# Automatic edge detection using vector distance and partial normalization

SHUHAN CHEN*
College of Automation
Chongqing University
No.174, Shazheng Street, Shapingba District, Chongqing
PEOPLE'S REPUBLIC OF CHINA
c.shuhan@gmail.com

WEIREN SHI
College of Automation
Chongqing University
No.174, Shazheng Street, Shapingba District, Chongqing
PEOPLE'S REPUBLIC OF CHINA
wrs@cqu.edu.cn

Kai Wang
College of Automation
Chongqing University
No.174, Shazheng Street, Shapingba District, Chongqing
PEOPLE'S REPUBLIC OF CHINA
akyle@163.com

*Abstract:* - This paper proposes a novel edge detection method for both gray level images and color images, and which can overcome the limitations of gradient-based edge detection methods. A vector distance between feature vector and minimum vector which determines the edge intensity is defined based on four directional summed magnitude differences in a mask, and partial normalization is applied to facilitate threshold selecting. This paper also proposes an improved approach to determine the edge direction. According to the improved edge direction, non-maxima suppression is applied to thin edges, and final edges are extracted automatically using OTSU, even in a changing environment. Extensive experimental results have demonstrated that the proposed method does well in keeping low-contrast edges, selecting threshold and processing time.

*Key-Words:* - Edge detection, Color edge detection, Vector distance, Partial normalization, Non-maxima suppression

## 1 Introduction

Edge detection has become one of the topics of most active and continuing interest because it is the basis and key issue in image processing and computer vision. There are many edge detectors proposed during the past two decades. Among all existing methods, those gradient-based detectors are the most classic and widely used due to their simplicity and efficiency, such as Sobel, Roberts, Prewitt, Laplacian, Canny [1, 2]. Nevertheless, they suffer from some practical limitations.

First, the traditional edge detection methods mostly adopt the first-order or second-order derivative to calculate gradient magnitude, a pixel belongs to the edge map only when the associated gradient magnitude is sufficiently large, so the gradient magnitude alone is insufficient to determine all the meaningful edges [3]. Based on the ambiguity caused by the underlying pixel pattern, scholars have proposed

* Corresponding author.

many improved and some new edge detection methods. The edge detector proposed by Meer and Georgescu uses a confidence measure to reduce this ambiguity, and detailed edges can be well preserved [4]. Liang and Looney propose a competitive fuzzy edge detection method (CFED for short) through introducing fuzzy competition, and it's isotropic and low sensitive to noise [5]. Chung-Chia Kanga and Wen-June Wang propose a novel edge detection method based on the maximizing objective function (MOF for short), this method can avoid the occurrence of double edges and is applicable for color images [6].

Secondly, the gradient-based edge detection methods increase the computational complexity because calculations, such as square root and arctangent [3]. Most of the existing methods compute edge direction by maximizing four-directional summed magnitude differences to replace arctangent [6]. To decrease the computational cost, D.S. Kim, W.H. Lee, I.S. Kweon determine the edge magnitude using only elementary operations and the edge orientation by indexing the lookup table (LUT), where the LUT is constructed offline a priori for 256 ideal binary patterns [3].

Thirdly, the gradient-based edge detection methods need to set an appropriate threshold, however this threshold usually requires the trial and error adjustment to produce a satisfactory edge result for each different input image [3]. Most of the existing thresholding techniques are the improvement of Otsu and Canny's double-threshold [7-9]. Recently, many new approaches are proposed. Shuai Wan et al. improve the threshold selecting method based on the human visual system, which can effectively remove insensitive edges for human eyes [10]. LUO Tao et al. propose a new thresholding method according to the histogram of gradient. However these methods may be ineffective in a changing environment [11].

Finally, these gradient-based edge detectors use gray level images, and they will useless for color images because the representation of a pixel is not only a gray level but a vector in color models, e.g. RGB, HUV, YCbCr, etc., and each model has its own properties in color science [6]. Many of the existing approaches extend gray level edge detectors to each color model, and then integrate them to get the final edges [12-14]. In this case, the vector properties of color image can't be well preserved. Trahanias and Venetsanopoulos propose the famous vector order statistics color edge detectors, such as vector range (VR) edge detector, vector dispersion (VD) edge detector, minimum vector range (MVR) edge detector, and minimum vector dispersion (MVD) edge detector [15, 16]. These approaches do not have the disadvantages of the previous ones.

However none of the methods mentioned above can overcome all of the above limitations successfully. So we present a novel edge detection method to overcome these limitations in this paper. Based on the maximizing vector distance, a new edge intensity calculation method is proposed. And we also improve the traditional edge direction calculation method. By applying partial normalization, final edges are extracted automatically using Otsu even in a changing environment. The proposed approach does not perform contour tracking as does the Canny edge detector, so it requires less time than Canny. By compared to some previous methods, we have shown the effectiveness of the proposed method.

The remainder of this paper is organized as follows. In section 2, the proposed approach is presented. Section 3 proposes a color edge detection method which is extended from the method described in Section 2. The experimental results compared to some previous methods are displayed in Section 4. Finally, we present conclusions in Section 5.

## 2 Edge detection methodology

A 3×3 mask containing the center pixel $p_5$ and its eight neighbor pixels is shown in Fig. 1 (a). Fig. 1 (b) shows the four directions in which edges may appear. The bi-directional summed magnitude differences calculated in CFED may be inaccurate because only two neighbor pixels in each edge direction are taken into account. So we improve it as bellow.

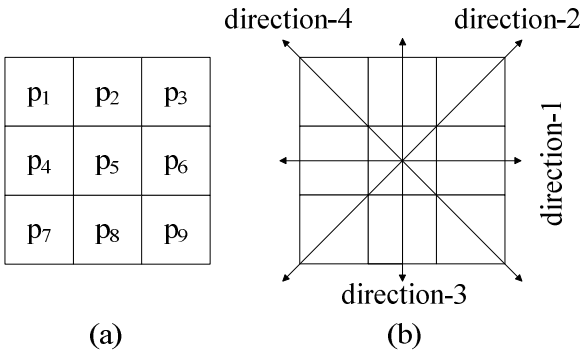


Fig. 1 (a) The 3×3 mask; (b) Four directions of edge.

In the edge pattern of direction-1, nine pixels can be divided into two sets, $S_0$ and $S_1$ as $S_0 = \{p_1, p_2, p_3, p_4, p_5, p_6\}$ and $S_1 = \{p_7, p_8, p_9\}$, or $S_0 = \{p_1, p_2, p_3\}$ and $S_1 = \{p_4, p_5, p_6, p_7, p_8, p_9\}$. The bi-directional

summed magnitude differences in gray level between $S_0$ and $S_1$ are designated by $g_1$, $g_2$, $g_3$ and $g_4$ for directions 1, 2, 3 and 4, respectively, are calculated by

$$g_1 = \frac{1}{12}\left|\left(p_1 + p_2 + p_3 + p_4 + p_5 + p_6\right) - 2 \times \left(p_7 + p_8 + p_9\right)\right| + \frac{1}{12}\left|\left(p_4 + p_5 + p_6 + p_7 + p_8 + p_9\right) - 2 \times \left(p_1 + p_2 + p_3\right)\right| \quad (1)$$

$$g_2 = \frac{1}{12}\left|\left(p_1 + p_2 + p_3 + p_5 + p_6 + p_9\right) - 2 \times \left(p_4 + p_7 + p_8\right)\right| + \frac{1}{12}\left|\left(p_1 + p_4 + p_5 + p_7 + p_8 + p_9\right) - 2 \times \left(p_2 + p_3 + p_6\right)\right| \quad (2)$$

$$g_3 = \frac{1}{12}\left|\left(p_1 + p_2 + p_4 + p_5 + p_7 + p_8\right) - 2 \times \left(p_3 + p_6 + p_9\right)\right| + \frac{1}{12}\left|\left(p_2 + p_3 + p_5 + p_6 + p_8 + p_9\right) - 2 \times \left(p_1 + p_4 + p_7\right)\right| \quad (3)$$

$$g_4 = \frac{1}{12}\left|\left(p_1 + p_2 + p_3 + p_4 + p_5 + p_7\right) - 2 \times \left(p_6 + p_8 + p_9\right)\right| + \frac{1}{12}\left|\left(p_3 + p_5 + p_6 + p_7 + p_8 + p_9\right) - 2 \times \left(p_1 + p_2 + p_4\right)\right| \quad (4)$$

The gradient magnitude of traditional gradient-based edge detection method is usually calculated as follow

$$g = \sqrt{g_1^2 + g_2^2 + g_3^2 + g_4^2} \quad (5)$$

However, this method neglects the relationship between each directional magnitude difference. For example, the gradient magnitude of some detailed region calculated by equation (5) may be very small because of interference or poor contrast. So it is difficult to extract these edges through an appropriate threshold. However, these edges can be detected through the following criterion: whether one directional magnitude difference is smaller than three other's. This paper is just proposed based on this principle, the concrete process are described as follow.

First, a four-dimensional feature vector is defined as $x = (g_1, g_2, g_3, g_4)$, where $l_o$ and $h_i$ represent low and high summed magnitude differences in the directions indicated. The feature vector can approximately be expressed as $c = (l_o, l_o, l_o, l_o)$ when four directional magnitude differences are small. Thus pixels can be divided into five classes according to the value of $g_k$.

Table 1 Pixel classes and their feature vectors.

| Pixel classes | Pixel properties | Feature vectors |
|---|---|---|
| Class 0 | background | $c_0 = (l_o, l_o, l_o, l_o)$ |
| Class 1 | edge | $c_1 = (l_o, h_i, h_i, h_i)$<br>$c_2 = (h_i, l_o, h_i, h_i)$<br>$c_3 = (h_i, h_i, l_o, h_i)$<br>$c_4 = (h_i, h_i, h_i, l_o)$ |
| Class 2 | background | $c_5 = (l_o, l_o, h_i, h_i)$<br>$c_6 = (l_o, h_i, l_o, h_i)$<br>$c_7 = (l_o, h_i, h_i, l_o)$<br>$c_8 = (h_i, l_o, l_o, h_i)$<br>$c_9 = (h_i, l_o, h_i, l_o)$<br>$c_{10} = (h_i, h_i, l_o, l_o)$ |
| Class 3 | background | $c_{11} = (l_o, l_o, l_o, h_i)$<br>$c_{12} = (l_o, l_o, h_i, l_o)$<br>$c_{13} = (l_o, h_i, l_o, l_o)$<br>$c_{14} = (h_i, l_o, l_o, l_o)$ |
| Class 4 | speckle | $c_{15} = (h_i, h_i, h_i, h_i)$ |

From Table 1, we can observe that one pixel belongs to edge pixel only when its feature vector satisfies "three large one small".

Then we define a minimum vector:

$$min = \left(min\left(c_l\right), min\left(c_l\right), min\left(c_l\right), min\left(c_l\right)\right), \quad l = 1, 2, \cdots, 15 \quad (6)$$

Thus the distance between feature vector and minimum vector can be expressed as:

$$d = \left\|c_l - min\right\|_2^2, l = 1, 2, \cdots, 15 \quad (7)$$

Remark: Square root calculation is omitted to decrease the computational cost. And $\|\bullet\ \|_2$ is $L_2$ norm, which represents the Euclidean distance between two vectors.

The distances of each pixel class are shown in Table 2.

Table 2 The distances of each pixel class.

| Pixel classes | Pixel properties | *min* | *d* |
|---|---|---|---|
| Class 0 | background | $(l_o, l_o, l_o, l_o)$ | 0 |
| Class 1 | edge | $(l_o, l_o, l_o, l_o)$ | $3(h_i - l_o)^2$ |
| Class 2 | background | $(l_o, l_o, l_o, l_o)$ | $2(h_i - l_o)^2$ |
| Class 3 | background | $(l_o, l_o, l_o, l_o)$ | $(h_i - l_o)^2$ |
| Class 4 | speckle | $(h_i, h_i, h_i, h_i)$ | 0 |

From Table 2, we can observe that the edge class has the largest distance. So edges can be detected

according to the vector distance instead of gradient magnitude.

In order to facilitate selecting threshold, we make partial normalization to vector $x$:

$$x'(i,j) = \frac{x(i,j)}{max(x(i,j)) + \varepsilon} \quad (8)$$

where parameter $\varepsilon$ affect the performance of detailed edges detection. The smaller the $\varepsilon$ is, the more meaningful edges can be extracted. The effectiveness of $\varepsilon$ will be shown in Section4.

Parameter $\varepsilon$ is set according to the histogram of vector $x$. Then $\varepsilon$ can be got as follow:

$$\sum_{l=1}^{\varepsilon} H(j) = q \sum_{l=1}^{n} H(j) \quad (9)$$

Here $q$ is set to 0.9~0.95 and $n$ is set to 256.

Minimum vector is redefined as:

$$min'(i,j) = \left(min(x'(i,j)), min(x'(i,j)), \right.$$
$$\left. min(x'(i,j)), min(x'(i,j)),\right) \quad (10)$$

Then we get the new vector distance as follow:

$$E(i,j) = \left\| x(i,j) - min'(i,j) \right\|_2^2 \quad (11)$$

In order to utilize non-maxima suppression to thin edges as it does in the Canny method, we have to calculate the edge direction. The edge direction angle of traditional edge detectors is usually defined as $tan^{-1}g_3/g_1$. But this method is inaccurate due to only taking into account two directional magnitude differences and it also increases processing time. It is not necessary to get the accurate angle because we only need to know which direction the edge pixel belongs to among four possible directions in the process of non-maxima suppression. Most of the existing approaches usually adopt the following formula to get edge direction:

$$D(i,j) = Arg\left(max\left(g_k(i,j)\right)\right), k = 1 \sim 4 \quad (12)$$

Although it simplifies the calculation, the result is inaccurate. For example, one pixel whose four directional magnitude differences are $g_1 = 160$, $g_2 = 150$, $g_3 = 0$, $g_4 = 170$, respectively, then its edge direction got by the above approach is 4, however its accurate direction is 1. Inaccurate edge direction may degrade the performance of non-maxima suppression, so we improve the above approach.

As mentioned above, feature vector of edge pixel satisfies "three large one small", the results of maximizing method may be inaccurate because the values of "three large" are similar. To overcome this disadvantage, we propose the minimizing method:

$$D(i,j) = Arg\left(min\left(g_k(i,j)\right)\right) \pm 2, k = 1 \sim 4 \quad (13)$$

where ± represents edge direction is opposite to the direction of the minimal directional magnitude differences. For example, direction 1 is opposite to direction 3 and direction 2 is opposite to direction 4.

Therefore, the edge map $E(i,j)$ and direction map $D(i, j)$ are generated. The following non-maxima suppression is applied to both edge and direction maps to thin edges:

(1) $D(i,j) = 1 \,\&\, g_1(i,j) > g_1(i-1,j) \,\&\, g_1(i,j) > g_1(i+1,j)$

(2) $D(i,j) = 2 \,\&\, g_2(i,j) > g_2(i-1,j+1) \,\&\, g_2(i,j) > g_2(i+1,j-1)$

(3) $D(i,j) = 3 \,\&\, g_3(i,j) > g_3(i,j-1) \,\&\, g_3(i,j) > g_3(i,j+1)$

(4) $D(i,j) = 4 \,\&\, g_4(i,j) > g_4(i-1,j-1) \,\&\, g_4(i,j) > g_4(i+1,j+1)$

Thus the center pixel is an edge point only when any one of the above four cases occurs. Excluding the above four cases, the center pixel is not an edge point.

The above analysis is summarized as the following procedure:

Step1: Generate the edge map $E(i, j)$ and direction map $D(i, j)$ from Eqs. (11) and (13);

Step2: Apply non-maxima suppression to edge map;

Step3: Adopt Otsu to extract the edge points;

Step4: Remove speckles.

# 3 Edge Detection for Color Images

Most images processed in practical applications are color images, and we have to convert them into gray images in order to use some classic gray image processing algorithms. However, this conversion not only increases processing time but also causes some image information distorting even missing. Therefore, we extend our method to color images. In a color image, RGB color model is considered here, so feature vector $x$ is replaced by

$$x(i,j) = \left[ g_{rm}(i,j), g_{gm}(i,j), g_{bm}(i,j) \right], m = 1,2,3,4$$

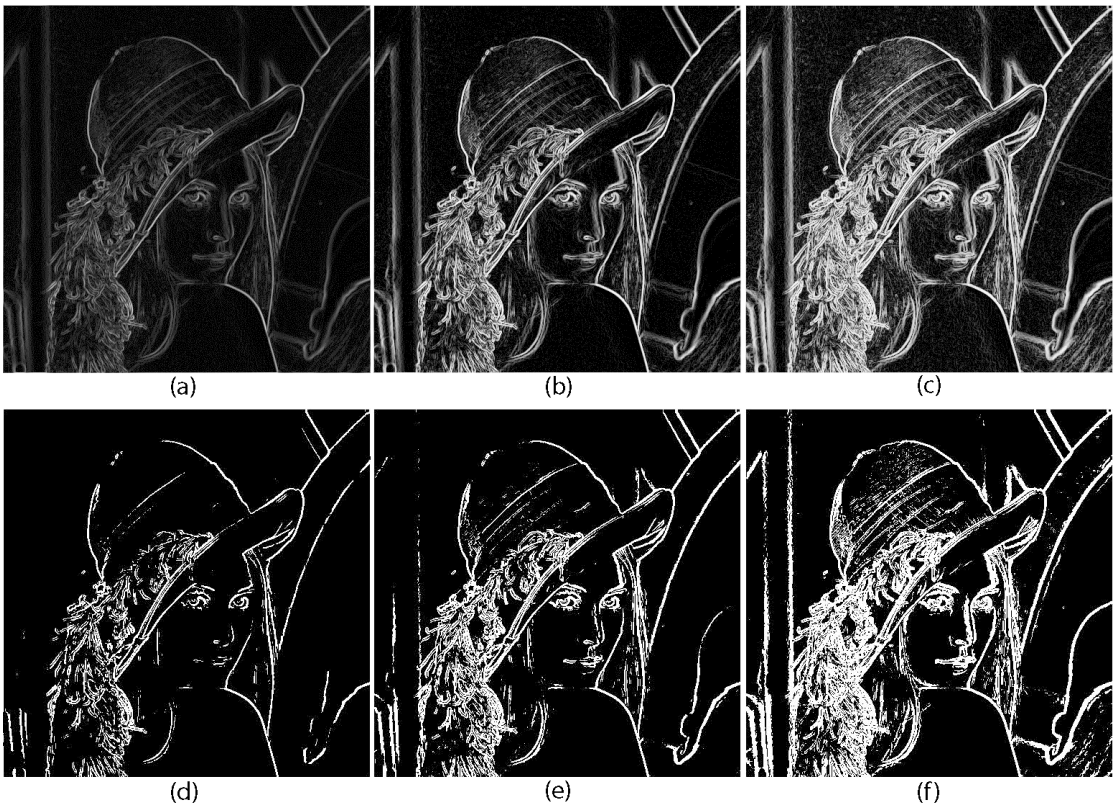


Fig. 2 Comparisons between gradient-based method and ours: (a) the edge map of gradient-based method; (b) the edge map of the proposed method with $\varepsilon = 1$; (c) the edge map of the proposed method with $\varepsilon = 3$; (d) the edge result of (a); (e) the edge result of (b); (f) the edge result of (c).

So the dimensions of feature vector and minimum vector are changed from 4 to 12. And non-maxima suppression is replaced by

(1) $D(i,j) = m \,\&\, g_{rm}(i,j) > g_{rm}(i-1,j) \,\&$
$g_{rm}(i,j) > g_{rm}(i+1,j) \,\&$
$g_{rm}(i,j) > g_{gm}(i-1,j) \,\&$
$g_{rm}(i,j) > g_{gm}(i+1,j) \,\&$
$g_{rm}(i,j) > g_{bm}(i-1,j) \,\&$
$g_{rm}(i,j) > g_{bm}(i+1,j)$

(2) $D(i,j) = m \,\&\, g_{gm}(i,j) > g_{rm}(i-1,j) \,\&$
$g_{gm}(i,j) > g_{rm}(i+1,j) \,\&$
$g_{gm}(i,j) > g_{gm}(i-1,j) \,\&$
$g_{gm}(i,j) > g_{gm}(i+1,j) \,\&$
$g_{gm}(i,j) > g_{bm}(i-1,j) \,\&$
$g_{gm}(i,j) > g_{bm}(i+1,j)$

(3) $D(i,j) = m \,\&\, g_{bm}(i,j) > g_{rm}(i-1,j) \,\&$
$g_{bm}(i,j) > g_{rm}(i+1,j) \,\&$
$g_{bm}(i,j) > g_{gm}(i-1,j) \,\&$
$g_{bm}(i,j) > g_{gm}(i+1,j) \,\&$
$g_{bm}(i,j) > g_{bm}(i-1,j) \,\&$
$g_{bm}(i,j) > g_{bm}(i+1,j)$

where $m = 1, 2, 3, 4$.

Consequently, the edge detection procedure in Section 2 is applied for the color image too.

# 4 Experimental Results

To demonstrate the efficiency of the proposed approach, several experiments based on some standard test images were conducted. We selected a few standard images: Lena, Cameraman, Airplane and Peppers. The resolution of these images is 8-bit per pixel and the size is 512×512.

## 4.1 The effectiveness of ε

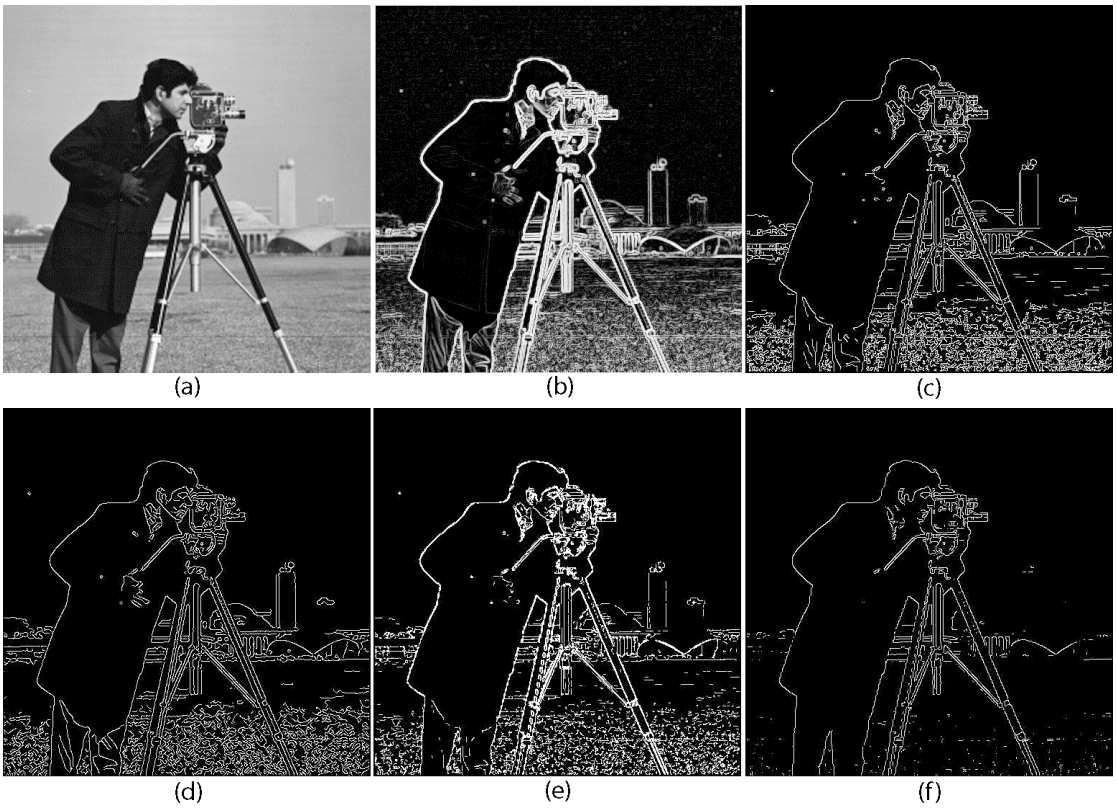


Fig. 3 Comparisons with different methods for gray level image "Cameraman": (a) the original "Cameraman" image;(b) the edge map by using the proposed method;(c) the result of the proposed method;(d) the result of Canny method;(e) the result of CFED;(f) the result of MOF.

Example 1: Consider the gray level image "Lena". The edge map generated by equation (5) is shown in Fig. 2(a), and the edge map generated by the proposed method are shown in Fig. 2(b) and Fig. 2(c), where $\varepsilon = 1$ and $\varepsilon = 3$ respectively. Apply Otsu to them, the results are shown in Fig. 2(d)-(f) respectively.

From Fig. 2(a) and Fig. 2(d) we can obviously observe that gradient-based method can only extract a few edges by applying Otsu to its edge map. But the proposed method can extract most of the meaningful edges using Otsu, and the smaller the $\varepsilon$ is, the more meaningful edges can be extracted. For instance, Fig. 2(f) has more meaningful edges than Fig. 2(d) and Fig. 2(e).

## 4.2 Edge Results

Example 2: Consider the gray level image "Cameraman" which is shown in Fig. 3(a). The edge results of the proposed method, Canny, CFED and MOF are shown in Fig. 3(c)-(f) respectively. Their parameters are set as below. The proposed method: $q = 0.9$. Canny: $Th = 30$, $Tl = 13$, $\sigma = 1$. CFED: $l_o = 5$, $h_i = 30$, $w = 240$. MOF: $w_1 = 90$, $w_2 = 40$, and threshold is generated by Otsu.

Example 3: Consider the gray level image "Airplane" which is shown in Fig. 4. The edge results of the proposed method, Canny, CFED and MOF are shown in Fig. 4(c)-(f) respectively. Their parameters are set as below. The proposed method: $q = 0.9$. Canny: $Th = 35$, $Tl = 15$, $\sigma = 1$. CFED: $l_o = 5$, $h_i = 30$, $w = 220$. MOF: $w_1 = 90$, $w_2 = 40$, and threshold is generated by Otsu.

From example 2 and example 3 we can clearly observe that the edges detected by the proposed method are one pixel width, clear and consistent. Furthermore, the proposed method detects most and clearly meaningful edges and the parameter is very easy to set. Note that the parameter in Fig. 3(c) is exactly the same as in the simulation of Fig. 4(c). And the threshold can be generated by Otsu for different images. Although the Canny method extracts consistent and one pixel width edges, some of the edges are distorted due to smoothing operation, especially in detailed regions, such as cameraman's f-

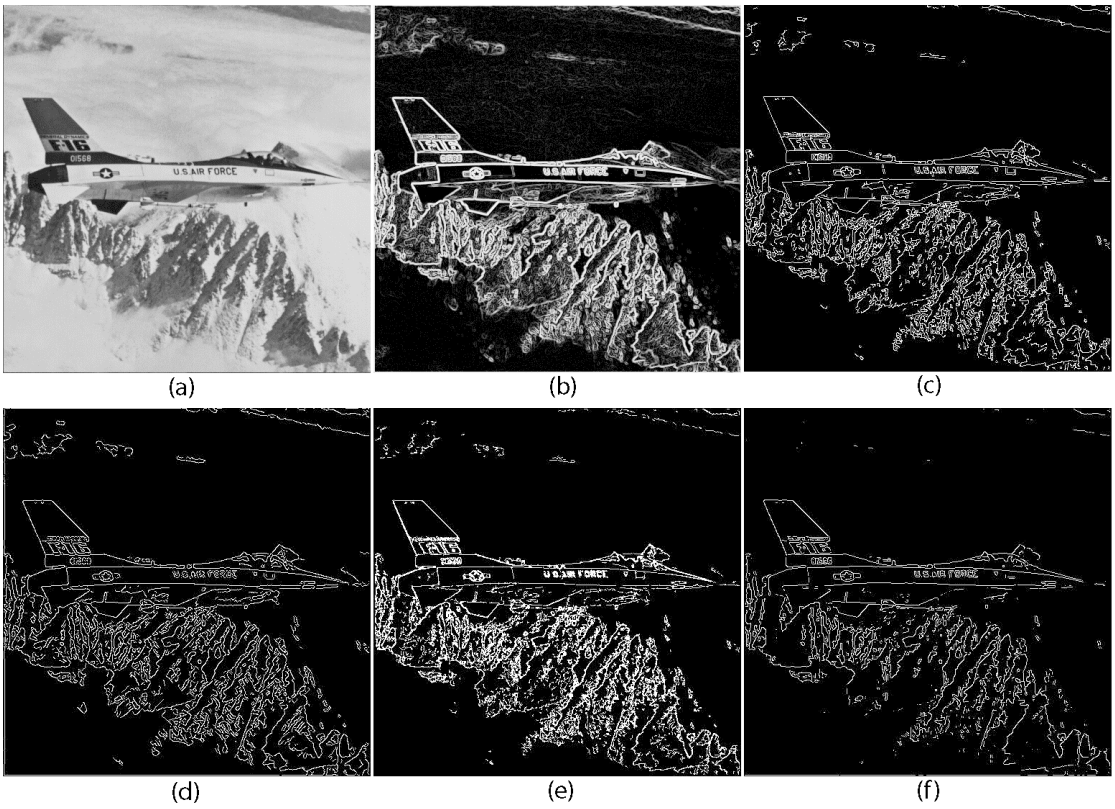


Fig. 4 Comparisons with different methods for gray level image "Airplane": (a) the original "Airplane" image;(b) the edge map by using the proposed method;(c) the result of the proposed method;(d) the result of Canny method;(e) the result of CFED;(f) the result of MOF.

ace and the camera in Fig. 3(d) and airframe's words in Fig. 4(d). The CFED method gets clear and consistent edges, however the edge lines in Fig. 3(e) and Fig. 4(e) are apparently thicker and rougher than those in Fig. 3(c)-(d) and Fig. 4(c)-(d). For MOF method, although it can detect edges successfully with an appropriate threshold, the results are unsatisfactory by applying Otsu, especially in Fig. 3(f).

Example 4: Consider the color image "Peppers" which is shown in Fig. 5(a). The edge results of the proposed method, VR, MVR and MVD are shown in Fig. 5(c)-(f) respectively. Their parameters are set as below. The proposed method: $q$ = 0.9. The thresholds of VR and MVD are generated by Otsu.

From example 4 we can easily observe that the proposed method extracts more meaningful edges and the edges are thinner compared to the vector order statistics color edge detectors. So we can say that our method is applicable for color images.

### 4.3 Speed Results

Table 3 lists the processing time for the Canny method and the proposed method using a Core 2 Duo 1.83 GHz PC for various image sizes and shows that the proposed method is slightly faster than the Canny method.

Table 3 Processing time for edge detectors.

| | 256×256 Lena | 512×512 Peppers | 640×480 Cal-box |
|---|---|---|---|
| Ours | 0.08s | 0.30s | 0.35s |
| Canny | 0.09s | 0.32s | 0.38s |

## 5 Conclusions

Based on the maximizing vector distance and partial normalization, this paper has provided a novel edge detection method to overcome the limitations of conventional gradient-based methods. The proposed method can not only apply to gray level images but also apply to color images. In addition, the edge direction computed by the proposed method is more accurate. At last, the edge results (target contour) are extracted by the simple thresholding method– Otsu even in a changing environment, which is essential in common applications of computer vision and pattern recognition such as targets tracking. The

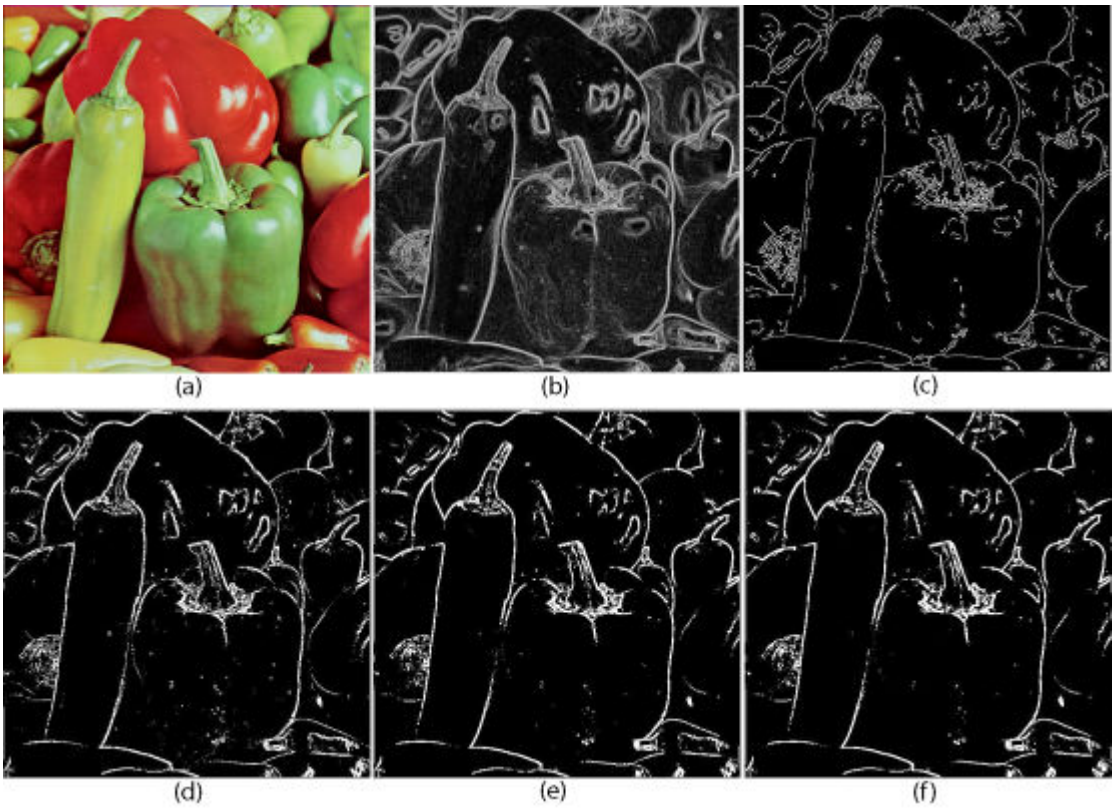


Fig. 5 Comparisons with different methods for gray color image "Peppers": (a) the original "Lena" image;(b) the edge map by using the proposed method;(c) the result of the proposed method;(d) the result of VR;(e) the result of MVR;(f) the result of MVR.

computer simulation results have also shown the comparison of edge detection results among many different methods. Obviously, the proposed method provides a better edge detection results than the previous method both in low-contrast edge detection, threshold selection and processing time.

## Acknowledgment

This research work is financially supported by National Key Technology R&D Program (2011BAK07B00), National Science and Technology Major Project (2009ZX07528-003-09) and Key Scientific and Technological Projects of Chongqing (CSCT, 2010AA2036). I would also like to give my special thanks to the anonymous reviewers of this paper for their contributions to this work.

*References:*

[1] R.C. Gonzalez, R.E. Woods, *Digital Image Processing*, Prentice-Hall, Upper Saddle River, NJ, 2002.

[2] J.F. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Analysis.* Mach. Intell. 8(6), 679–698 (1986).

[3] D.S. Kim, W.H. Lee, I.S. Kweon, Automatic edge detection using 3×3 ideal binary pixel patterns and fuzzy-based edge thresholding, *Pattern Recognition Lett*. 25 101–106 (2004).

[4] Peter Meer, Bogdan Georgescu, Edge Detection with Embedded Confidence, *IEEE Transactions on Pattern Analysis and Machine Intelligence* , VOL. 23, NO. 12, DECEMBER (2001).

[5] L.R. Liang, C.G. Looney, Competitive fuzzy edge detection, *Appl. Soft Comput*, J. 3.123–137 (2003).

[6] Chung-Chia Kanga, Wen-June Wang, A novel edge detection method based on the maximizing objective function, *Pattern Recognition*, 40.609-618 (2007).

[7] R. Medina-Carnicer, Determining Hysteresis Thresholds for Edge Detection by Combining the Advantages and Disadvantages of Thresholding Methods, *IEEE Transactions on Image Processing*, VOL. 19, NO. 1, January (2010).

[8] Yuan-Kai Huo, Gen Wei, Yu-Dong Zhang, Le-Nan Wu, An Adaptive Threshold for the Canny Operator of Edge Detection, *International Con-*

*ference on Image Analysis and Signal Processing*, Page(s):371-374 (2010).

[9] Otsu N, A threshold selection method from gray-level histogram, *IEEE Trans Systems Man Cybernetic*, (8) : 62- 65 (1978).

[10] Shuai Wan, Gradient-threshold Edge Detection based on Perceptually Adaptive Threshold Selection, *2008 3rd IEEE Conference on Industrial Electronics and Applications*, Page(s): 999 – 1002 (2008).

[11] LUO Tao, ZHENG Xi-feng, DING Tie-fu, Self-adaptive Threshold Canny Operator in Color Image Edge Detection, *Proceedings of the 2009 2nd International Congress on Image and Signal Processing*, CISP'09 (2009).

[12] R. Nevatia, A color edge detector and its use in scene segmentation, *IEEE Trans. Syst. Man Cybernetic*, 7 (11): 820–825 (1977).

[13] G.S. Robinson, Color edge detection, *Opt. Eng*, 16 (5): 479–484 (1977).

[14] S.D. Zenzo, A note on the gradient of a multi-image, *Compute Vision, Graphics, Image Process*, 33: 116–125 (1986).

[15] P.E. Trahanias, A.N. Venetsanopoulos, Color edge detection using vector order statistics, *IEEE Trans. Image Process*, 2 (2): 259–264 (1993).

[16] P.E. Trahanias, A.N. Venetsanopoulos, Vector order statistics operators as color edge detectors, *IEEE Trans. Syst. Man Cybernetic*, Part B: Cybern. 26 (1): 135–143 (1996).