Robust Algorithm for Detecting the Maximum Inscribed Circle

Renbo Xia, Weijun Liu, Jibing Zhao, Hongyou Bian, Fei Xing
(Advanced Manufacture Lab, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China, 110016)
xiarb@sia.cn, wjliu@sia.cn, jbzhao@sia.cn, bhy@sia.cn, xingfei@sia.cn

Abstract

In this paper, we propose a new robust algorithm for the detection of the maximum inscribed circles (MIC) in images. We first use a vector distance transformation (VDT) strategy to create a distance field. Then, we globally search the maximal value in distance field to extract the medial axes. Finally, we give a procedure to determine the center and radius of MIC. Our experimental results show indeed that new algorithm is capable of detecting the MIC with excellent accuracy and high efficiency under various image conditions.

1. Introduction

Modern imaging systems such as ICT, CT, MRI and others generate a large amount of cross-section images every day. In reverse engineering, many researches are focused on converting a series of parallel cross-section images into a CAD model. For a given set of data point sets $P = \{p_i \in \Re^d \mid i = 1,2,...,n\}$, the maximum inscribed circle is the largest empty circle that is totally enclosed by these data points. MIC is one of the most basic geometric elements of data points. Whether or not MIC can be detected accurately and efficiently will directly influence the reconstruction precision of 3D model. In past decades, the problem has generated a considerable literature. An early solution of the problem is an algorithm whose worst-case running time is $O(n^3)$ [1]. Shomos and Hoey [2] showed that is problem can be solved in $O(n\log n)$ time with the help of voronoi diagram. Toussaint [3] suggested the algorithm outlined by Shamos does not always work correctly and proposed a different approach in a $O(n\log n)$ time. Roy and Zhang [4] claimed that different MIC can be constructed starting from different initial points, and the MIC with maximum width can be selected. This leads to several trials and frequently ambiguous results are obtained. Sun and Che [5] designed an iterative algorithm to solving MIC, but their algorithm does not guarantee a global optimal solution to long and narrow problem. Moreover, the selection of primal three points may affect the time spent on running program.

Although the above described methods that are reasonable solution to unorganized data points can be adopted indirectly in image processing field, they are not optimal due to without using the location information among regular pixels. To overcome those demerits of previous works, in this paper, a robust and fast algorithm for detecting the MIC in image is presented on the basis of vector distance transform.

2. Computing distance field

Assume a digital image of object has been obtained in the form of a gray scale image. The system starts with performing an edge detection process to determine the location of object contour. The contour image is then thinned so that the object contour is represented by one pixel width line.

The distance from point p to the object is the smallest distance from p to any point of the object. If the distance from any point in image to the object is known, a distance field, i.e. an image where the gray scale value is the distance to the object, is built. The maximal value in distance field and the corresponding location is regarded as the radius and center of MIC respectively. Therefore, it is suggested that MIC problem can be converted into a maximum problem of distance field and any standard method for computing the Euclidean distance can be used to solve this problem.

A direct method for obtaining a true Euclidean distance field is to repetitively calculate the distance between a point and every other point on the contour, taking the minimum value as the result. This naive, brute-force method is extremely computationally

978 -1-4244-1579-3/07/$25.00 © 2007 IEEE

expensive. It can be improved through the use of an octree and various neighbor information, but even this significant improvement does not render the method feasible. The computational expense of the Euclidean distance calculation is due to its global nature. We know that the distances vary smoothly in the distance field, so that it must be possible to deduce the value of the field in one pixel from the values of field around it. That is the fundamental idea used in all distance transformation (DT) algorithms.

Distance transformation, also known as distance field calculations, devised by Rosenfeld and Pfaltz [6], approximates Euclidean distance via local distance propagation, thereby reducing the global operation to simple addition. This reduces the execution time considerably, but at the cost of reduced accuracy. Since the original proposal, DT has been the subject of numerous research papers; see Borgefors [7, 8]. The magnitude of this interest has resulted in numerous improvements to DT, including the introduction of vector distance transform (VDT). Vector distance transform was first introduced (in two dimensions) by Danielsson [9], through a description of the sequential Euclidean distance mapping (SED) algorithms, the 4SED and 8SED, where the numeral denotes the number of neighbors used in a 3×3 matrix. Implemented in a manner similar to chamfer distance transforms [7], the SED algorithms provide a 180° propagation angle and compute a distance field in four passes.

First, the initial contour pixels, C, are segmented from the binary image domain u where the two values are often 0 for black and 1 for white

$$C = \{(x, y) : u(x, y) = 0\}\text{, where } x, y \in Z \qquad (1)$$

The second stage of a vector distance transform is the propagation of local distances throughout the field, which is initialized as

$$\overline{D}(p) = \begin{cases} (0,0) & if\ \ p \in C \\ (\infty,\infty) & otherwise \end{cases} \qquad (2)$$

To view the distance field more conveniently, a signed field is necessary and can be done by seed fill method. It is required that the image domain should be classified to indicate whether a pixel is internal, external, or on the curve

$$S(p) = \begin{cases} 1 & p \in inside(C) \\ 0 & p \in C \\ -1 & p \in outside(C) \end{cases} \qquad (3)$$

The unsigned distance field is thus the signed field multiplied by the corresponding classification value

$$\left|\overline{D}(p)\right|_{signed} = \left|\overline{D}(p)\right| \times S(p) \qquad (4)$$

Local distance propagation is achieved with a number of passes of a distance matrix, d_{mat}

$$\overline{D}(x, y) = \min(\left|\overline{D}(x+i, y+j)\right| + d_{mat}(i, j) \quad \forall i, j \in d_{mat}) \qquad (5)$$

where $x, y, i, j \in Z$ and $d_{mat}(i.j) = (m\vec{a}t_x, m\vec{a}t_y)$, mat_x is the x component of the vector element (i,j) in the mask.

Four masks (see Figure 1) are moved in the distance map in the following way. Starting from the top of the map, scan the whole map line by line by moving mask 1 form left to right along line and then mask 2 from right to left along the same line. When the forward scan is finished, starting from the bottom of map, scan the whole map line by line by moving mask 3 from right to left along line and then mask 4 from left to right along the same line.

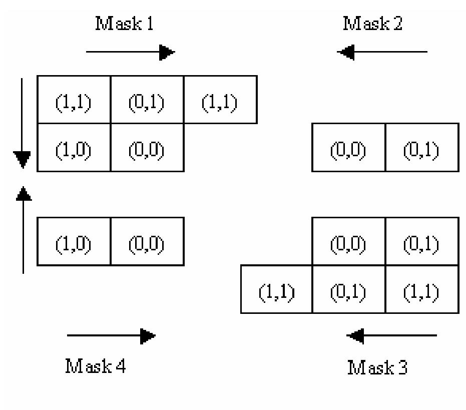


Figure 1. 8SED distance

During each pass the vector components are added to the necessary vector position, a decision is made whether any of the new vectors are minimal, and, if so, the minimal vector is stored. The minimal vector is found by calculating the magnitude of the vectors,

$$\left|\overline{D}(x, y)\right| = \sqrt{\overline{D}_x^2 + \overline{D}_y^2}$$

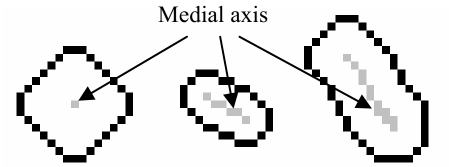
and comparing the results.

When the two scans (forward and backward scans) are finished, we regard the length of the vector as a final value of distance field. We can obtain the length by use of a two-index look-up table. The table is indexed by the two components of the vector and the content of each element in the table is the length of the vector.

The time complexity of vector distance transformation is $O(N)$, where N is the total number of pixels in the image domain. In order to make the distance field in three-dimensional dataset situations, one can adopt the vector-city vector distance transforms that were first introduced by Satherley [10].

3. Maximum inscribed circle in image

The section provides the details how to effectively find the maximum inscribed circle using distance field. We start with globally searching the maximal point whose value in distance field is maximum, and storing all those maximal points connected in the same Freeman chain code. In general case, we can get N_f Freeman chain codes for an image. It means there are N_f medial axes. As is well known, the center of MIC must be on the medial axis in graphics, but it is not always true in image because that the width of medial axis may be two pixels (see Figure 2(c)).



(a) (b) (c)

Figure 2. Computing the MIC

For the $j^{th}(j=1,2,...,L)$ point in the $i^{th}(i=1,2,...,N_f)$ Freeman chain with length L, Count the number of the maximal point in its 8-neighborhood. Assume p is a point whose count N_p is maximum,

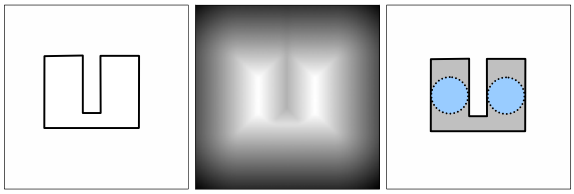
(1) If $N_p < 3$, it implies that the medial axis represented by the i^{th} Freeman chain is one pixel width line. Then, an arbitrary point on the medial axis can be considered as an eligible center of MIC and the maximal distance value of corresponding location is the radius of MIC.

(2) If $N_p >= 3$, it implies that the i^{th} medial axis is two pixels width line. Let $P(p,N_p)$ is a point sets consist of p and all maximal points in its 8-neighborhood, then the centroid p_c of $P(p,N_p)$ serves as the optimal center of MIC. Using the neighborhood information of the centroid, it is not difficult to determine the radius of MIC. Assume p_k is a point inside $P(p,N_p)$, compute the distance D_k from p_c to the object contour on the basis of the relative position between p_c and p_k, the radius of MIC is $min(D_k)$,where $k=1,2,...,N_p+1$.

4. Experimental results

We have conducted extensive experiments with various images. The algorithm is implemented in C++ with MS Visual C++ 6.0 compiler on MS Windows XP running on a desktop PC with Intel PIV 2.0G MHz processor and 512M RAM.

First, we apply the algorithm to an image that has a regular object. Figure 3(a) shows an object contour with a deep groove. Figure 3(b) shows the corresponding signed distance field, where the brightest place is the candidate center of MIC. Figure 3(c) shows the algorithm has correctly detected the MICs represented by dotted lines. In this example, there are two medial axes in object contour, for simplicity, only one MIC is drawn for every medial axis. Figure 4 shows the detection of MIC of long-thin object with little hole. To further demonstrate the robustness of our algorithm, we applied this algorithm to the detection of MIC of helical object. From the Figure 5 we can see that such special situation can be handled very well.

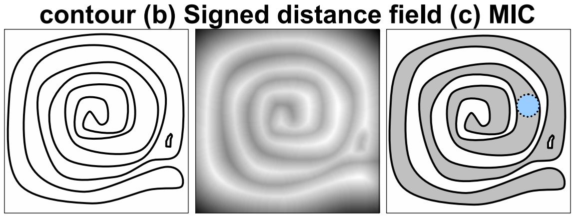


(a) (b) (c)

Figure 3. The MIC of regular object (a) Object contour (b) Signed distance field (c) MIC

(a) (b) (c)

Figure 4.The MIC of long-thin object (a) Object contour (b) Signed distance field (c) MIC



(a) (b) (c)

Figure 5. The MIC of helical object (a) Object contour (b) Signed distance field (c) MIC

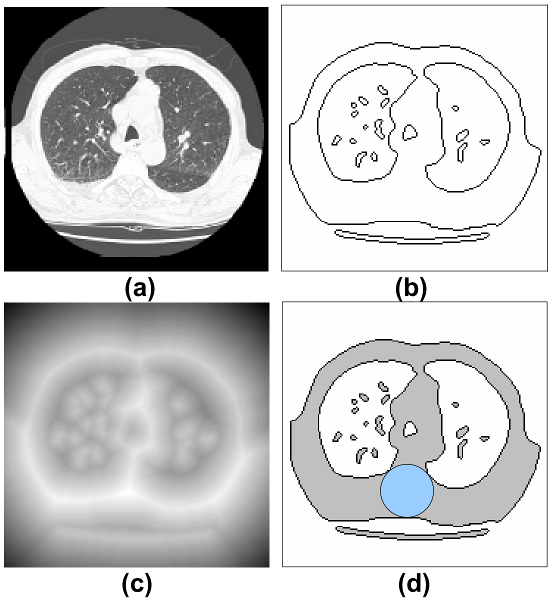


Figure 6. The MIC of CT image (a) CT image (b) Contours of CT image (c) Signed distance field (d) MIC

To show the ability of our algorithm to deal with real data, a CT image is first detected by using Canny operator [11] to extract the object contours. Then, the signed distance field is given in Figure 6(c). Finally, The MIC is found accurately. This example also demonstrates that the proposed algorithm can dispose the image with interior contours.

The experimental results are listed in Table 1, where *Er*, *Ex*, and *Ey* is respectively error of radius, x coordinate of center, and y coordinate of center of MIC. We can see some general trends from the table: for the same image, the processing time increases slightly with the number of object contour points. The error is very well maintained to a reasonable level.

Table 1. Experimental results of MIC detection

Image size *N*	*n*	*t(s)*	*Er*	*Ex*	*Ey*
256× 256	2678	0.0335	0.000117	0.000082	0.000065
	3154	0.0346	0.000092	0.000055	0.000073
512× 512	7195	0.0918	0.000122	0.000093	0.000047
	14328	0.0980	0.000128	0.000069	0.000085
1024× 1024	38604	0.4501	0.000137	0.000106	0.000071
	125742	0.4781	0.000163	0.000151	0.000098

5. Conclusions

In the paper, we have a new algorithm based on vector distance transformation for the detection of maximum inscribed circle in image. This algorithm selects vector distance transformation for creating distance field, which enable us to detect the MIC with high efficiency and excellent accuracy. Our experimental results have shown that the proposed algorithm is robust, efficient, accurate, capable of detecting multiple MICs in image.

To further improve our algorithm, we are investigating ways to reduce the errors caused by distance transformation, and developing more adaptive propagation rules to achieve a better performance in accuracy and efficiency. We are now using the algorithm in recognition of synthetic marks.

Acknowledgments

This project is supported by the Presidential Foundation of the Chinese Academy of Sciences, China (Grant No. 07AR210201).

References

[1] B. Dasarathy and L. J. White, “On Some Maximin Location of Classifier Problems”, *Computer Science Conferenc*e, Washington, D.C., 1975.

[2] M.I. Shamos and D. Hoey, “Closest-point Problems”, *Proc. 16th IEEE Symposium on Foundations of Computer Science*, Oct. 1975, pp.151-162.

[3] G. T. Toussaint, “Computing Largest Empty Circles with Location Constraints”, *International Journal of Computer and Information Sciences*, 12, 1983, pp.347-358.

[4] U.R Roy and X. Zhang. “Development and Application of Voronoi Diagrams in the Assessment of Roundness Error in an Industrial Environment”, *Computers in industrial engineering,* 26(1), 1994, pp.11-26.

[5] Y.Q. Sun and R.S. Che, “Novel Method for Solving Maximum Inscribed Circle”, *Optics and Precision Engineering*, 11(2), 2003, pp.181-186.

[6] Rosenfeld and J. L. Pfaltz, “Sequential operations in digital picture processing”, *J. ACM*, 13(4), 1966, pp.471-494.

[7] G. Borgefors, “Distance transformations in digital images”, *Comput. Vision Graph. Image Process*. 34(3), 1986, pp.344-371.

[8] G. Borgefors, “On digital distance transforms in three dimensions”, *Comput. Vision Image Understand*. 64(3), 1996, pp.368-376.

[9] P.E. Danielsson, “Euclidean distance mapping”, *Comput. Graph. Image Process*. 14, 1980, pp.227–248.

[10] R Satherley, M.W. Jones, “Vector-city vector distance transform”, *Computer Vision and Image Understanding*, 82, 2001, pp.238-254.

[11] J. Canny, “A computational approach to edge detection”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6), 1986, pp. 679-697.