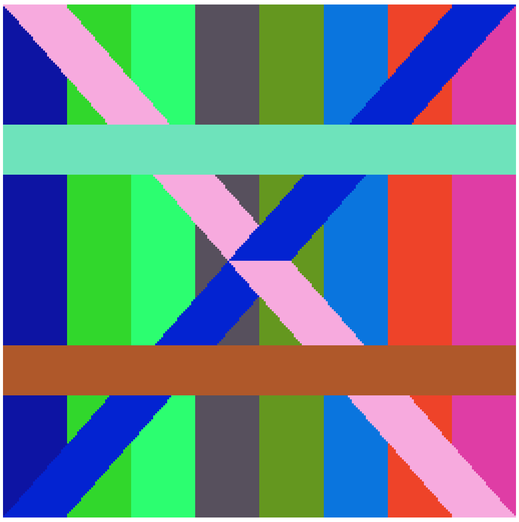
where *TN* represents the number of pixels which are detected as an non-edge pixel and belong to an ideal non-edge pixel, and *FP* represents the number of pixels which are detected as an non-edge pixel but belong to an ideal edge pixel. Finally, we calculate the normalized accuracy (NACC) by

$$NACC = \frac{TPR + TNR}{2} \times 100\% \tag{4.27}$$

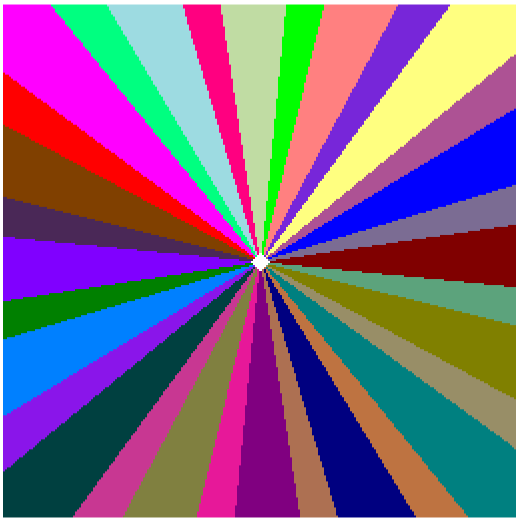
NACC = 100% corresponds to a perfect match between the ideal edge and detected edge points and as the deviation of the edge points increase, the NACC approaches to zero percentage.

4.4.2 Quantitative Evaluation in Synthetic Color Images

Canny [3] presented the very popular aspects that good edge detection must not miss the true edge nor detect non-edge points as the edge points and produce thin and continuous lines. For these criteria, we also use two kinds of 256×256 synthetic color images which are shown in Figs. 4.5(a) – (b) for quantitative evaluation of the color edge detectors. The performances of our automatic color edge detection techniques are compared to those by the compass operator of Ruzon and Tomasi [9], Canny edge detector [3], RCMG detector [11], and MVD edge detector [6].



(a)

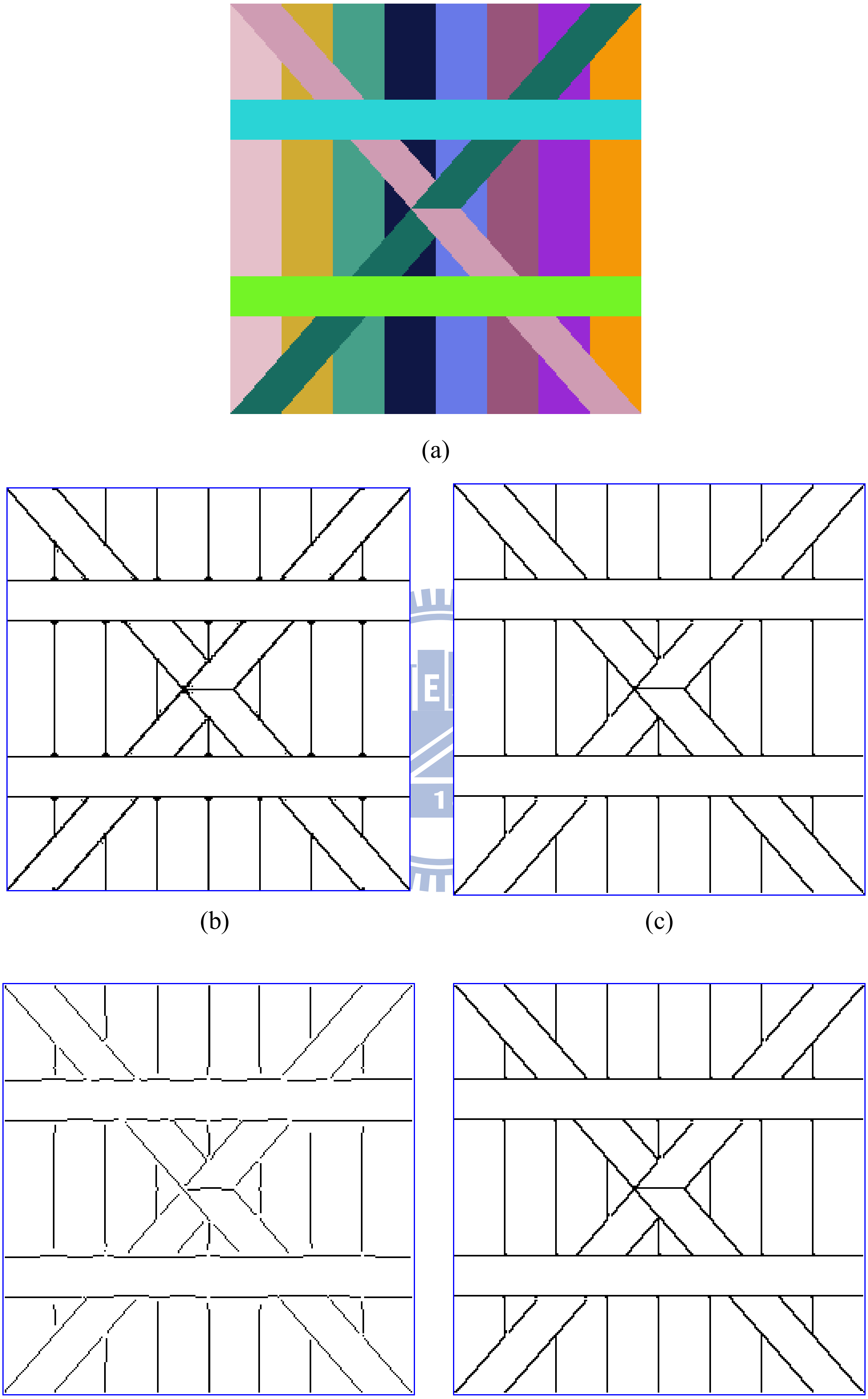


(b)

Fig. 4.5 (a) One kind of original image. (b) One kind of original image.

Fig. 4.6 show the edge detection results of 256×256 synthetic images which is Fig. 4.5(a) for comparison. For the Fig. 4.6(b), using the parameter $\sigma = 0.94$, adjust the Medina *et al.* thresolding method for the compass operator with NMS. We can see that the result in Figs. 4.6(b) detect much noise in the regions near the corners. For MVD detector, we use the parameter $k = 2$, $l = 4$, and thresholding by Medina *et al.* method with thinning process as shown in Fig 4.6(c). To apply Canny detector to color images, a method named Color Canny individually use Canny detector to detect edges for three dimensions in the color space, and determine the edge result by the majority vote fusion rule.

In the edge result detected by Color Canny detector, which is shown in Fig. 4.6(d), the continuity of the edges performs worse than the others, especially in the corners. Fig. 4.8(e) shows the result detected by RCMG with the parameter $s=1$ in the 3×3 window. Fig. 4.6(f) shows the result with our method, and it detects less noise and produce continuous lines for edge detection. More results are shown in Figs. $4.7-4.9$. In Fig. 4.7(b) and Fig. 4.9(b), some ideal edges are missed by the compass operator thresholding by Medina *et al.* method.



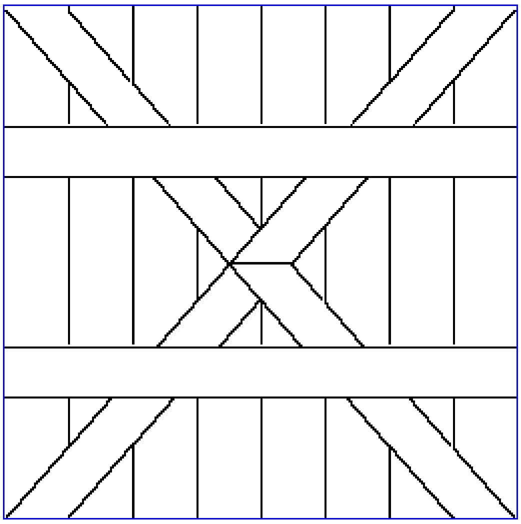
(a)

(b)

(c)

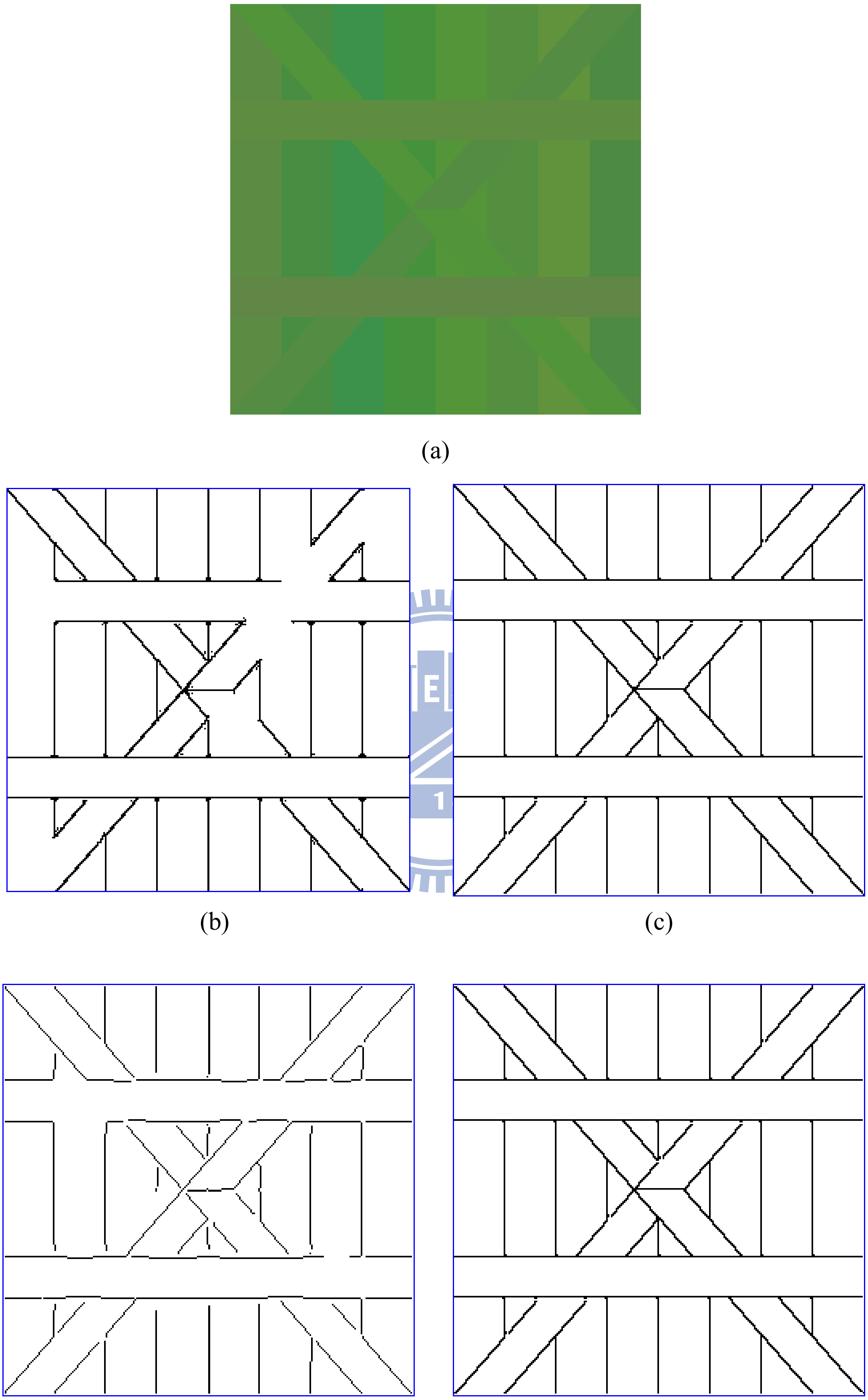
(d)

(e)



(f)

Fig. 4.6 Edge detection results of the 256×256 synthetic image Sample 1 detected by different color edge detectors. (a) Original image. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector.



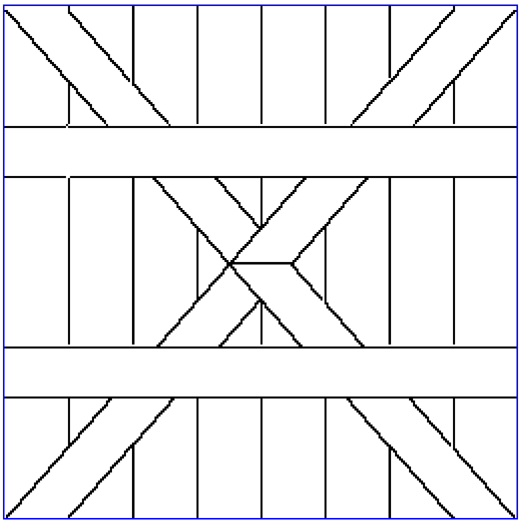
(a)

(b)

(c)

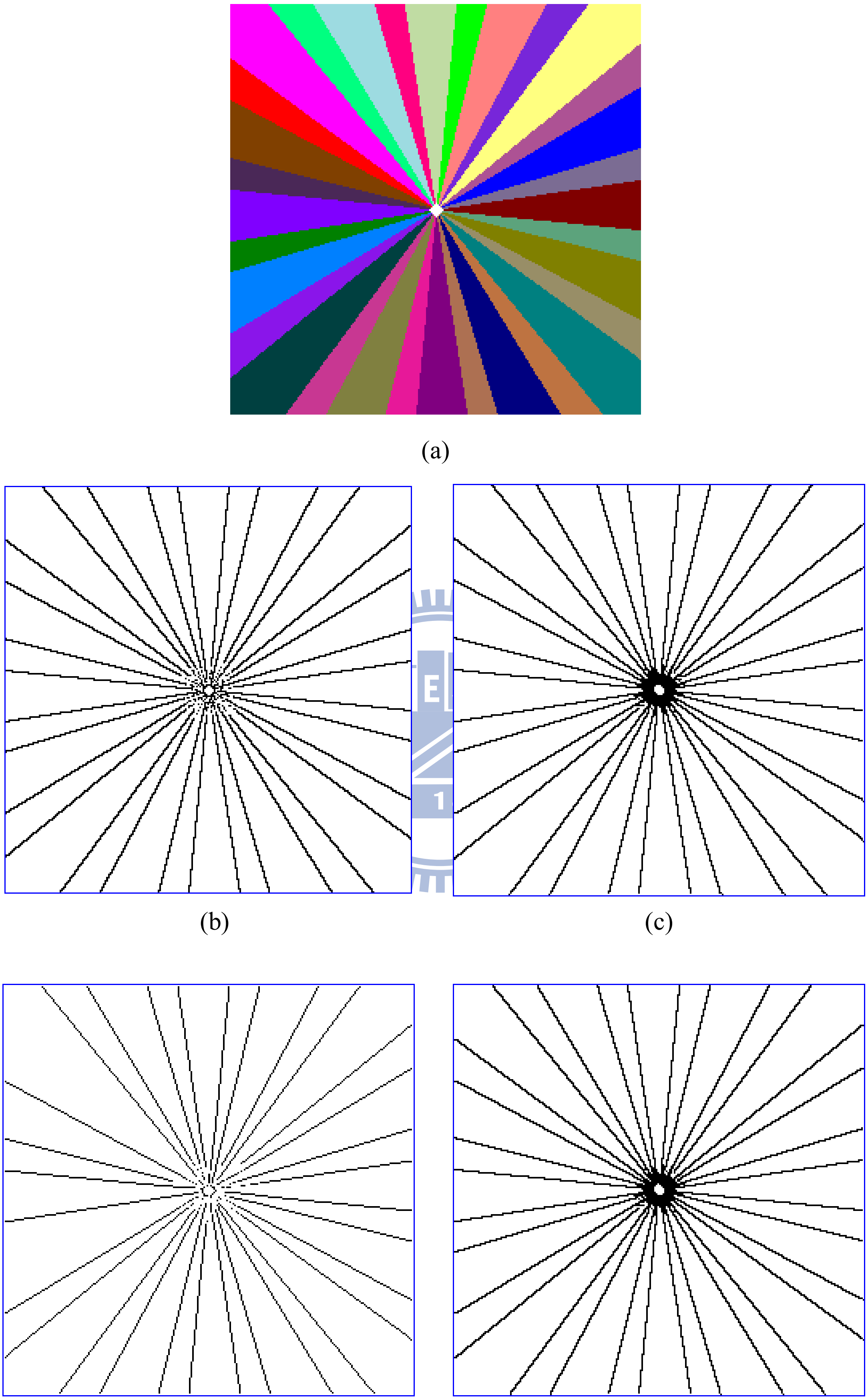
(d)

(e)



(f)

Fig. 4.7 Edge detection results of the 256×256 synthetic image Sample 2 detected by different color edge detectors. (a) Original image. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector.



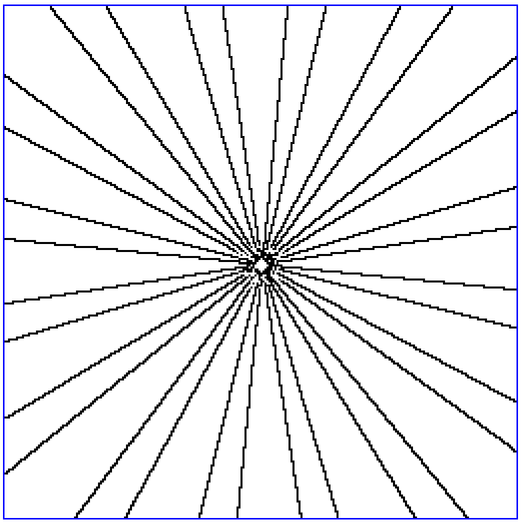
(a)

(b)

(c)

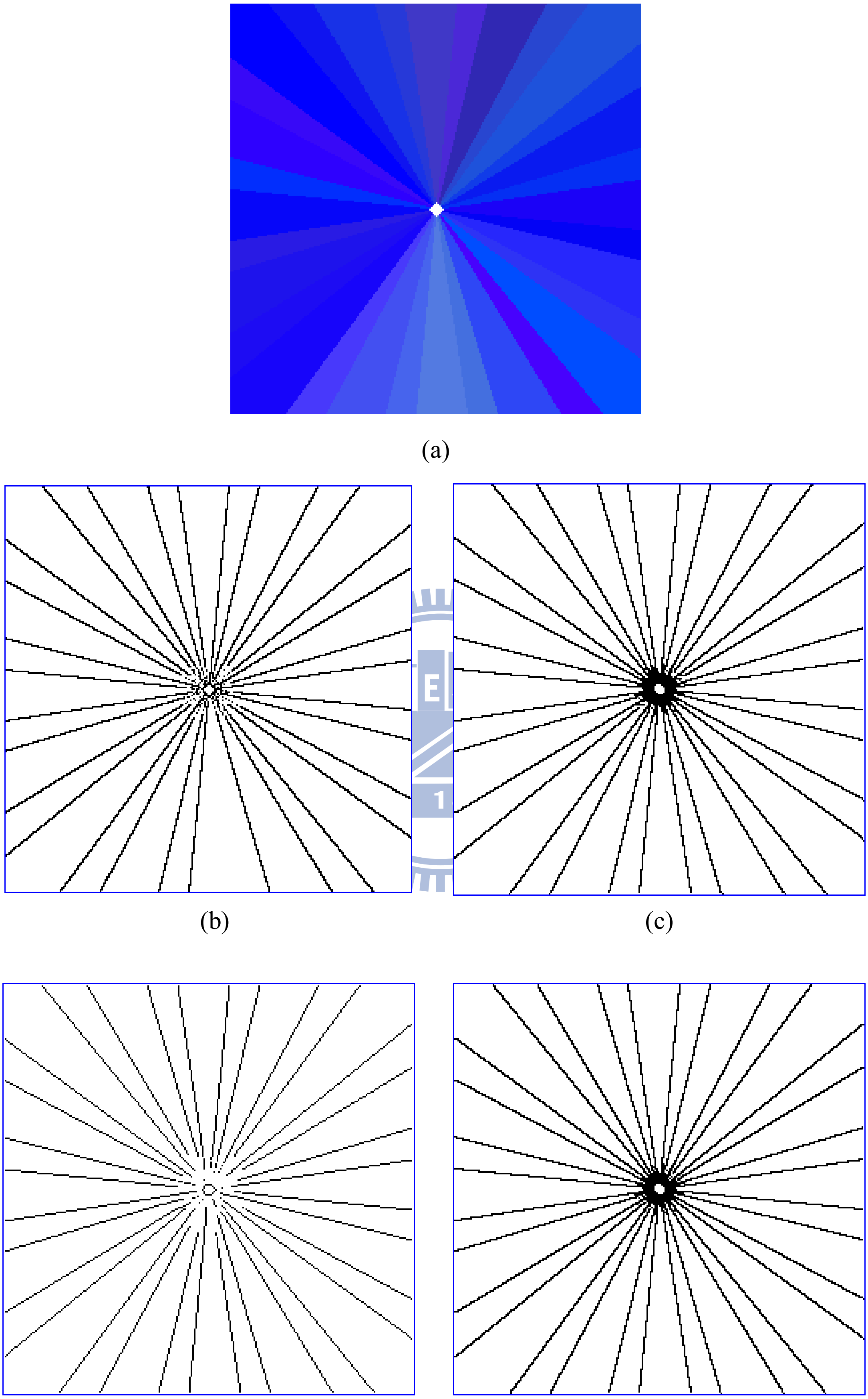
(d)

(e)



(f)

Fig. 4.8 Edge detection results of the 256×256 synthetic image Sample 3 detected by different color edge detectors. (a) Original image. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector.



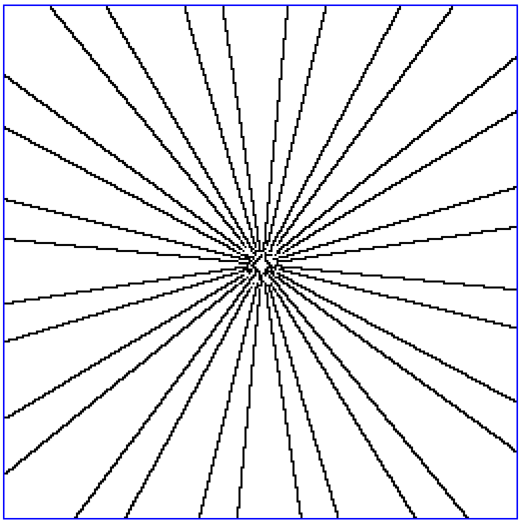
(a)

(b)

(c)

(d)

(e)



(f)

Fig. 4.9 Edge detection results of the 256×256 synthetic image Sample 4 detected by different color edge detectors. (a) Original image. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector.

(A) Quantitative Performance Comparison

Tables 4.2, 4.3 and 4.4 show the average performances of the thirteen 256×256, eleven 256×256 synthetic images, and the total images for the compared detector. The order of the column 1 are the compass operator with NMS and Medina *et al.* method, MVD detector with thinning process and Medina *et al.* method, Color Canny detector, RCMG detector with thinning process and Medina *et al.* method, and our automatic color edge detector. The column 2 to column 5 represents the quantitative evaluations of FOM, TPR, TNR, and NACC in percentage, respectively. For the criteria, a detector, which can detect less erroneous, thin, and continuous edges, will get high values of the FOM and NACC.

TABLE 4.2

The average evaluation results of the thirteen 256×256 synthetic color images detected by the following detectors

Method	FOM (%)	TPR (%)	TNR (%)	NACC (%)
Compass with NMS and Medina *et al.* method	98.14_4	92.99	99.87	96.43_4
MVD with thinning and Medina *et al.* method	99.70_2	98.54	99.91	99.22_3
Color Canny with manual thresholding	86.56_5	68.48	98.29	83.36_5
RCMG with thinning and Medina *et al.* method	99.79_1	98.98	99.94	99.46_2
Our method	99.25_3	99.91	99.95	99.93_1

TABLE 4.3

The average evaluation results of the eleven 256x256 synthetic color images detected by the following detectors

Method	FOM (%)	TPR (%)	TNR (%)	NACC (%)
Compass with NMS and Medina *et al.* method	96.33_4	99.64	99.62	99.63_2
MVD with thinning and Medina *et al.* method	99.61_2	97.65	99.93	98.79_4
Color Canny with manual thresholding	88.82_5	96.29	99.10	97.70_5
RCMG with thinning and Medina *et al.* method	99.65_1	97.92	99.91	98.92_3
Our method	98.36_3	99.73	99.82	99.78_1

TABLE 4.4

The average evaluation results of the total synthetic color images detected by the following detectors

Method	FOM (%)	TPR (%)	TNR (%)	NACC (%)
Compass with NMS and Medina *et al.* method	97.24_4	96.31	99.74	98.03_4
MVD with thinning and Medina *et al.* method	99.66_2	98.10	99.92	99.01_3
Color Canny with manual thresholding	87.69_5	82.39	98.70	90.53_5
RCMG with thinning and Medina *et al.* method	99.72_1	98.45	99.93	99.19_2
Our method	98.81_3	99.82	99.89	99.85_1

A noisy edge map may be good performance in the FOM evaluation because of the FOM only considers the accuracy of edge points and uses a scaling constant α for the penalty between smeared and offset edges. In other hand, the NACC calculates not only the accuracy of edge points but also the accuracy of non-edge points and strictly forbids the deviation between ideal and detected edge (non-edge) points. Therefore, although both MVD and RCMG with thinning and Medina *et al.* method are better than our method for the FOM evaluation, TPR and TNR shows the fact that they produce more smeared edge points and misses more ideal edge points than our method. Indeed, the NACC evaluation supply more reliable results by using the TPR and TNR, and our method is the best one in the NACC evaluation.

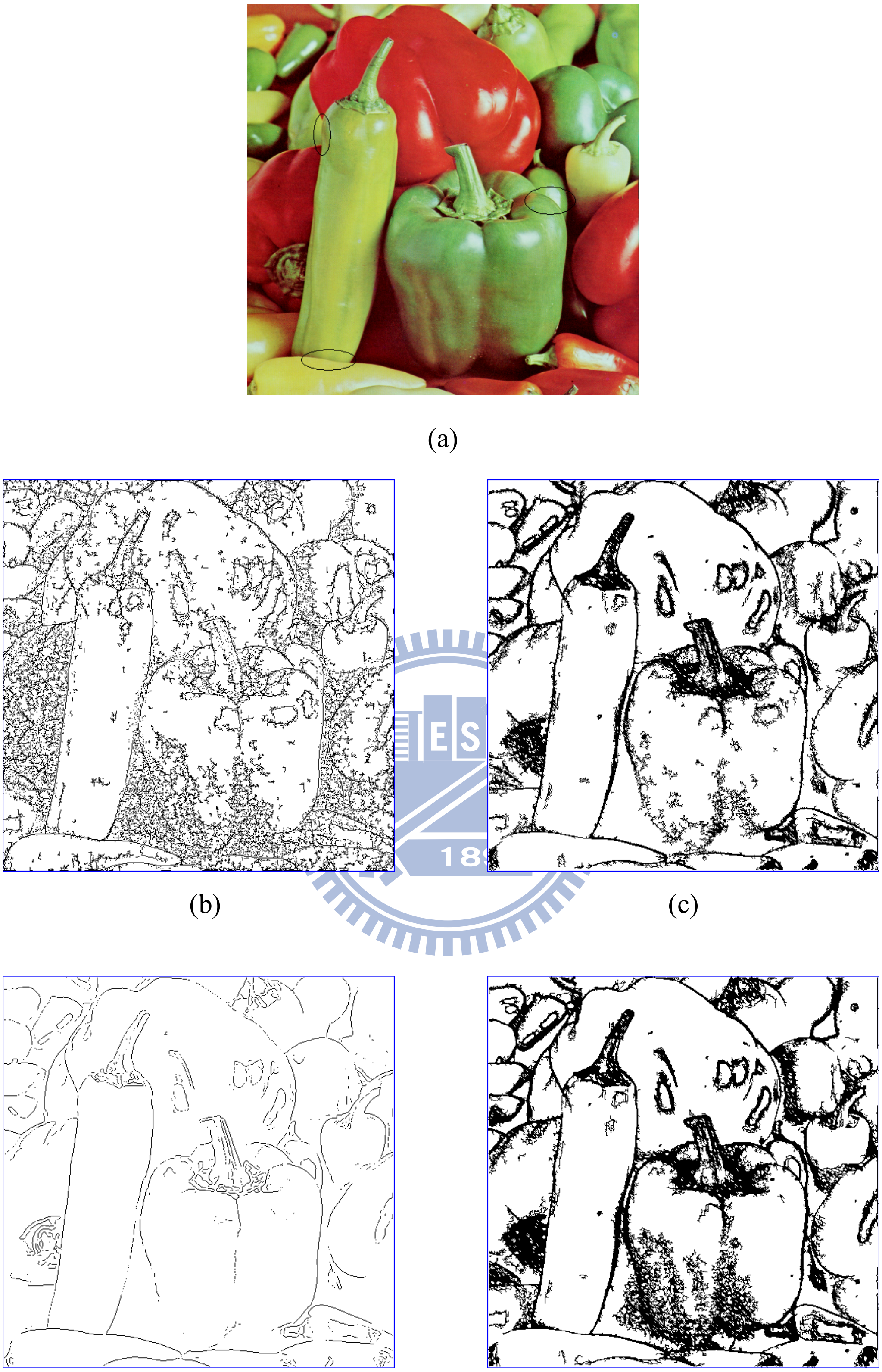
4.4.3 Comparison of Nature Color Images

In this section, we will compare our method in nature color images with other detectors mentioned above. Unlike the synthetic images, we can not use the FOM evaluation or ROC analysis to provide the absolute quality measures when GT images in real world images are both difficultly and subjectively chosen, but we can provide the information for relatively robustness and reliability. Several examples are shown to show the effectiveness of various methods.

(A) Example 1, “papers” image

Figs. 4.10 show the edge detection results of the “Peppers” image. For the Fig. 4.10(b), using the parameter $\sigma = 0.94$, we adjust the Medina *et al.* thresolding method for the compass operator with NMS. For MVD detector, we use the parameter $k = 2$, $l = 4$, and thresholding by Medina *et al.* method with thinning process as shown in Fig 4.10(c). Figs. 4.10(d)$-$(f) show the result of Color Canny, RCMG and our method, respectively.

The compass operator with NMS and thresholding by Medina *et al.* method produces very noisy edge as shown in Fig. 4.10(b). In Fig 4.10(c), MVD detects more true edges but less noise although it provides very thick edge responses. From Figs. 4.10(d)$-$(f), Color Canny and our method not only provide thinner and less noisy edges but also obtain the boundaries of the three difficult edges marked.



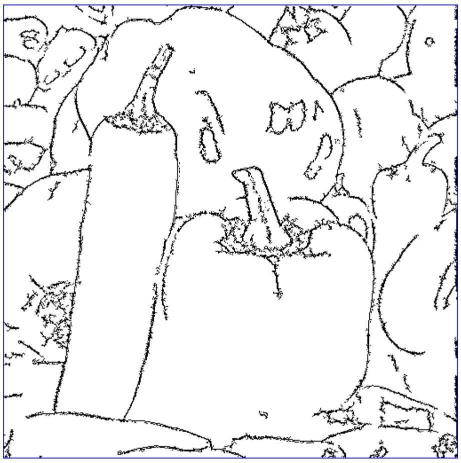
(a)

(b)

(c)

(d)

(e)



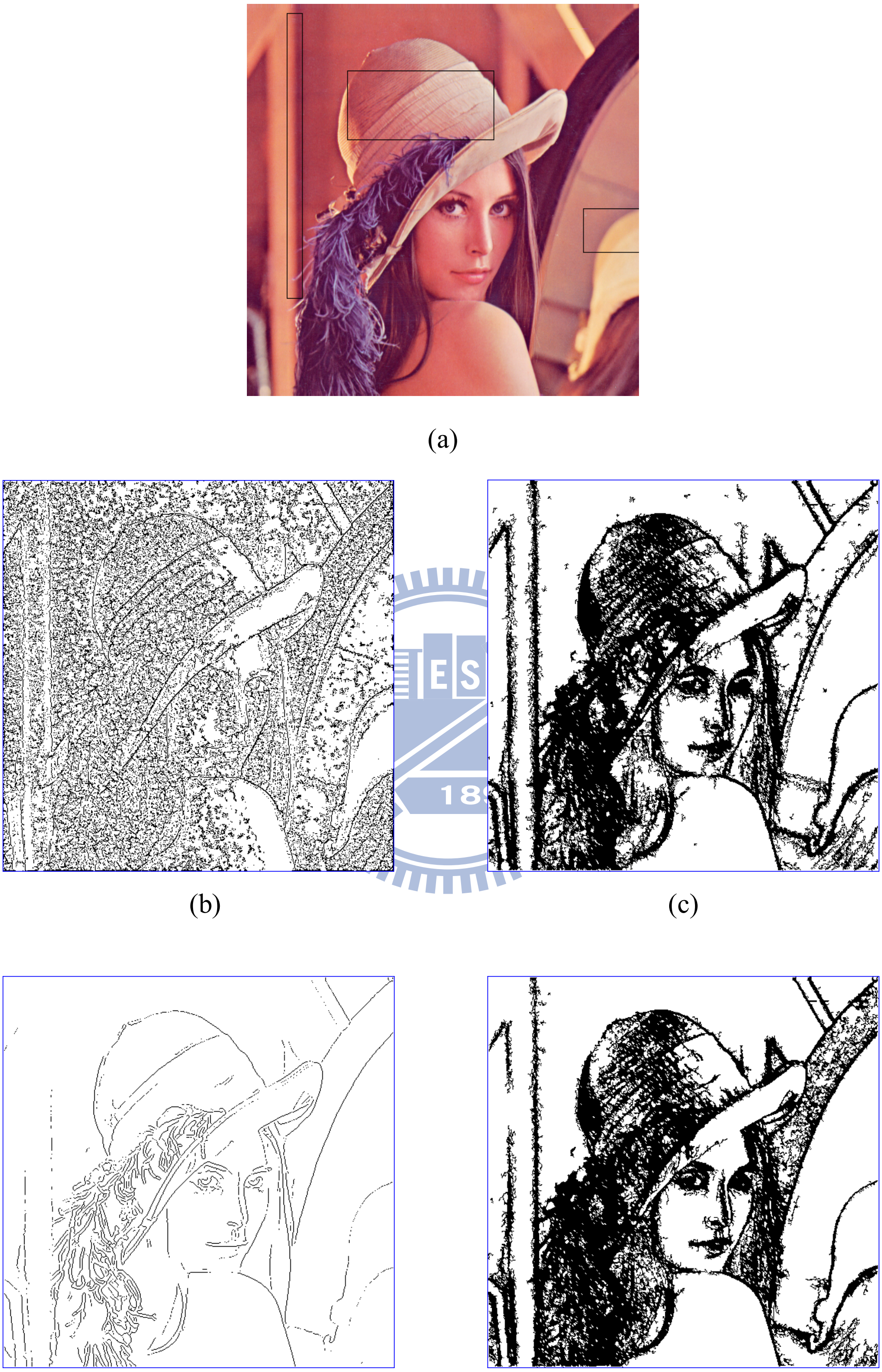
(f)

Fig. 4.10 Edge detection results of the "Peppers" image detected by different color edge detectors. (a) Original image. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector.

(B) Example 2, "Lena" image

Figs. 4.11 show the edge detection results of the "Lena" image. For the Fig. 4.11(b), using the parameter $\sigma = 0.94$, we adjust the Medina *et al.* thresolding method for the compass operator with NMS. For MVD detector, we use the parameter $k = 2$, $l = 4$, and thresholding by Medina *et al.* method with thinning process as shown in Fig 4.11(c). Figs. 4.11(d) – (f) show the result of Color Canny, RCMG and our method, respectively.

We are interesting in comparing the marked rectangle regions. Figs. 4.11(b) and (c) detect more edge in these regions, but they also provide too much noise. In the regions of the middle and right rectangles, the results of Figs. 4.11(d) and (f) detected by Color Canny and our method are provide thinner and less noisy edges.



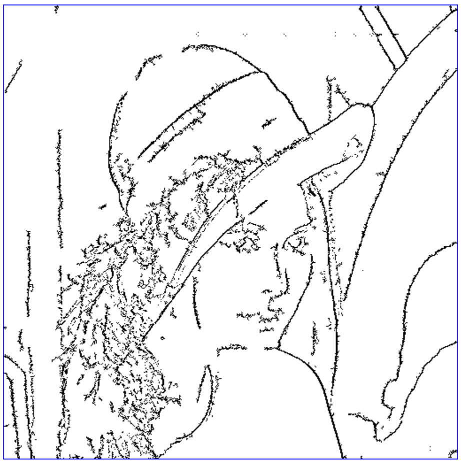
(a)

(b)

(c)

(d)

(e)



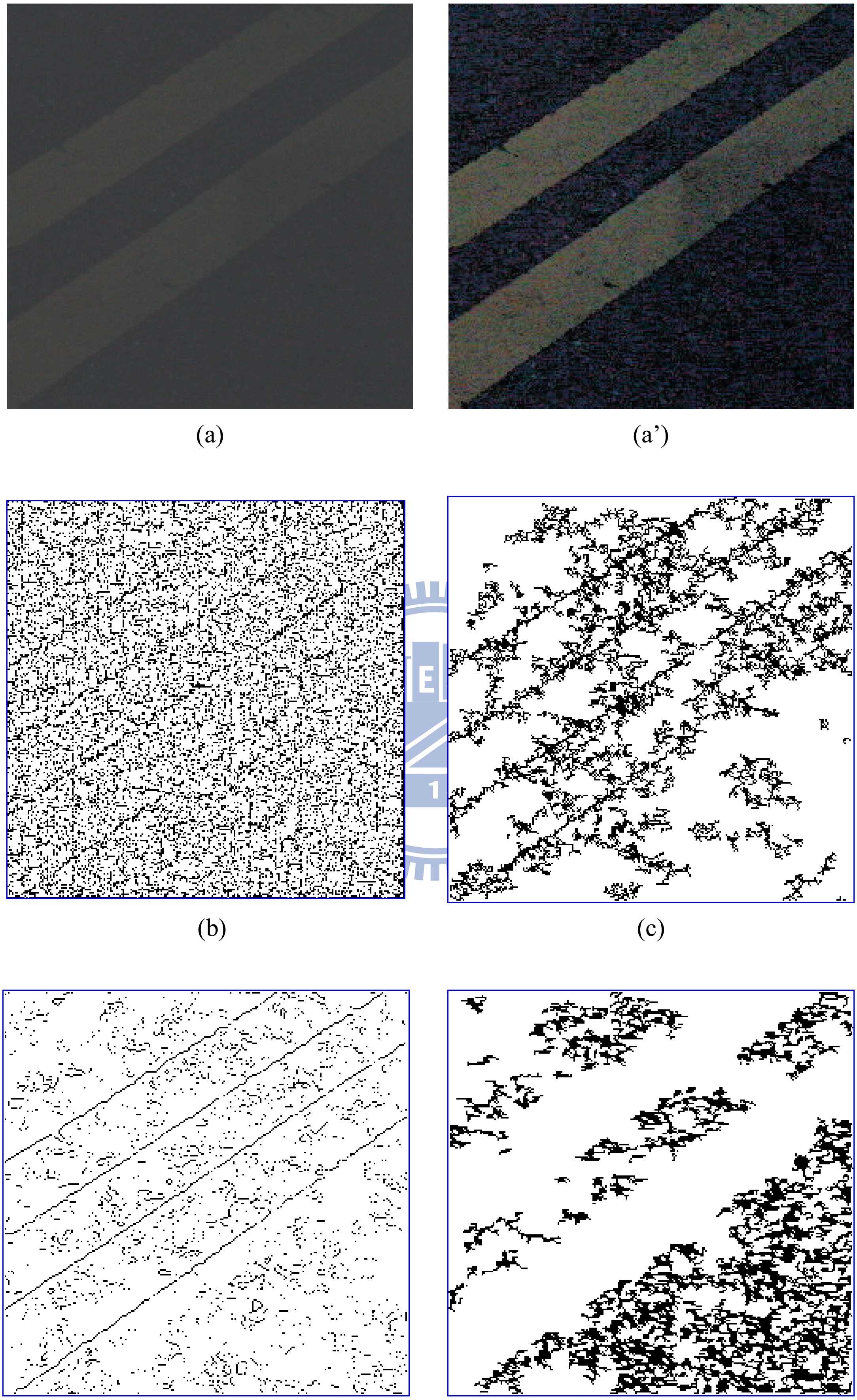
(f)

Fig. 4.11 Edge detection results of the "Lena" image detected by different color edge detectors. (a) Original image. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector.

(C) Example 3, "Road" image

Fig. 4.12 shows the edge detection results of the low illumination image, in which the road has double yellow lines.

We can see from Fig. 4.12(b) that the compass result detects much noisy image. Although Fig. 4.12(c) is cleaner than Figs. 4.12(b), it is still noisy in the output. In Fig. 4.12(e), although the road region is clean, it looks very noisy outside the road and the boundaries are incomplete. In summary, Color Canny and our method provide complete road edges and less noisy, which are better than the compass operator.



(a)

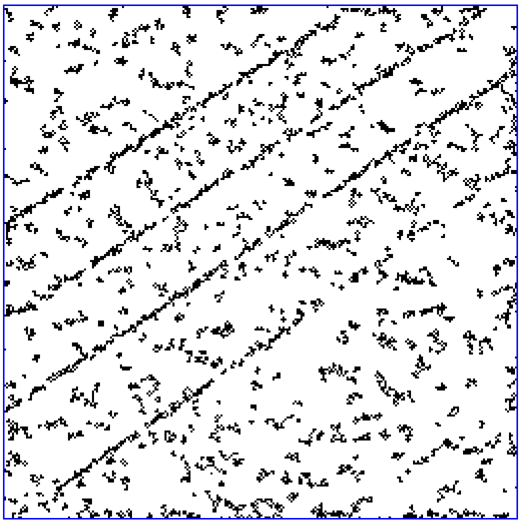
(a')

(b)

(c)

(d)

(e)



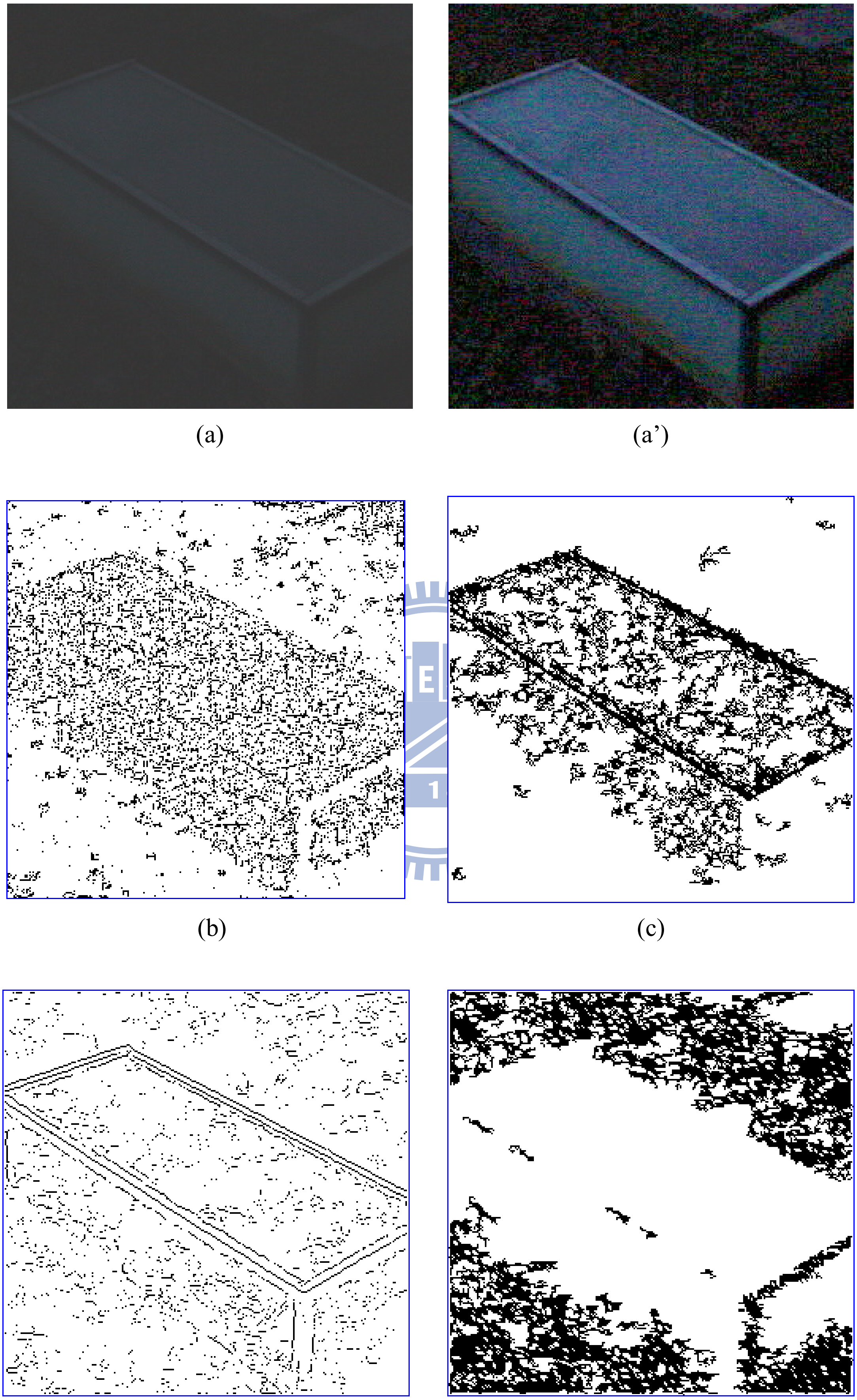
(f)

Fig. 4.12 Edge detection results of the "Road" image detected by different color edge detectors. (a) Original image. (a') gray-level histogram equalized of image (a), used for visual aide and comparison only. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector.

(D) Example 4, "Chair" image

Figs. 4.13 show the edge detection results of the low illumination image, in which there is a chair.

We are interesting in comparing the chair's rectangle region. Fig. 4.13(b) detects a little edges in the regions, but its rectangle boundaries are incomplete and noisy. Fig. 4.13(c) detects the rectangle region completely, but it is very noisy. In Fig. 4.13(e), it is too noisy outside of the rectangle region. The results of Figs. 4.13(d) and (f) detected by Color Canny and our method produce complete boundaries and less noisy edges.



(a)

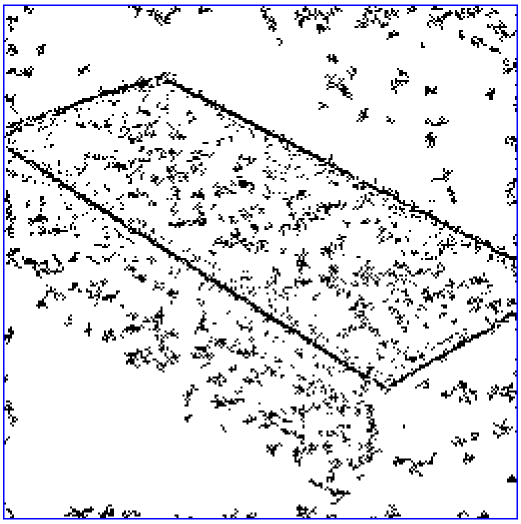
(a’)

(b)

(c)

(d)

(e)



(f)

Fig. 4.13 Edge detection results of the “Chair” image detected by different color edge detectors. (a) Original image. (a’) gray-level histogram equalized of image (a), used for visual aide and comparison only. (b) The compass operator with NMS and thresholding by Medina *et al.* method. (c) MVD with thinning process and thresholding by Medina *et al.* method. (d) Color Canny result. (e) RCMG with thinning process and thresholding by Medina *et al.* method. (f) Our automatic color edge detector.

Chapter 5 Conclusion

In this thesis, we use edge detection and fuzzy rules to find bad pixel map of a SWIR sensor. Although the bad pixel map obtains include some extra good pixels or edge points, the bad pixels can mostly detected. We use two median filters sequentially to correct them, while the extra good pixels can almost retain their original values. Furthermore, we employ two-point correction method to correct non-uniformity among pixels of SWIR sensor. In the future, more advanced NUC method will be investigated to improve SWIR sensor performance. Integration and co-design of our proposed algorithm in the ROIC design of SWIR sensor would be our future work.

To enhance the tools for bad pixel correction, we have also proposed vector order statistics and fuzzy gradient to automatic color edge detection of images. By using the fuzzy derivative estimation, the fuzzy rules are fired to consider the gradient direction of every processing pixel. Additionally, the shape of the membership function is adapted to the local variation around the processing pixel. The proposed detector improves the drawbacks of the original VMD detector because the gradient directions can be better estimated. Moreover, our thresholding method can choose a reasonable parameter set from all possible attempts and then find the best hysteresis threshold set from them.

Experimental results have shown that our automatic color edge detection techniques produce excellent edge detection accuracy in both the synthetic and real-world images. In this way, the performances of higher level image processing tasks such as bad pixel correction, segmentation and object recognition can be more powerful because of the improvement of our edge detection schemes.

References

[1] A. F. Milton, F. R. Barone, and M. R. Kruer, "Influence of nonuniformity on infrared focal plane array performance," *Opt. Eng.*, vol. 24, pp. 855–862, 1985.

[2] P. Fillon, A. Combette P. Tribolet, "Cooled IR detectors calibration analysis and optimization," *Proc. of SPIE,* vol. 5784, pp. 343-354, 2005.

[3] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.

[4] C. L. Novak and S. A. Shafer, "Color edge detection," in *Proc. of DARPA Image Understanding Workshop*, 1987, pp. 35–37.

[5] P.E. Trahanias and A.N. Venetsanopoulos, "Color edge detection using vectororder statistics," *IEEE Trans. Image Process.*, vol. 2, no. 2, pp. 259–264, Apr. 1993.

[6] P.E. Trahanias and A.N. Venetsanopoulos, "Vector order statistics operators as color edge detectors," *IEEE Trans. Syst., Man, Cybern., B*, vol. 26, no. 1, pp. 135–142, Feb. 1996.

[7] D. Van De Ville, M. Nachtegael, D. Van der Weken, E.E. Kerre, W. Philips, and I. Lemahieu, "Noise reduction by fuzzy image filtering," *IEEE Trans. Fuzzy Systems,* vol. 11, no. 4, pp.429–436, Aug. 2003.

[8] S. Schulte, V. De Witte, M. Nachtegael, D. Van der Weken, E.E. Kerre, "Fuzzy random impulse noise reduction method," *Fuzzy Sets and Systems*, vol.158, pp.270–283, February 2007.

[9] M. A. Ruzon and C. Tomasi, "Edge, junction, and corner detection using color distributions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no.11, pp. 1281–1295, Nov. 2001.

[10] J. Scharcanski and A.N. Venetsanopoulos, "Edge detection of color images

using directional operators," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 26, no. 1, pp. 135–142, Feb. 1996.

[11] A. N. Evans and X. U. Liu, "A morphological gradient approach to color edge detection," *IEEE Trans. Image Process.*, vol. 15, no. 6, pp. 1454–1463, Jun. 2006.

[12] A. Cumani, "Efficient contour extraction in color images," in *ACCV '98: Proceedings of the Third Asian Conference on Computer Vision*, London, UK, 1998, vol. 1, pp. 582–589, Springer-Verlag.

[13] N. Otsu, "A threshold selection method from grey-level histograms," *IEEE Trans. Syst. Man Cybern.*, vol. SMC-9, no. 1, pp. 62–66, Jan. 1979.

[14] P. L. Rosin, "Unimodal thresholding," *Pattern Recognit.*, vol. 34, pp.2083–2096, 2001.

[15] R. Rakesh, P. Chaudhuri, and C. A. Murthy, "Tresholding in edge detection: A statistical approach," *IEEE Trans. Image Process.*, vol. 13, no. 7, pp. 927–936, Jul. 2004.

[16] H. A. David, *Order Statistics*. New York: Wiley, 1980.

[17] V. Barnett, "The ordering of multivariate data," *J. Royal Statistical Society A*, vol. 139, Part 3, pp. 318–343, 1976.

[18] J. Astola, P. Haavisto and Y. Neuvo, "Vector median filters," *Proc. IEEE,* vol. 78, no. 4, pp. 678–689, Apr. 1987.

[19] Y. Yitzhaky and E. Peli, "A method for objective edge detection evaluation and detector parameter selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 1027–1033, Aug. 2003.

[20] R. Medina-Carnicer, A. Carmona-Poyato, R. Muñoz-Salinas, and F. J. Madrid-Cuevas, "Determining hysteresis thresholds for edge detection by combining the advantages and disadvantages of thresholding methods," *IEEE*

Trans. Image Process., vol. 19, no. 1, pp.165–173, Jan. 2010.

[21] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters,* vol. 27, no. 8, pp. 861–874, 2006.

[22] I. Abdou. And W. Pratt, “Quantitative design and evaluation of enhancement/thresholding edge detectors,” *Proc. IEEE*, vol. 67, no. 5, pp. 753–763, May. 1979.

[23] M. Heath, S. Sarkar, T. Sanocki, and K.W. Bowyer, “Comparison of Edge Detectors: A Methodology and Initial Study,” Computer Vision and Image Understanding, vol. 69, no. 1, pp. 38-54 Jan. 1998.

[24] K. Bowyer, C. Kranenburg, and S. Dougherty, “Edge Detector Evaluation Using Empirical ROC Curves,” Computer Vision and Image Understanding, vol. 84, no. 1, pp. 77-103, Oct. 2001.

[25] N. L. Fernández-García, A. Carmona-Poyato, R. Medina-Carnicer, and F. J. Madrid-Cuevas, “Automatic generation of consensus ground truth for the comparison of edge detection techniques,” *Image Vis. Comput.*, vol. 26, pp. 496–511, 2008.