

Investigation of Algorithms for Calculating Target Region Area

Yueqiu Jiang^{1*}, Hongwei Gao², Lei Jin¹

¹ Communication and Network Institute, Shenyang Ligong
University, Shenyang 110159, China

² College of Information Science and Engineering, Shenyang Ligong
University, Shenyang 110159, China

* Corresponding author's E-mail: missjiangyueqiu@sina.com

Abstract: It is a fundamental problem to calculate target area in pattern recognition. This paper compares the common algorithms for calculating target area, and analyzes the advantages and disadvantages of these algorithms. In order to overcome the shortcomings of common algorithms and obtain a more simple and effective algorithm, the knowledge of mathematical morphology is studied. Then, a new algorithm based on an improved algorithm of contour extracting for calculating target area is proposed. The algorithm will obtain the coordinate and type information of the contour points by extracting the contour of the target region and calculate the target area based on weighting summation of the abscissa of contour point. At last, detect whether the algorithm is effective and practical or not by comparing with several common algorithms. The result of experiment shows that this algorithm is simple, effective and easy to implement.

Keywords: Region area; contour extracting; weighted sum

1. Introduction

In pattern recognition, target area is not only an important character, but also an important basis to extract other characters. At the same time, as a region descriptor, target area is also used to calculate the roundness and measure the compactness of target region[1]. Therefore, it often needs to calculate target area. For instance, in medical image processing, in order to distinguish the type of cells, it is necessary to calculate the area of various cells. As to the image of fire, it often needs to analyze the fires according to the size of the fire points. So, it is very important to obtain a precise target area.

The common algorithms for calculating target area include count [2], region growing algorithm[3], (a algorithm based on chain code technique)[4, 5], gradient-based method, active contour model[6] and level set[7]

etc. Among these algorithms, the algorithm of count is simple and easy to understand. The algorithm in [3] is based on region filling[8]. There are many algorithms based on chain code technique. For instance, the algorithm of Tang mentioned in [9] is based on freeman chain code (8-direction). In [10], it adopts the marking algorithm to judge the type of contour points, then, it concludes a algorithm for calculating target area[9]. These methods are mainly used for binary images. The methods which are based on active contour model or level set can be directly used for gray or color images. Therefore, the application of these methods is more and more extensive. However, these algorithms often require manual initialization and the iteration number for target contour is uncertain. Therefore, it is difficult for real-time image processing.

So, several commonly used methods for binaryim-

age will be analyzed and compared. Then, a new algorithm for calculating target area based on an improved algorithm of contour extracting is proposed. The results of experiments show that this algorithm is easier to understand, simpler to operate, and has anti-interference to single pixel noise and boundary burr.

2. The Common Algorithms

According to common algorithm for calculating target area, this paper will introduce the algorithm of count, the algorithm of Tang and the marking algorithm in [10].

2.1 The algorithm of count

The basic idea of count is: In a binary image which contains only target region, find out the target pixel by traversing the whole binary image, record the number of them simultaneously, and then, the target area will be obtained by unit conversion. In the binary image, assuming the pixel value of target region is equal to 0, the pixel value of background is equal to 255, the target area can be defined as follows:

$$Area = \sum_{f(x,y)=0} 1 \quad (1)$$

where $f(x, y)$ stands for the pixel value of image pixel.

The algorithm of count is simple and easy to implement; but there is a huge limitation when being used. For example, it can not be used for the images that have a lot of interferences. When there are many target regions in image, it can not be used to calculate the area of each target region respectively.

2.2 Freeman chain code

In image processing, chain code is a commonly used coding technique of region boundary. It is firstly proposed in 1961 by Freeman, which is divided into 4-direction freeman chain code and 8-direction freeman chain code. See Fig. 1.

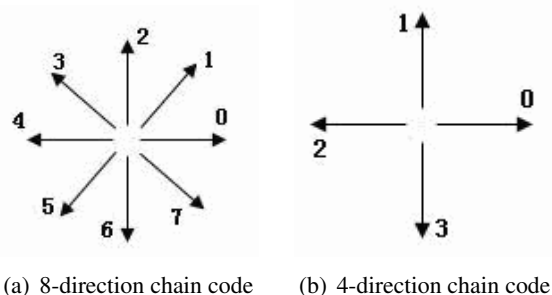


Figure 1 Freeman chain code

In the 8-direction chain code, eight directions point to eight neighborhood pixels of the target pixel. They are substituted by 0, 1, 2, 3, 4, 5, 6, 7 in the encoding process, which means that different direction chain code stands for different direction. The encoding process is as follows:

Firstly, select a boundary point as the starting point of encoding process (usually it is the boundary point of lower-left corner). Then, according to a specified direction move along the boundary pixels of target region. Meanwhile, record the direction between the adjacent pixels, until return to the starting point. So, the direction sequence is freeman chain code. The freeman chain code of target boundary is as follows:

$$s = \{a_1, \dots, a_n\}, a_i = \{0, 1, 2, 3, 4, 5, 6, 7\} \quad (2)$$

where a_i is a direction chain code between i pixel and $i+1$ pixel. If it is 4-direction chain code, a_i is as follows:

$$a_i \in \{0, 1, 2, 3\}.$$

2.3 The algorithm of Tang

The basic idea of Tang algorithm based on 8-direction freeman chain code for calculating target area is as follows:

Firstly, encode the boundary of the target region by 8-direction freeman chain code; meanwhile, record the coordinates of boundary points. Then, decide the type of boundary points according to the chain code. The type of boundary points is divided into left boundary point, right boundary point, useless boundary point and sharp point. Finally, target area can be calculated according to the type of boundary points.

The following table shows how to judge the type of boundary point according to chain code (Assuming encoding process based on counterclockwise). L is left boundary point, R is right boundary point, N is useless boundary point and LR is sharp point.

In order to calculate target area, first of all, the type of each boundary point must be judged according to Table 1. Then, it needs to combine with the type of boundary point to calculate the target area. Specific correspondence between the type of boundary point and area is as follows:

$$Area = \begin{cases} Area + Point[i].x + 1 & \text{the type is } R \\ Area - Point[i].x & \text{the type is } L \\ Area + 1 & \text{the type is } LR \end{cases} \quad (3)$$

where $Point[i].x$ is the boundary point's abscissa.

Table 1 The type of boundary points

a_i	a_{i+1}						
	0	1	2	3	4	5	6
0	N	R	R	R	N	N	N
1	N	R	R	R	LR	N	N
2	N	R	R	R	LR	LR	N
3	N	R	R	R	LR	LR	LR
4	L	N	N	N	L	L	L
5	L	LR	N	N	L	L	L
6	L	LR	LR	N	L	L	L
7	L	LR	LR	LR	L	L	L

The algorithm can obtain more precise target area. But to calculate the area of multi-target region, the algorithm will become more complex as the number of region increases.

2.4 Marking algorithm in [10]

The algorithm based on marking algorithm for calculating target area is also one of the algorithms based on freeman chain code. The basic idea of it is as follows:

Firstly, encode the boundary of target region by freeman chain code; meanwhile, record the coordinate of boundary point. Secondly, mark the boundary of target area according to freeman chain code. Marking process is divided into former-vector mark and latter-vector mark. The former-vector mark is based on the chain code which points to the current pixel. The latter-vector mark is based on the chain code which points out the current pixel. Specific principle of marking is as follows:

$$s(i) = \begin{cases} -1 & a_i = 1, 2, 3 \\ 0 & a_i = 0, 4 \\ 1 & a_i = 5, 6, 7 \end{cases} \quad (4)$$

Then, the former-vector mark adds the latter-vector mark. And according to certain rules (see [9]), the final mark will be obtained. At last, combine the final mark and the coordinates of boundary points to calculate the target area. Specific formula of area is as follows:

$$Area = \sum_{i=0}^{N-1} Point[i].X \times s(i) + N_1 \quad (5)$$

where $Point[i].X$ is the boundary point's abscissa. N_1 is the number of boundary points whose vector mark is equal to 1.

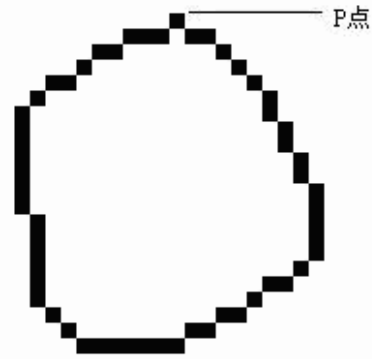


Figure 2 A sharp point on the boundary

Although it can be used for the target region which has a hole or multi-target after improved, the sharp point can not be marked in the marking process (see point P in Fig. 2). Therefore, the algorithm can not guarantee the accuracy of the target area as well.

3. The Proposed Algorithm

In order to obtain a simple and effective algorithm for calculating target area, the contour extracting algorithm based on morphology is studied.

3.1 Mathematical morphology

Mathematical morphology studies the morphological characteristics of the image. It is mainly to describe the basic characteristics and structure of image, that is, to describe the relationship between part and part, and the relationship between element and element. Erosion and dilation are two basic operations of mathematical morphology. The following are a few basic concepts in set theory.

Set It is a whole formed by things that have common nature and are usually marked in capital letters such as A, B. In image processing, usually an image is a set.

Element Each item of a set is called an element of the set. It is usually marked in lower case letters such as a, b. Each pixel point of the image is called an element. For a digital image, if pixel a is contained in image set A, then a is an element of image set A. Denoted $a \in A$. For the image set A and B, if $A \subseteq B$, then, A is called a subset of B.

Intersection It is denoted $A \cap B$, which means that the elements of a new set belong to both set A and set B. Mathematical expression is as follows:

$$A \cap B = \{a | a \in A \text{ and } a \in B\} \quad (6)$$

Union It is denoted $A \cup B$, it means that the elements

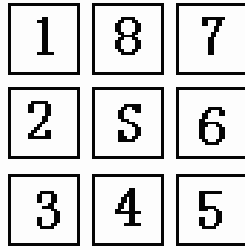


Figure 3 Square structuring element

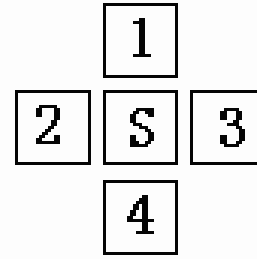


Figure 4 Cross structure element

of set belong to set A or set B . Mathematical expression is as follows:

$$A \cup B = \{a | a \in A \text{ or } a \in B\} \quad (7)$$

Image For instance, the image of image set A is denoted . Mathematical expression is as follows:

$$A = \{x | x = -a, x \in A\} \quad (8)$$

Structure Element It is one of the most important and basic concepts in mathematical morphology. In morphological transformation, its role is like the "filter window" in signal processing. It is denoted $B(x)$.

Dilation It means structure element B is used to dilate the image set A . Firstly, get the origin image of B , then, move the image as x , the result of dilation is set x when the intersection of set A and B 's image is not an empty set. Mathematical expression is as follows:

$$A \oplus B = \{x | (\hat{B})_x \cap A \neq \Phi\} \quad (9)$$

Erosion It means structure element B is used to erode the image set A . Move B as x , the result of erosion is set x when B is a subset of A . Mathematical expression is as follows:

$$A \ominus B = \{x | (B)_x \subseteq A\} \quad (10)$$

3.2 Traditional algorithm

By morphological principle, it is relatively simple to get the boundary contour of target region. Assuming is the boundary of image set A , its mathematical model can be expressed as follows:

$$\alpha(A) = A - (A \ominus B) \quad (11)$$

It means to use structure element B to erode the image set A , and then use A to subtract the result of corrosion. Structural element B is shown below.

According to the above analysis, the traditional algorithm of contour extracting based on morphology can be equivalent to the following procedures: as to a

target point, first is to get the pixel values of its eight neighborhoods; next is to judge whether the gotten pixel values are all equal to 0. If so, the point is an interior point, and then set the pixel value of corresponding point in new memory space as 255. Move the pointer to the next target point. If not, the point is a contour point. Move the pointer to the next target point. Repeat the above cycle and judgment, until the pointer crosses the border.

It can be seen from the above process, the traditional algorithm of contour extracting based on morphology is to get its interior point of the target region, and then to extract the contour of the target region by subtraction. So, the corresponding information of contour points can not be obtained. In order to calculate the area of the target region, the traditional algorithm must be improved to obtain the coordinate information and the type of contour points.

3.3 Improved algorithm

Compared with traditional algorithm, the algorithm and structure elements B of improved algorithm are different. Structure element B will use the cross structure, which is shown below. In order to understand, the structure element B can be seen as a convolution template in the whole process.

As to a binary image, the point whose pixel value is equal to 0 is seen as the target point, and the point whose pixel value is equal to 255 is seen as the background region point. The improved algorithm of contour extracting divides the target point into three major categories: the interior point, the contour point and the interference point (including boundary burr and single-pixel noise). At the same time, the contour points are divided into: left boundary point, right boundary point, left and right boundary point and independent boundary point. The definition and classification of contour points are shown in fig.5. In fig.5, the black pixels stand for the target points, and the

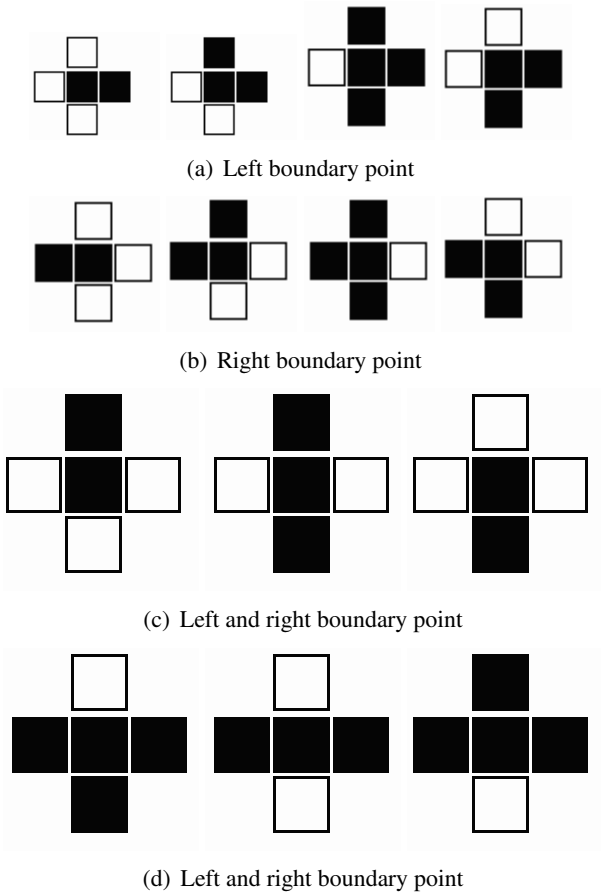


Figure 5 Classification of contour point

white pixels are background points.

In order to judge the type of target points, to a binary image, the pixels whose pixel value are equal to 255 can be recorded as 1, and pixels whose pixel value are equal to 0 as 0. When structure element B moves on the image set A, the result of intersection between the points of image set A and the points of structure element B can form 16 kinds of permutation and combination way. With fig.5 and table 2, it can be concluded that each permutation and combination corresponds to a kind of contour points. Table 2 shows the specific correspondence. So long as the relationship between target point and its neighbors can be obtained, the type information of the current target point will be obtained.

In Table 2, because S point is the current judging point, its value is always denoted as 0. The target point whose combination type is 11011 corresponds to two situations: one is that the target point is the single pixel noise in the background region; the other is that the target point is the boundary burr interference in the target region. Either point may it be, it will not be deposited in contour points array or do anything to

the corresponding type marker.

Table 2 The relationship between neighborhood point's permutation and combination, and the type of contour points

Pixel value					Combination type	Type of contour points
1	2	S	3	4		
0	0	0	0	0	00000	Interior point
0	0	0	0	1	00001	Independent boundary point
0	0	0	1	0	00010	Right boundary point
0	0	0	1	1	00011	Right boundary point
0	1	0	0	0	01000	Left boundary point
0	1	0	0	1	01001	Left boundary point
0	1	0	1	0	01010	Left and right boundary point
0	1	0	1	1	01011	Left and right boundary point
1	0	0	0	0	10000	Independent boundary point
1	0	0	0	1	10001	Independent boundary point
1	0	0	1	0	10010	Right boundary point
1	0	0	1	1	10011	Right boundary point
1	1	0	0	0	11000	Left boundary point
1	1	0	0	1	11001	Left boundary point
1	1	0	1	0	11010	Left and right boundary point
1	1	0	1	1	11011	Interference point

The specific process of improved algorithm is as follows:

Step 1: Make structure element B move on the image set A, and get the corresponding point of S point in the image set A. If the pixel value of the corresponding point is equal to 255, continue moving the pointer to the next. If it is equal to 0, go to step 2.

Step 2: Dispose the points of structure element and the points of image set, just as the following operation.

$$R = (B_1 \wedge A_1) \wedge (B_2 \wedge A_2) \wedge (B_3 \wedge A_3) \wedge (B_4 \wedge A_4) \wedge (B_5 \wedge A_5) \quad (12)$$

The result of R has two possibilities, as follows.

$$R = \begin{cases} 1 & \text{Interior point} \\ 0 & \text{contour point} \end{cases} \quad (13)$$

If R is equal to 1, move B, to find the next target point. If R is equal to 0, go to step 3.

Step 3: A permutation and combination will be formed by the result of $B_1 \wedge A_1$, $B_2 \wedge A_2$, $B_3 \wedge A_3$ and $B_4 \wedge A_4$, according to Table 2. Then, distinguish the type of the current contour points according to the combination of permutations and Table 2.

Step 4: If the combination value meets any type in the four types of contour point, put the coordinates of the pixel into the array of *Point*. At the same time store the type marker of the contour point to the array of *Nick* (The specific correspondence relationship can reference the part of the area's calculation), then the

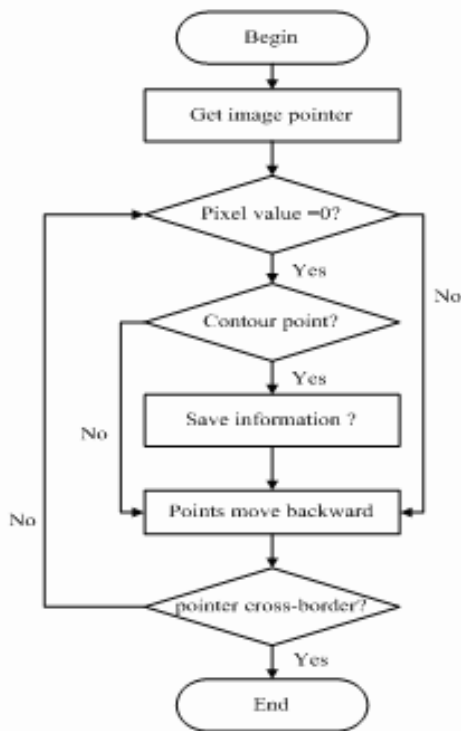


Figure 6 Flow chart of improved algorithm

pointer moves backward. If not, let the pointer point to the next directly.

Step 5: Repeat the judgment until the pointer crosses the border.

Algorithm's flow chart is as follows:

If it is only to calculate the area of one target region or do not need to calculate the area of each target region respectively, the algorithm is sufficient. If it needs to calculate the area of each target region, the algorithm needs to be improved as follows:

Steps 1: The first coordinate of Point will be saved into a new array C0, at the same time, delete the first coordinate.

Steps 2: Look for the neighborhood coordinate point through the array Point. If it is found, it will be stored in C0, and delete the original Point. Repeat this operation. If not found, go to step 3.

Steps 3: Similar to step 1. Only replace the C0 with a new array, such as 4C1.

Steps 4: Repeat the above operation, until the Point is empty.

Steps 5: Each new array obtained through the above steps is a set of target region contour points. So, the area of each target region can be calculated by reference rules.

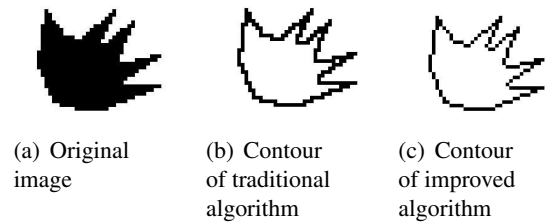


Figure 7 Contour extracting

3.4 Analysis and comparison

Although the traditional algorithm of contour extracting can extract the contour of the target region very well, they can not obtain the contour point's coordinate and type information, so it can not be used to calculate target area. The improved algorithm has the ability to obtain the information of the contour points, compared with the traditional algorithm. There exist the following differences:

- 1) The traditional algorithm uses the square structure element; but improved algorithm uses the cross structure element. Thereby the computation is reduced.
- 2) The traditional algorithm extract contour by judging whether the target point is the interior point; but the improved algorithm extract contour by judging whether the target point is the contour point.
- 3) The traditional algorithm doesn't record any information of contour point; but the improved algorithm not only records the coordinate information of the contour point, but also judges the type of the contour point.
- 4) Compared with the traditional algorithm, the improved algorithm more accurate to extract contour.

3.5 Calculation of target area

Target area can be obtained by weighting summation. Its basic principle is: to the contour points that have the same ordinate, add one to the right boundary point's abscissa and subtract the abscissa of the left boundary points, then add the number of the left and right boundary points. Next, change the ordinate and calculate line by line. At last target area is the sum of each row's data.

To calculate the area of the region, it also needs to set up different markers for different types of contour

points. Because target area is obtained by weighting summation of the abscissa of contour points, we just need the abscissa of the contour point and type marker in the process of calculation. So it needs neither record the interior point of contour and the interference point into the Point array nor make any marker in the process of judgment. The following formula is the relationship between type and marker of contour points:

$$Nick[i] = \begin{cases} -1 & \text{Left boundary point} \\ 0 & \text{Independent boundary point} \\ 1 + 1/Point[i].x & \text{Right boundary point} \\ 1/Point[i].x & \text{Left and right boundary point} \end{cases} \quad (14)$$

With the formula, the expression for the area of the target region is as follows:

$$Area = \sum_{i=0}^{n-1} Point[i].X * Nick[i] \quad (15)$$

where n is the number of contour points; $Point[i].X$ means the abscissa of contour points and $Nick[i]$ is the type maker of the contour point.

Because the judgment of target point just determines to which type the contour points belongs directly, so the type marker which corresponding to the contour point will be obtained, if the type of the target point can be affirmed. In the process of contour extracting, if the current point is one of the contour points, then Area which is target area will add the value which is equal to the contour point' abscissa $Point[i].X$ multiply the type maker of contour points $Nick[i]$. In this way, as long as the contour of target area is obtained, target area can be obtained. So it greatly improves the efficiency of algorithm's operation.

4. Experiment and Result Analysis

4.1 Computational complexity analysis

Suppose there are N pixels in image, no matter whether it is a single or multi-target region, this algorithm only traverses the image once. So, the time complexity is $O(N)$. Similarly, it is easy to understand that the time complexity of count is $O(N)$. But to the algorithm based on chain code technique, its time complexity has something to do with the number and contour of target region. It is $O(N + \sum_{i=0}^n L_i)$. Where n is the number of target region, L_i is the perimeter of target region. Therefore, this algorithm has better performance, especially for the multi-objective region.

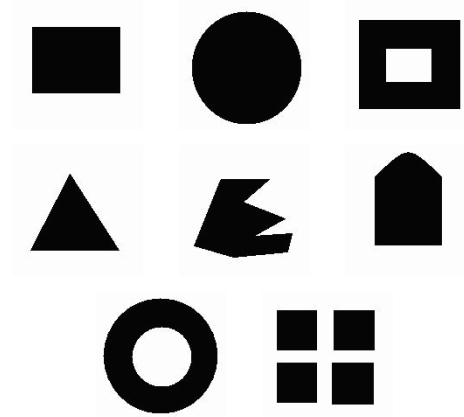


Figure 8 Test images

Table 3 Test result

Result \ Image	Image a	Image b	Image c	Image d	Image e	Image f	Image g	Image h
Actual pixels	7676	12701	9856	4532	6334	7343	9899	8698
Test results	7676	12701	9856	4532	6334	7343	9899	8698

4.2 Effectiveness analysis

To inspect the effectiveness of this algorithm, two experiments have been designed. Experiment one is used to verify the accuracy of this algorithm, while experiment two is used to verify the superiority of this algorithm, comparing with other algorithms.

Experiment one

This experiment uses a group of binary image to verify the accuracy of this algorithm. The specific images are shown in Figure 8. Table 3 is the result of experiment one. It can be seen that this algorithm has high accuracy.

Experiment two

In this experiment, a group of coin images are used. In order to accurately calculate the area of residual coins, firstly the coins' binary image are gotten through segmentation and anti-color processing (see Fig. 10). There are some noises in Figure 10(a) and in image Figure 10(b) and Figure 10(c), there are some holes. In order to test the validity of the improved algorithm, this experiment compares it with the algorithm A, B, C and D, where algorithm A being the algorithm based on algorithm of marking, algorithm B the algorithm in [3], algorithm C the algorithm of count and algorithm D Tang algorithm.

Table 4 Comparison results

Image	Area(Pixel)					
	actual area	algorithm A	algorithm B	algorithm C	algorithm D	improved algorithm
a	85742			85782		85752
b	96217	96215	97530	96217	96217	96217
a	91185	91185	98082	91185	91185	91185

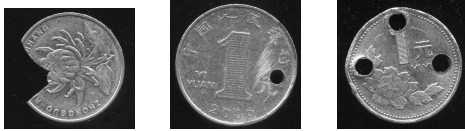


Figure 9 Original image

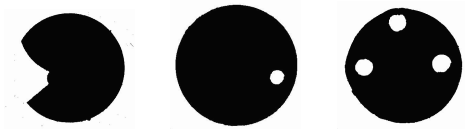


Figure 10 Binary image

The results of various algorithms are shown in Table 4, where “-” indicates that the algorithm can not be directly used to calculate the area of the target region.

In Table 4, none of the previous four algorithms can overcome the impact of the above three conditions simultaneously. However, the improved algorithm not only can accurately calculate target area, but also has some abilities to resist the interference of single pixel noise. So a precise area of the target region can be obtained by this algorithm.

5. Conclusion

This paper gives an effective and accurate algorithm for calculating target area. This algorithm is based on the improved algorithm of contour extracting, which not only has the ability to extract contour, but also can judge the coordinate information and the type of the contour points. The results of experiments show that this algorithm not only can calculate the area of single or multi-target regions accurately, but also has certain anti-interference ability. But this algorithm also has its shortcomings. For example, the anti-disturbance ability can not overcome all the interferences, and it will be the focus of later research work.

References

- [1] Rafael C.Gonzalez and Richard E.Woods, *Digital Image Processing*, Vol.2, Publishing House of Electronics Industry press,Beijing, 2003.
- [2] Guangyan Qi and Xiaodan Ma, Investigation on the Measuring Method of Leaf Area Based on Image Processing, In: *Journal of Jiamusi University* , pp.201-202, 2009.
- [3] Weihua Ge and Youguang Chen, Calculation of image's region area based on contour tracing, In: *Computer Applications and Software*, pp. 239-240, 2008.
- [4] Hongge Ren and Haitao Xu, Area statistical method of image target object based on chain code technology,

In: *Application Research of Computers*, pp.303-305, 2008.

- [5] Yuanmin Wu, Method for calculating areas of multiple regions in image based on freeman chain code, In: *Computer Engineering and Applications*, pp.199-201, 2008.
- [6] Peng Zhao, Guoqiang Ni, Zhaobang Pu and Tianwen Zhang, Simultaneous measurement of area for multiple image objects based geodesic active contour, In: *Optics and Precision Engineering* , pp.308-312, 2008
- [7] Bin Wang and Xinbo Gao. Automatic Image Segmentation Method Using Sequential Level Set, In: *Journal of Software* , pp.1185-1193, 2009.
- [8] CC. Marius and SN. Olli, An algorithm for contour-based region filling, In: *Computers Graphics*, pp.441-450, 2005.
- [9] Wankou Yang, Mingwu Ren and Jingyu Yang, Object area algorithms based on chain code in digital image, In: *Computer Engineering*, pp.30-33, 2008.
- [10] Bo Li, Donghua Liu, Guangming Liang and Chaojing Tang, A new algorithm for calculating the area of arbitrary enclosed shape, In: *Journal of National University of Defense Technology* , pp.61-64, 2002.
- [11] V. Lattuati and D. Lemoine, Closed contour extraction application to meteorological pictures, In: *Pattern Recognition* , pp. 145-152, 1982.
- [12] J.D. Dessimoz, Specialized edge-trackers for contour extraction and line-thinning , In: *Signal Processing*, pp. 71-73, 1980.