

An Image Edge Thinning Algorithm Based on Fuzzy classifications

He Yi^{1, a}, Fu Rong^{1, b}, Liu LiYan^{1, c}, Zhang Yingqian^{1, d}

¹City Institute, Dalian University of Technology, Dalian, Liaoning Province, 116600,
P. R. China

^aheyi517@126.com, ^bfurong880@126.com, ^cliuliyang81@sina.com,

^dzhangyq@dlut.edu.cn

Keywords: Image processing; Fuzzy classification theory; Edge Thinning

Abstract. This paper has presented secondary determination of image edge based on fuzzy classification theory and pixel directivity, as well as real edge sharpening. Experiment show the algorithm can detect fuzzy image edge effectively with strong anti-noise ability, and the edge is obviously thinned.

Introduction

Edge detection is a research hotspot of the current image processing field, but many of the traditional edge detection method based on edge gradient to extract the edge information, and edge detection is so rough that affects the display effect. The reason is that the real image usually contains a large amount of incomplete or vague information, and especially the edge of the image information is often due to various factors hidden. Due to the complexity of visual features and image information is complex, correlation, incomplete and inaccurate, fuzzy theory is applied to image edge detection is not only feasible, complex but very effective [1-2]. This paper detects the edge pixels based on direction feature vectors of fuzzy classification theory and center pixel, the basic principle of the algorithm is to detect the curve or straight line with the form of radiation and with width in image, make the edge thinner through standard and remove the noise around the main edge pixels through parameter control. The algorithm has small amount of calculation, the edge is obviously thinned, noise and false edges are reduced, and the detection of edge microscopic image has an ideal effect.

Basic Structure of Fuzzy Classification Network

A basic structure of fuzzy classification network [3-4] comprises following three aspects:

- (1) Input layer: N elements, usually N -dimensional feature vectors $X=(x_1, x_2 \dots x_n)$;
- (2) There is no intermediate layer (also often called "hidden layer") in the most basic fuzzy classification network;
- (3) Output layer: usually K branch nodes, representing K kinds of outcome classifications.

In the algorithm of this paper, the feature vector in the input layer is 4-dimensional pixel direction feature vector with $N=4$. The pixel representation diagram and pixel direction vector schematic diagram are as shown in Figure 1.

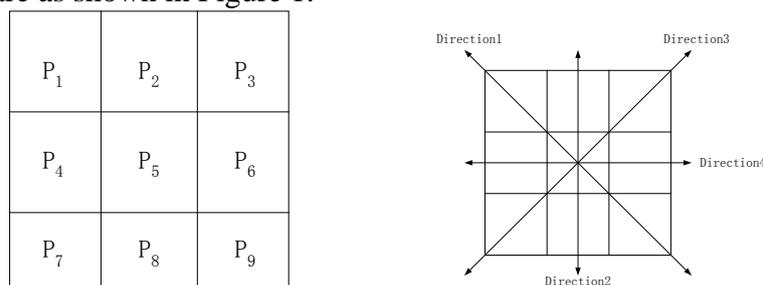


Fig.1. 3×3 Pixel template and pixel adjacent four-direction vector diagram

The pixel representation diagram takes p_5 as the center pixel, surrounding four directions as 3×3 pixel templates of adjacent pixels, direction value d_i is integrated with the direction vectors of p_5 and four directions of surrounding pixels, which are respectively d_1, d_2, d_3, d_4 and respectively aimed at the directions *Direction1*, *Direction2*, *Direction3*, *Direction4* as shown in pixel direction vector diagram. The calculation expression of d_i is as shown in Table 1:

Tab.1. Calculation Formula of Direction Vector

Direction	Direction Vector	Calculation Formula
<i>Direction1</i>	d_1	$d_1 = p_1 - p_5 + p_9 - p_5 $
<i>Direction2</i>	d_2	$d_2 = p_2 - p_5 + p_8 - p_5 $
<i>Direction3</i>	d_3	$d_3 = p_3 - p_5 + p_7 - p_5 $
<i>Direction4</i>	d_4	$d_4 = p_4 - p_5 + p_6 - p_5 $

For each pixel that is not beyond the range of image in the input image, values of the center pixel with the adjacent pixels in four directions shall be calculated according to the above formula, and then integrated into the direction vector of pixels $X=(d_1, d_2, d_3, d_4)$ as the input layer element in structure of fuzzy classification, X is called the direction feature vector of pixel here.

The classification method presented in this paper divides different pixels into 6 edge classes: the first class is background class c_0 ; the second class is noise class c_5 (all pixels identified as noise shall be classified in this class); and the remaining four classes are edge classes in four directions obtained according to the above, which are respectively first direction class c_1 , second direction class c_2 , third direction class c_3 and fourth direction class c_4 .

The basic method of edge classification is as follows: firstly give a maximum and a minimum in permissible differences, and then respectively judge which class it belongs to through the obtained vector values above from four directions. For example, the first class is background class, that is to say, it is the set of pixels which have small differences with the adjacent pixels in four directions; the second class is noise class, it is in contrast to the background class, that is to say, the noise class includes pixels which have great differences with the adjacent pixels in four directions; and the remaining four edge classes shall be determined according to the difference values of the center pixel and adjacent pixels in the four directions. Meanwhile, construct a set containing 6 elements, which is the calibration feature vector set of edge classification in this algorithm $C=\{c_0, c_1, c_2, c_3, c_4, c_5\}$, use it to represent the set of 6 classes mentioned above, and each vector c_i further comprises 4 direction elements for respectively representing the difference values of adjacent pixels in the four directions. At this time, the values of the four direction elements shall be determined according to the maximum and minimum values given in advance, and are respectively represented by h_i and l_0 , the maximum or minimum is related to the region contract and density of image. Values of the 6 classes are as shown in Table 2:

Tab.2 Classifications and Vector Representations

Classification Number	Description	Classified Vector
<i>Class 0</i>	<i>Background</i>	$c_0 = (l_0, l_0, l_0, l_0)$
<i>Class 1</i>	<i>Edge</i>	$c_1 = (l_0, h_i, h_i, h_i)$
<i>Class 2</i>	<i>Edge</i>	$c_2 = (h_i, l_0, h_i, h_i)$
<i>Class 3</i>	<i>Edge</i>	$c_3 = (h_i, h_i, l_0, h_i)$
<i>Class 4</i>	<i>Edge</i>	$c_4 = (h_i, h_i, h_i, l_0)$
<i>Class 5</i>	<i>Speckle edge</i>	$c_5 = (h_i, h_i, h_i, h_i)$

The minimum and maximum values h_i and l_0 are determined by user according to the specific characteristics of the image, so as to expect to achieve the best effect. For example, the minimum value can be determined as adjacent difference gray value 5, and the maximum value can be 20 or 30. These minimum and maximum values determine the calibration feature vector set

$C=\{c_0,c_1,c_2,c_3,c_4,c_5\}$, so as to further determine the standard of edge classification.

Fuzzy Classification Standard Member *Ep* Function

Epanechnikov function [5] (hereinafter referred to as *Ep* function) is applied as a fuzzy classification standard member function in the algorithm of this paper, this function has high efficiency and high accuracy, is easy to carry out fuzzy classification of feature vectors, has a stable structure, and is also easy to increase the distinction of prominent differences. The whole process is as follows: the input feature vector is performed with functional operation with all calibration feature vectors through *Ep* function, the maximum value (accuracy of fuzzy degree) can determine which class the vector will be classified into.

The fundamental formula of *Ep* function is:

$$E(r) = \max\{1 - r^2, 0\} \quad (1)$$

When $r=0$, $E(r)$ will obtain the maximum value 1, but with r approaching 1, the value of the function shall approach 0. If the classified calibration feature vector set is added, the fundamental formula of *Ep* function will become into:

$$E(x) = \max\{1 - \mu\|x - c_i\|^2, 0\} \quad (2)$$

x is the inputted n -dimensional feature vector, c is the i th n -dimensional element in the preset classified calibration feature vector set, and μ is the control parameter. The specific expression formula of *Ep* function is as shown in Table 3.

Tab.3 *Ep* Function Calculation Formula of Input Feature Vector x

Classification Number	Description	Function Calculation Formula
<i>Class 0</i>	<i>Background</i>	$\mu_0(x) = \max\left\{0, \frac{1 - \ x - c_0\ ^2}{w^2}\right\}$
<i>Class 1</i>	<i>Edge</i>	$\mu_1(x) = \max\left\{0, \frac{1 - \ x - c_1\ ^2}{w^2}\right\}$
<i>Class 2</i>	<i>Edge</i>	$\mu_2(x) = \max\left\{0, \frac{1 - \ x - c_2\ ^2}{w^2}\right\}$
<i>Class 3</i>	<i>Edge</i>	$\mu_3(x) = \max\left\{0, \frac{1 - \ x - c_3\ ^2}{w^2}\right\}$
<i>Class 4</i>	<i>Edge</i>	$\mu_4(x) = \max\left\{0, \frac{1 - \ x - c_4\ ^2}{w^2}\right\}$
<i>Class 5</i>	<i>Speckle edge</i>	$\mu_5(x) = \max\left\{0, \frac{1 - \ x - c_5\ ^2}{w^2}\right\}$

In the formula, the width coefficient w should be large enough for ensuring that every *Ep* function can cover the area of $[0,255]^4$ (here the distance of digital channels is: $\{4 \times 256^2\}^{1/2}=512$), the quality of edge detection depends on coefficients l_0 , h_i and w , as well as the contrast of image. Here the w used is preliminarily set between 200 and 256.

Actually, the inputted feature vector often falls in one or more member functions through operation, and these functions often have cross sections, the edges obtained so far are fairly thick, the thinning thereof is involved in determining real directional edge of this paper.

Determining Standard of Real Edge Pixel

According to the four classes of edges in this paper, only the edge with largest difference values in different directions can be real edge, the standard is based on the calibration feature vector, and the directional diagram of determining real edge pixel is as shown in Figure 2.

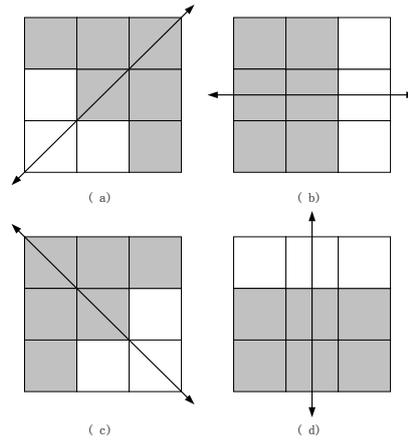


Fig.2. Directional diagram of determining edge pixel

After setting the direction of edge pixel, the real edge pixel can be determined according to the direction, and the principle of determining real edge pixel is as shown in Figure 3:

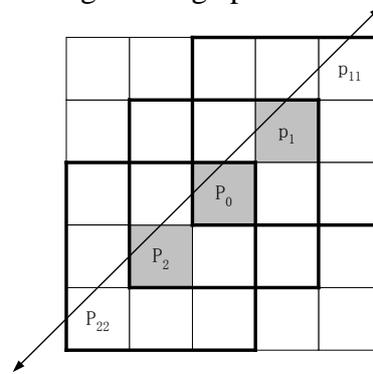


Fig.3. Principle schematic diagram of determining real edge pixel

For example, p_0 pixel is determined as an edge in first step of the algorithm, and belongs to the edge class1 (namely c_1 in C vector set), p_1 and p_2 can be respectively obtained on the two sides of p_0 along the direction d_3 (note: at this time, p_1 and p_2 have already been determined as edge pixels through the classification of the first step), at this time, the real edge pixel shall be determined among the three edge pixels through operation, for p_0 : $d_0 = |p_1 - p_0| + |p_2 - p_0|$; for p_1 : $d_1 = |p_{11} - p_1| + |p_0 - p_1|$; and for p_2 : $d_2 = |p_0 - p_2| + |p_{22} - p_2|$.

Experimental Results

Respectively using fuzzy classification theory and algorithm proposed in this paper for image named “tool” edge detection, the experimental results show in Figure4. Figure4 (a) and Figure4 (b) are the original image of “tool” and the experimental results only using fuzzy classification respectively. Figure4 (c) is the obtained image by using the proposed algorithm in this paper.

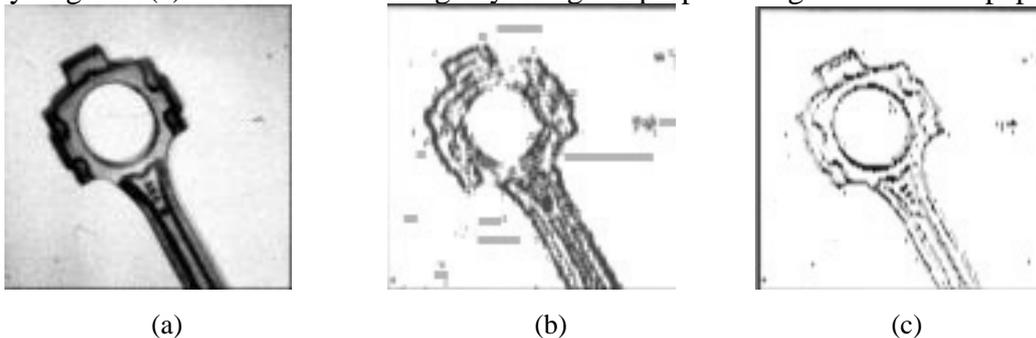


Fig.4. the experimental results

By the experimental results we can see the algorithm proposed in this paper not only detects fuzzy image edge effectively with strong anti-noise ability but gets the edge of the more clear, thinner, and more detailed.

Conclusion

This paper has carried out secondary edge determination and parameter control smoothing and sharpening of the image according to the directivity, the former has identified the false edges which are classified into edges in fuzzy classification theory, and made the detection edges thinned; the latter has selectively removed or smoothed the noise, isolated points and false edges produced in the detection process, and has sharpened the real edge. The algorithm has refined detection edge, produces little noise and false edge, and especially has better effect for detecting edge microscopic image.

References

- [1] J ACQU E Y F, COMB Y F, S TRAU SS O. Fuzzy edge detection for omnidirectional images [J]. Fuzzy Sets and Systems, 2008 , 159 (15) : 1991-2010.
- [2] L I GAN G, ZHAN G J IN2 RON G, ZHU L EI, et al. Edge detection algorithm of color remote sensing image based o n multi2 direction fuzzy mathematical morphology [J]. Micro Electronics & Computer , 2009 , 26 (8) :104-107.
- [3] SU ZH W,CAO ZH SH,WANG Y ZH.Interpretation of ambiguous zone to improve thinning result of handwritten chinese characters[J].International Journal of Pattern Recognition and Artificial Intelligence,2010,24(2):303-320.
- [4] YOSHIDA H,TANAKA N. INOVE T.On Misuse of the Connectivity Condition in the Hilditch Thinning Algorithm.The Journal of the Institute of Image Electronics Engineers of Japan,2010,39(4):490-493.
- [5] S.Abe.R.Thawonmas. A fuzzy classifier with ellipsoidal regions[J], IEEE Trans,1997,153(24):256-264.