ELSEVIER

A novel edge detection method based on the maximizing objective function

Chung-Chia Kang[a], Wen-June Wang[a, b,∗]

[a]*Department of Electrical Engineering, National Central University, Jhong-Li 320, Taiwan*
[b]*Department of Electrical Engineering, National Chi Nan University, Puli 545, Taiwan*

Abstract

This paper proposes a novel edge detection method for both gray level images and color images. The 3×3 mask in the image is considered and two pixel sets S_0 and S_1 in the mask are used to define an objective function. The values of the objective function corresponding to four directions determine the edge intensity and edge direction of each pixel in the mask. After all pixels in the image have been processed, the edge map and direction map are generated. Then we apply the non-maxima suppression method to the edge map and the direction map to extract the edge points. The proposed method can detect the edge successfully, while double edges, thick edges, and speckles can be avoided.

1. Introduction

In common applications of machine vision and pattern recognition, discriminating the objects from their background is one of the important tasks. In order to extract the contour of an object in an image, we must detect the complete information of the edges, so edge detection is an indispensable part of image processing. An important property of the edge detection method is its ability to extract the accurate edge line with good orientation in the considered image, and much literature on edge detection has been published in the past two decades. However, there is not yet any general performance index to judge the performance of the edge detection methods. The performance of an edge detection method is always judged subjectively because different users have different requirements for the same image. But in general, it is agreed that for a good edge detection, the edge line should be thin and with few speckles.

Some popular edge detection methods, such as Sobel method [1], Laplacian method [1], and Marr–Hildreth method [2] etc., utilize masks to do convolution on the image to detect the edges based on the abrupt change of the gray level. But in the region with smooth gray level variation, the detected edge is always thicker. Furthermore, those methods have a common problem regarding the orientation of the edge. They cannot record the direction of the edge points. Therefore, the non-maxima suppression (NMS) [3] cannot be applied in the stage of edge extraction. Most of the above papers need a threshold to determine the edge points. Canny [3] modifies Sobel method and determines the direction angle of the edge point by analyzing vertical and horizontal edge intensities of the pixel, and then uses NMS to extract the edge points. Unfortunately Canny method cannot detect the high frequency information, especially, the line of one-pixel-width in the image but double edges appear instead. Law et al. [4] design fuzzy rules for edge detection, although this method requires a rather large and complicated rules set. Tizhoosh [5] proposes three fast edge detection methods to detect the rough edge map by fuzzy logic. Liang and Looney [6] propose a competitive

∗ Corresponding author. Tel.: +886 49 2910960x4000;
fax: +886 49 2917812.
E-mail addresses: wjwang@ncnu.edu.tw, wjwang@ee.ncu.edu.tw
(W.-J. Wang).

fuzzy edge detection (CFED) method. Both of these methods divide edge types into six patterns and use a fuzzy classifier to determine which pattern the edge type belongs to. In some detailed regions of an image, CFED cannot detect the delicate texture and it usually generates speckles. Rakesh et al. [7] estimate a threshold based on statistics for the whole image and tracks the edge points. NMS is applied to those edge regions to extract the edge pixel with largest edge intensity. However, this method is time-consuming for processing the whole image and some edge pixels cannot be detected continuously. Kim et al. [8] propose an edge confidence measure as a probability called edge shape factor (ESF) which is obtained from the statistics of many experiments processed in advance by the Sobel method.

It is noted that all those methods mentioned above detect edges using gray level images, and those methods will be useless for color images because the representation of a pixel is not only a gray level but a vector in a color space. A color image can be represented by many color models, e.g. RGB, CMY, YIQ, and HIS, etc. [1], and each color model has its own properties in color science. The previous approaches to color edge detection usually use extensions of the gray level edge detectors for color images [9–11]. In this case, the edge occurring in the adjacent pixels which have the same values in any one color component may not be detected. Trahanias and Venetsanopoulos[12,13] 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, to detect the correct edge in color image. These methods do not have the disadvantages of the previous ones [9–11].

In this paper, we present a novel edge detection method that uses both gray level image and color image, in which maximum objective function is proposed and NMS is used such that more accurate and thinner contours of the objects are extracted; and double edges, thick edges, and speckles can be eliminated as much as possible. By numerous example illustrations and simulations, we have shown the effectiveness of the proposed method.

This paper is organized as follows. Section 2 reviews some typical edge detection methods such as the Sobel method, Canny method, CFED, and vector order statistics color edge detectors. Section 3 proposes a new edge detection method, which is the main contribution of this paper. Section 4 proposes a color edge detection method which is extended from the method proposed in Section 3. Section 5 displays our edge detection results compared with the results of those methods reviewed in Section 2. Finally we present conclusions in Section 6.

2. Reviews of some previous algorithms

Before introducing the proposed algorithm, this section reviews some of the main edge detection methods, such as

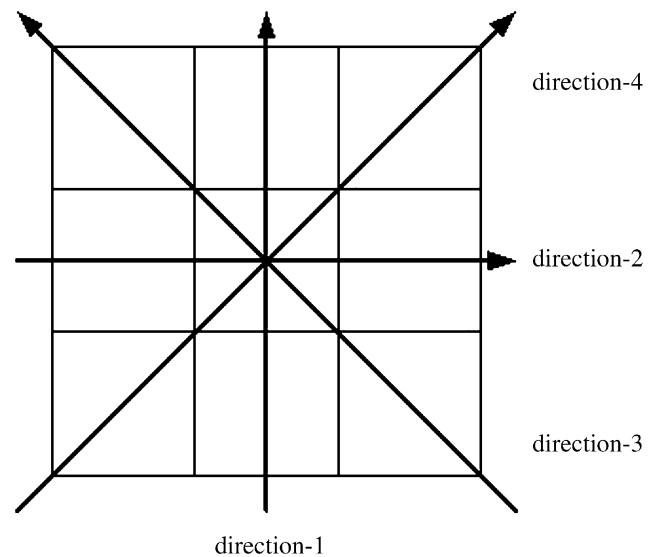Fig. 1. Two convolution masks in Sobel method.

Fig. 2. Four directions of edge.

the Sobel method, Canny method, CFED, and vector order statistics color edge detectors. The Sobel method [1] utilizes two masks, S_x and S_y, shown in Fig. 1, to do convolution on the gray image and then obtain the edge intensities E_x and E_y in the vertical and horizontal directions, respectively. The edge intensity of the mask center is defined as $|E_x| + |E_y|$. If the edge intensity of each pixel is larger than an appropriate threshold T, then the pixel is regarded as an edge point. Unfortunately, the edge line detected by Sobel method is usually thicker than the actual edge [6].

Canny method uses the same edge intensity as that of Sobel method to define the edge angle as $\tan^{-1} E_y/E_x$. Four possible edge directions in a mask, shown in Fig. 2, are used to approximate the edge angle. With the aid of information on both edge intensity and direction, NMS is used to extract the edge point with largest edge intensity alone the direction perpendicular to its own direction. The Canny method can detect much thinner edges than the Sobel method. However, both the Canny and Sobel methods will obtain zero intensity if a line of one-pixel-width passes through the mask, shown
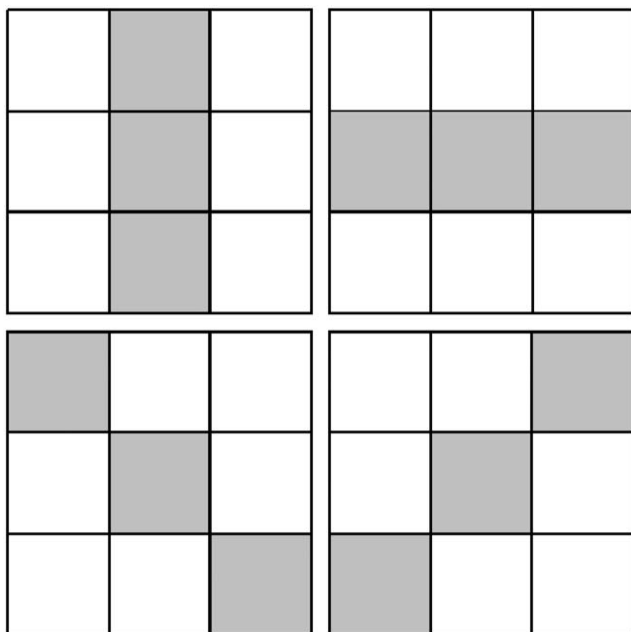
Fig. 3. The lines with one-pixel-width pass through the mask as a midline.

Fig. 4. The 3×3 mask.

Fig. 5. The ideal local edge images.

in Fig. 3, and then the incorrect edge called "double edges" on the both sides of this line will be generated. Furthermore, the edge intensity values obtained by the Sobel method and Canny method may be very large and out of some known range, so we must set the threshold based on many heuristic attempts in an unknown range.

In order to avoid the occurrence of double edges, CFED [6] is a good edge detection method. Fig. 4 shows that a 3×3 mask contains the center pixel p_5 and its eight neighbor pixels. Using CFED, the edge intensity of the 3×3 mask is measured from a feature vector $\mathbf{x} = (d_1, d_2, d_3, d_4)$, where $d_1 = |p_1 - p_5| + |p_9 - p_5|$, $d_2 = |p_2 - p_5| + |p_8 - p_5|$, $d_3 = |p_3 - p_5| + |p_7 - p_5|$, and $d_4 = |p_4 - p_5| + |p_6 - p_5|$. Considering two ideal local edge images shown in Fig. 5, CFED produces the same value of $\mathbf{x}$ for both images. This indicates that CFED may obtain the same edge intensities in two neighboring masks whose edge patterns are shown as Fig. 5, and then a thick edge occurs.
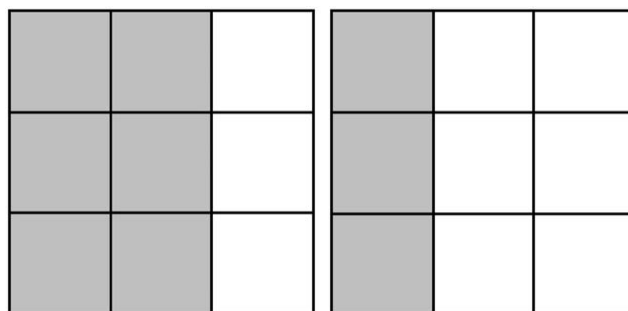
Vector order statistics color edge detectors [12,13] are the best-known edge detection methods for processing color images. Order statistics [14] plays an important role in robust signal analysis and is usually utilized to filter outliers and noises whose ranks are the highest in the sorted data. There are a number of ways proposed to perform the order statistics such as marginal ordering (M-ordering), reduced ordering (R-ordering), partial ordering (P-ordering), and conditional ordering (C-ordering). In R-ordering method [12,13], let represent a p-dimensional multivariate $\mathbf{X}^i = [x_1^i, x_2^i, \ldots, x_p^i]^{\mathrm{T}}$, where x_j^i, $j = 1, 2, \ldots, p$, are random variables. $\mathbf{X}^i$, $i = 1, 2, \ldots, n$, are n sample data. If we employ as a distance metric the aggregate distance of $\mathbf{X}^i$ to the set of vectors $\mathbf{X}^1, \mathbf{X}^2, \ldots, \mathbf{X}^n$, then

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

where $\| \cdot \|$ represents any appropriate vector norm. The arrangement of the d_is in ascending order ($d_{(1)} \leqslant d_{(2)} \leqslant \cdots \leqslant d_{(n)}$) associates the same ordering to the multivariate $\mathbf{X}^i$s

$$\mathbf{X}^{(1)} \leqslant \mathbf{X}^{(2)} \leqslant \cdots \leqslant \mathbf{X}^{(n)} \tag{2}$$

$\mathbf{X}^{(1)}$ is the vector median of the sample data with the lowest rank and $\mathbf{X}^{(n)}$ has the highest rank in (2), then the basic vector order statistics color edge detector, VR edge detector, is defined as

$$\mathrm{VR} = \|\mathbf{X}^{(n)} - \mathbf{X}^{(1)}\|, \tag{3}$$

where VR expresses the deviation of the vector outlier in the highest rank from the vector median in mask in a quantitative way. In order to avoid Gaussian noise, the VR edge detector is modified to the following VD edge detector

$$\mathrm{VD} = \left\| \mathbf{X}^{(n)} - \sum_{i=1}^{l} \frac{\mathbf{X}^{(i)}}{l} \right\|, \quad l < n, \tag{4}$$

where l is a parameter defined by the user. $\mathbf{X}^{(1)}$ in the VR of Eq. (3) is dispersed to the mean of $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(l)}$ in the VD of Eq. (4). Additionally, in order to avoid the disturbance of the outliers (salt and pepper noises),

the MVR edge detector and MVD edge detector are defined as follows:

$$\text{MVR} = \min_{j}\{\|\mathbf{X}^{(n-j+1)} - \mathbf{X}^{(1)}\|\},$$
$$j = 1, 2, \ldots, k; \quad k < n, \tag{5}$$

$$\text{MVD} = \min_{j}\left\{\left\|\mathbf{X}^{(n-j+1)} - \sum_{i=1}^{l}\frac{\mathbf{X}^{(i)}}{l}\right\|\right\},$$
$$j = 1, 2, \ldots, k; \quad k, l < n, \tag{6}$$

where k is also a parameter defined by the user. Eqs. (5) and (6) indicate that the sample data with the $k - 1$ highest ranks are ignored. Vector order statistics color edge detectors can detect only the edge intensity but cannot detect the edge direction of the center pixel in the mask under processing. Hence, NMS cannot be utilized to extract the edge as it does in the Canny method.

3. Edge detection methodology

First, the gray level image is considered. In order to avoid the occurrence of double edges, thick edges, or speckles, we propose a new method to detect the edge of an image. The edge intensity is normalized in $[0, L - 1]$ and L is the number of gray level in the digital image. For the 3×3 mask defined in Fig. 2, an edge pixel usually belongs to one of four possible direction edges (see Fig. 6). In the edge pattern of direction-1, nine pixels can be divided into two sets, S_0 and S_1 as $S_0 = \{p_1, p_2, p_4, p_5, p_7, p_8\}$ and $S_1 = \{p_3, p_6, p_9\}$, in which it does not matter whether S_0 or S_1 has higher gray level. Similarly, $S_0 = \{p_1, p_2, p_3, p_4, p_5, p_6\}$

Fig. 6. Four directions of edge.

and $S_1 = \{p_7, p_8, p_9\}$ for the edge of direction-2, $S_0 = \{p_1, p_2, p_3, p_5, p_6, p_9\}$ and $S_1 = \{p_4, p_7, p_8\}$ for the edge of direction-3, and $S_0 = \{p_1, p_2, p_3, p_4, p_5, p_7\}$ and $S_1 = \{p_6, p_8, p_9\}$ for the edge of direction-4.

The principle of the proposed method is described as follows. Consider a 3×3 mask in a local image in which any one type of edges described above may exist. It is known that if the interset distance between S_0 and S_1 in the mask is large and the intraset distances of S_0 and S_1 are small, then the compactness of sets S_0 and S_1 is high and edge intensity is large. Therefore, we need an objective function that contains the distances described above in order to estimate the edge intensity. Let us define an objective function corresponding to the edge direction-j as Eq. (7a)

$$f_i = (L - 1)\frac{N_f}{D_f}, \tag{7a}$$

where

$$N_f = \min\left(1, \frac{|m_0 - m_1|}{w_1}\right), \tag{7b}$$

$$D_f = 1 + \frac{1}{15}\sum_{\substack{p_m, p_n \in S_0 \\ m > n \\ m \neq n}}\min\left(1, \frac{|p_m - p_n|}{w_2}\right)$$
$$+ \frac{1}{3}\sum_{\substack{p_m, p_n \in S_1 \\ m > n \\ m \neq n}}\min\left(1, \frac{|p_m - p_n|}{w_2}\right), \tag{7c}$$

$w_1 = 90$, $w_2 = 40$, $m_0 = (1/6)\sum_{p_i \in S_0} p_i$, and $m_1 = (1/3)\sum_{p_i \in S_1} p_i$. It is noted that there are different values of S_0 and S_1 for different directions of j. For an 8-bit gray level image, L is equal to 256. The numerator N_f of Eq. (7a) is the degree of the interset distance between S_0 and S_1. It will be set to 1 if the distance is larger than w_1. The edge intensity is large enough if the interset distance between S_0 and S_1 is larger than the gray level w_1. If the interset distance is within the interval $[0, w_1]$, then we use a linear function to describe N_f. Here w_1 is set to be 90, which is a value selected from numerous experiments. The denominator D_f is the sum of two degrees of the intraset distances within two sets. The first term 1 is used to prevent D_f being zero. The edge intensity is small if the intraset distances are large and we can regard the degree of the intraset distance as large while the intraset distance is larger than w_2 gray levels. Hence, w_2 is set to be 40, which is a value selected from many experiments, and we utilizes a linear function to represent the degree of the intraset distance if the intraset distance is within the interval $[0, w_2]$. f_i is normalized in $[0, L - 1]$ because of the multiplication $[L - 1]$ in Eq. (7a). For a 3×3 mask in the image, we compute four objective function values, $f_1, \ldots, f_4$, corresponding to four directions, respectively. We can obtain the edge intensity $E(x, y)$ from Eq. (8) and the edge
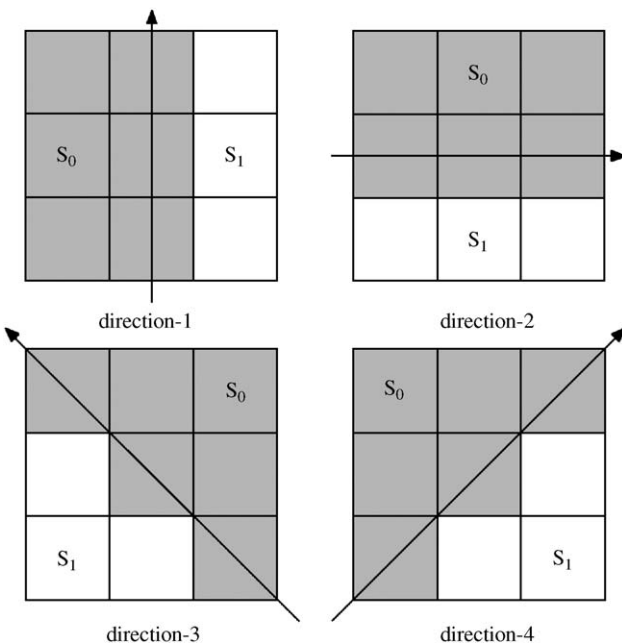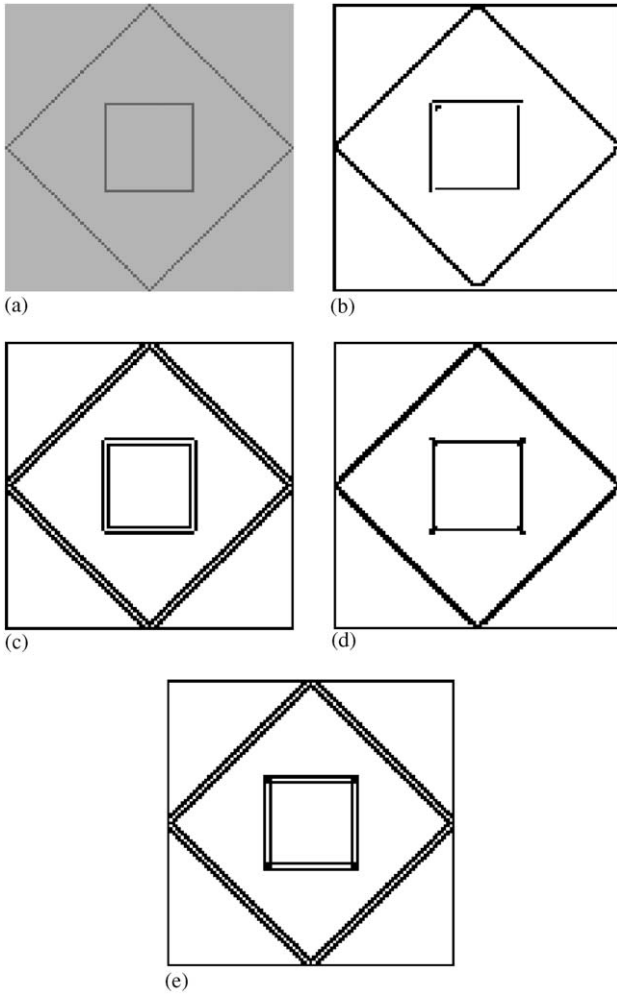
(a)   (b)   (c)   (d)   (e)

Fig. 7. Line detection comparisons using different methods.

Fig. 8. (a) The original image; (b) the result of the proposed method; (c) the result of other methods.

direction $D(x, y)$ from Eq. (9) for the center pixel (x, y) in the mask.

$$E(x, y) = \max(f_1, f_2, f_3, f_4), \tag{8}$$

$$D(x, y) = Arg(\max(f_1, f_2, f_3, f_4)). \tag{9}$$

Therefore, the edge map with each pixel (x, y) replaced by $E(x, y)$ and the direction map with each pixel replaced by $D(x, y)$ are generated. In other words, $E(x, y)$ and $D(x, y)$ of all pixels are obtained.

If $E(x, y) < T$, where T is a threshold, then the edge intensity is too small to extract any edge point. Under the case $E(x, y) \geqslant T$, the following NMS is applied to both edge and direction maps to extract the edge points. Thus the center pixel is an edge point when any one of the following four cases occurs.

(1) $D(x, y) = 1$ and $E(x, y) > E(x, y - 1)$ and $E(x, y) > E(x, y + 1)$.
(2) $D(x, y) = 2$ and $E(x, y) > E(x - 1, y)$ and $E(x, y) > E(x + 1, y)$.

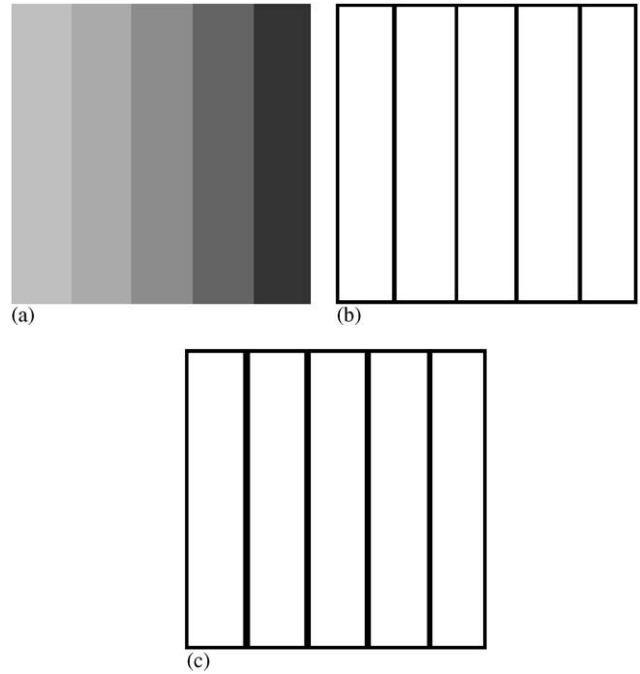(3) $D(x, y) = 3$ and $E(x, y) > E(x - 1, y + 1)$ and $E(x, y) > E(x + 1, y - 1)$.
(4) $D(x, y) = 4$ and $E(x, y) > E(x - 1, y - 1)$ and $E(x, y) > E(x + 1, y + 1)$.

Excluding the above four cases, the center pixel is not an edge point.

The above analysis is summarized as the following procedure.

Step 1: Compute four objective function values $f_1, \ldots, f_4$ from Eq. (7), respectively.

Step 2: Generate the edge map and direction map from Eqs. (8) and (9).

Step 3: Apply NMS to the edge and the direction maps and extract the edge points in the image.

Let us consider the case shown in Fig. 3, where f_i is never to be 0 no matter which direction line with one-pixel-width passes through the mask. By applying NMS, only the pixel whose edge intensity is the largest along the direction perpendicular to its edge direction can be regarded as an edge point. Therefore, the width of almost any edge line detected in the image is only one pixel. This avoids the occurring of the speckles and the double edges which are caused by the neighbor edge points closing the center edge point in the mask.

4. Edge detection for color images

This methodology is applicable not only for gray level images, but also for color images. In a color image, let the

Fig. 9. Comparisons with different methods for "cameraman" image: (a) the original "cameraman" image; (b) the edge map by using the proposed method with $w_1 = 90$, $w_2 = 40$; (c) the result of the proposed method with $T = 50$; (d) the result of Sobel method with $T = 120$; (e) the result of Canny method with $T = 120$; (f) the result of CFED with $lo = 8$, $hi = 50$, and $w = 240$.

ith pixel be expressed by a RGB color model such as $p_i = (r_i, g_i, b_i)$. The distance concept in Section 3 is also applied to the color model with vector expression $p_i = (r_i, g_i, b_i)$. Therefore the objective function (7a) is used again, but N_f in Eq. (7b) and D_f in Eq. (7c) are replaced by Eqs. (10) and (11), respectively.

$$N_f = \min\left(1, \frac{\|m_0 - m_1\|}{w_1}\right), \tag{10}$$

$$D_f = 1 + \frac{1}{15} \sum_{\substack{p_m, p_n \in S_0 \\ m > n \\ m \neq n}} \min\left(1, \frac{\|p_m - p_n\|}{w_2}\right)$$

$$+ \frac{1}{3} \sum_{\substack{p_m, p_n \in S_1 \\ m > n \\ m \neq n}} \min\left(1, \frac{\|p_m - p_n\|}{w_2}\right), \tag{11}$$

where $m_0 = (1/6)\sum_{p_i \in S_0} p_i$ and $m_1 = (1/3)\sum_{p_i \in S_1} p_i$. It should be noted that p_m or p_n in Eq. (11) is one of pixels in Fig. 4 with vector expression. Hence Eqs. (10) and (11) should use the norm value instead of the absolute value. Consequently, the edge detection procedure in Section 3 is applied for the color image too.

5. Experimental results

Several examples for gray images are illustrated to compare the proposed method with the Sobel method, Canny method, and CFED. Additionally, the proposed method in Section 4 is compared with the color edge detectors, such as VR, VD, MVR, and MVD, in Examples 5 and 6.

Example 1: Consider the image shown in Fig. 7(a). The width of those lines in the image is only one pixel. Fig. 7(b) shows that the proposed method detects the edge correctly but with one little speckle, which is disconnected and can be removed easily by any speckle reduction method. However the Sobel method and Canny method extract double edges, as shown in Figs. 7(c) and (d), respectively. The appearance of double edges should be avoided in many precise pattern recognition systems. Fig. 7(e) shows that CFED successfully detects the correct edge instead of double edges, but the edge line is thicker than those edges in Fig. 7(b).

Example 2: Fig. 8(a) shows an image with the five different gray levels of 190, 170, 140, 100, and 50 from left to right. As the threshold increases, the edges will disappear from left to right one by one. We can say that an algorithm
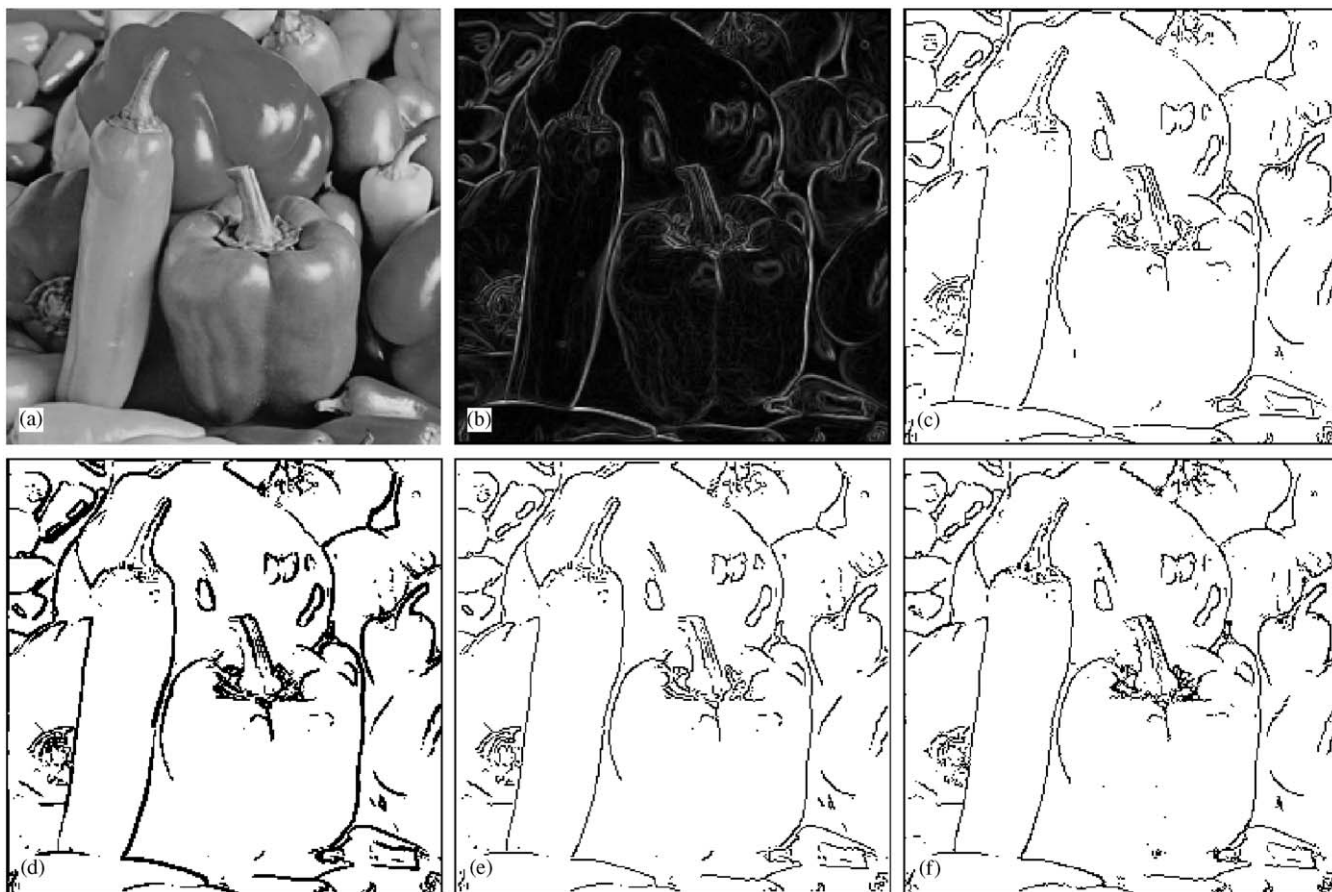
Fig. 10. Comparisons with different methods for "pepper" image: (a) the original "pepper" image; (b) the edge map by using the proposed method with $w_1 = 90$, $w_2 = 40$; (c) the result of the proposed method with $T = 50$; (d) the result of Sobel method with $T = 180$; (e) the result of Canny method with $T = 180$; (f) the result of CFED with $lo = 10$, $hi = 60$, and $w = 220$.

is robust if it is easy to tune a set of parameters such that every two neighbor regions are separated accurately. Fig. 8(b) is the edge detection result of the proposed method. Every two neighbor regions are separated accurately by a thin line. We also can obtain the same results if we set any threshold within the interval [0, 56]. By applying the Sobel method, Canny method, and CFED, the results are the same, as shown in Fig. 8(c). The thresholds from the Sobel and Canny methods are set within the interval [0, 80], which is chosen in a large unknown range. Furthermore, it is observed that the edge lines in Fig. 8(c) are clearly thicker than those in Fig. 8(b).

Example 3: Consider the "cameraman" image (Fig. 9(a)) which is of the size 256×256 and with 8-bit gray level. The proposed method with $w_1 = 90$ and $w_2 = 40$ obtains its edge map (see 9(b)). By applying NMS with $T = 50$, we extract the edge (see Fig. 9(c)). Fig. 9(d) is the result by applying Sobel method with the optimal threshold $T = 120$. We easily observe that the edge line in Fig. 9(d) is very thick and rough. Fig. 9(e) is the result of the Canny method with the optimal threshold value $T = 120$. Fig. 9(f) is the result of CFED with $lo = 8$, $hi = 50$, and $w = 240$. In some parts of

the cameraman's face and the camera, the edge points have many speckles, which can touch the actual edge points and make the details in the image disappear.

Remark: The optimal threshold value $T = 120$ in Example 3 for the Sobel and Canny methods is obtained by a series of trial and error.

Example 4: Next we consider the "pepper" image of the size 256×256 and with 8-bit gray level (Fig. 10(a)). The edge map and result of the proposed method with $w_1 = 90$, $w_2 = 40$, and $T = 50$ are shown in Figs. 10(b) and (c). Note that those parameters in Figs. 10(b) and (c) are exactly the same as in the simulations of Figs. 9(b) and (c). The Sobel method still extracts the thick edge (see Fig. 10(d)). The Canny and CFED methods (Figs. 10(e) and (f)) obtain better results than that of Sobel method by a series of parameter tuning. In Fig. 10(f), some speckles even appear on the stalk of the pepper.

Example 5: Fig. 11(a) shows the color "pepper" image. The edge map and the edge detection result of the proposed method are shown in Fig. 11(b) and (c), respectively. In this example, the RGB color model and the Euclidean norm are used. The parameters, $w_1 = 90$, $w_2 = 40$, and $T = 50$, are
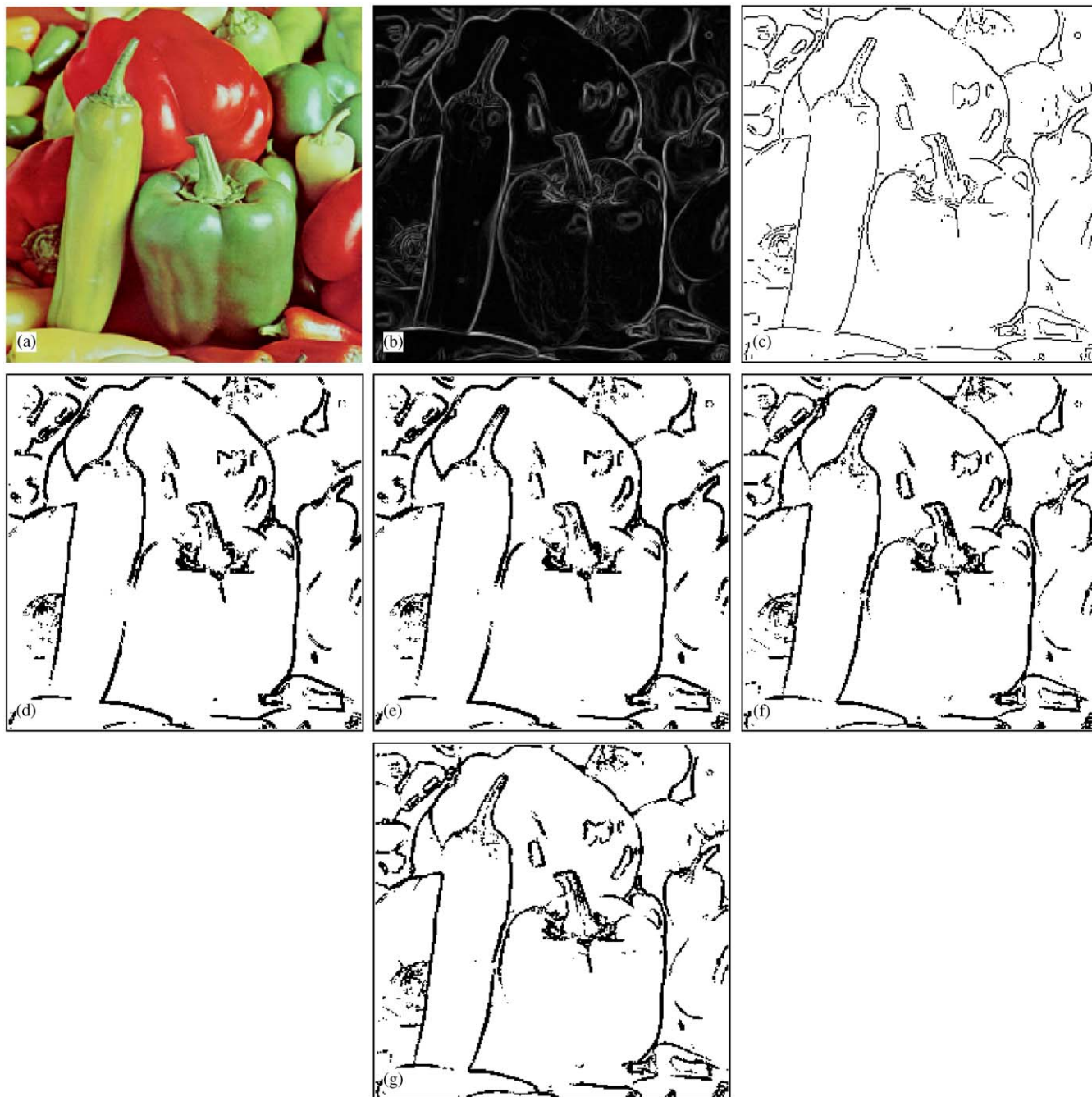
Fig. 11. Edge detection result using the proposed method for color "pepper" image: (a) the original color "pepper" image; (b) the edge map by using the proposed method with $w_1 = 90$, $w_2 = 40$; (c) the edge detection result of the proposed method with $T = 50$; (d) the result of VR with $T = 80$; (e) the result of VD with $T = 80$ and $l = 3$; (f) the result of MVR with $T = 80$ and $k = 3$; (g) the result of MVD with $T = 80$, $l = 3$, and $k = 3$.

the same as those in Examples 3 and 4. The vector order statistics operators, VR, VD, MVR, and MVD, detect the thicker edges shown in Figs. 11(d)–(g), respectively.

Example 6: Fig. 12(a) shows the color "airplane" image. Figs. 12(b) and (c) are the edge map and the edge detection result of the proposed method. Figs. 12(d)–(g) are the results of VR, VD, MVR, and MVD, respectively. In this example, the RGB color model and the Euclidean

norm are used conventionally. The proposed method still obtains the best edge detection result among those edge detectors.

6. Conclusions

Based on the maximizing objective function and NMS, this paper has provided a novel method to detect the edge
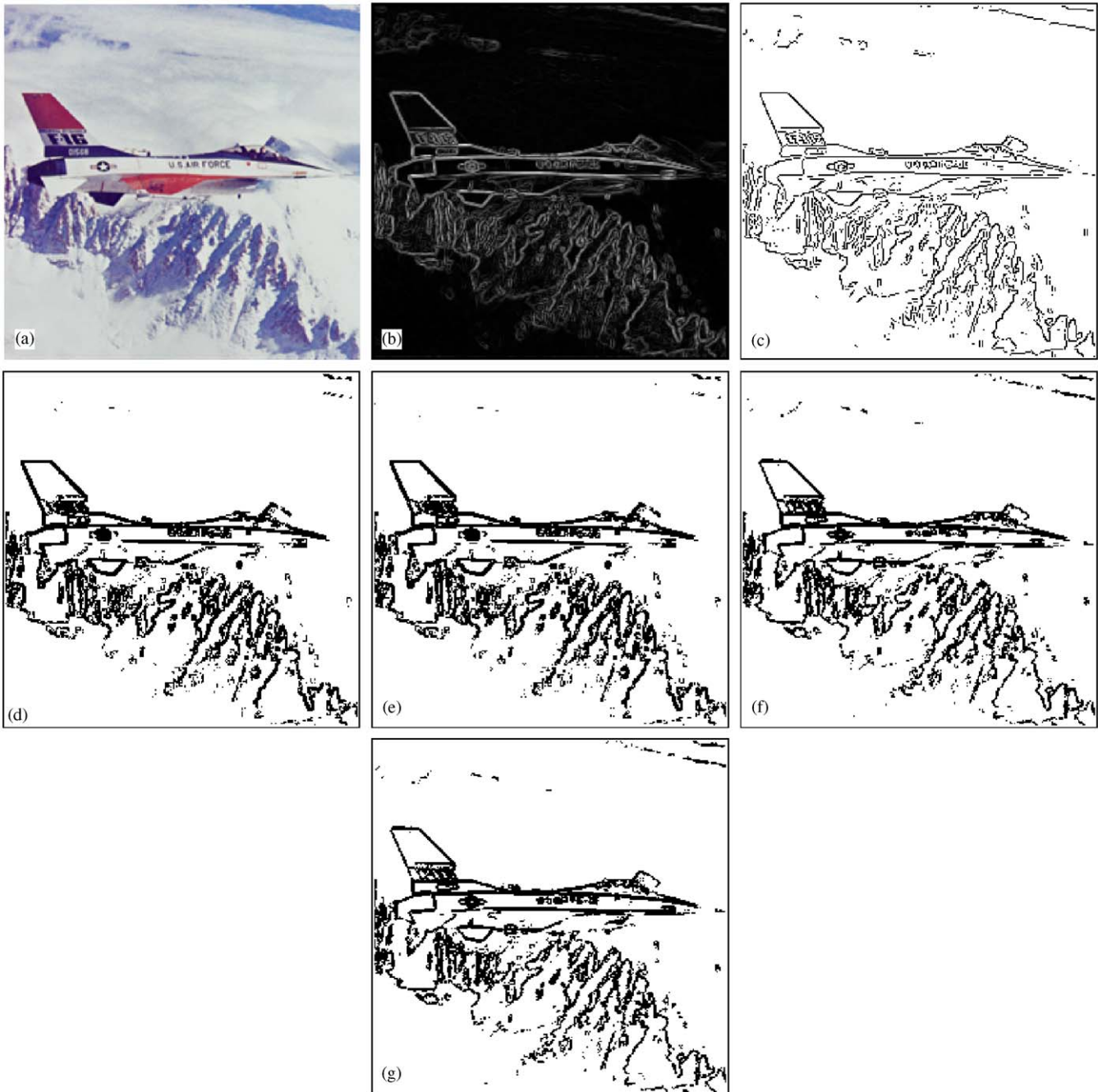
Fig. 12. Edge detection result using the proposed method for color "airplane" image: (a) the original color "airplane" image; (b) the edge map by using the proposed method with $w_1 = 90$, $w_2 = 40$; (c) the edge detection result of the proposed method with $T = 50$; (d) the result of VR with $T = 80$; (e) the result of VD with $T = 80$ and $l = 3$; (f) the result of MVR with $T = 80$ and $k = 3$; (g) the result of MVD with $T = 80$, $l = 3$, and $k = 3$.

in both gray level images and color images. The proposed method not only measures the intensity of the edge but also detects its direction, so we can obtain the edge map and the direction map. By applying NMS, the largest edge in the local image is extracted. The computer simulation results have also shown the comparison of edge detection results among many different methods. Clearly, the proposed method provides a better edge detection results than the previous methods do for the illustrated images.

Acknowledgment

References

[1] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, New York, 1992.

[2] D. Marr, E. Hildreth, Theory of edge detection, Proc. R. Soc. London 207 (1980) 197–217.

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

[4] T. Law, H. Itoh, H. Seki, Image filtering, edge detection, and edge tracing using fuzzy reasoning, IEEE Trans. Pattern Analysis Mach. Intell. 18 (1996) 481–491.

[5] H.R. Tizhoosh, Fast fuzzy edge detection, Proc. North Am. Fuzzy Inf. Process. Soc. 27–29 (2002) 239–242.

[6] L.R. Liang, C.G. Looney, Competitive fuzzy edge detection, Appl. Soft Comput. J. 3 (2003) 123–137.

[7] R.R. Rakesh, P. Chaudhuri, C.A. Murthy, Thresholding in edge detection: a statistical approach, IEEE Trans. Image Process. 13 (7) (2004) 927–936.

[8] 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 (2004) 101–106.

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

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

[11] S.D. Zenzo, A note on the gradient of a multiimage, Comput. Vision, Graphics, Image Process. 33 (1986) 116–125.

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

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

[14] H.A. David, Order Statistics, Wiley, New York, 1980.