Study on Algorithms of Graphic Element Recognition for Precise Vectorization of Industrial Computed Tomographic Image

Fenglin Liu*,[1,2], Bing He[2] and Bi Bi[1]

[1] *Key Laboratory of Optoelectronic Technology and System of the Education Ministry of China, Chongqing University, Chongqing 400030, China*

[2]*College of Mechanical Engineering, Chongqing University, Chongqing 400030, China*

Abstract: Circle, line and circular arc are the common basic graphic elements in industrial computed tomography (ICT) image. The algorithm of recognizing such elements is the key to industrial CT image precise vectorization. An industrial CT image vectorization system has been studied, including different recognition methods for these elements. Firstly, based on facet model, the sub-pixel edge of an industrial CT image is extracted. Then, the circles are recognized by an improved algorithm based on probability of existence map, while the lines are recognized with the set intersection algorithm of fitting a straight line, and the circular arcs are recognized by the combination of the perpendicular bisector tracing algorithm and least squares function. Finally, the graphic element parameters are measured according to recognition results, and the drawing exchange file (DXF) is produced and transmitted into the computer aided design (CAD) system to be edited and consummated. The experimental results show that these methods are capable of recognizing graphic elements in industrial CT image with an excellent accuracy, besides, the absolute errors of circles are less than 0.1 mm, and the relative errors are less than 0.5%. It can satisfy the industrial CT vectorization requirements of higher precision, rapid speed and non-contact.

Keywords: Computed tomography, facet model, edge detection, vectorization.

1. INTRODUCTION

There has been remarkable research achievement in developing and applying computerized tomography (CT) technology for medical and industrial applications, particularly in the industrial area, such as aerospace, aviation, military, machinery and automobile. Industrial CT is not only effective to detect the inner construction and flaws of an object, but also necessary to nondestructively measure the size of workpiece [1, 2]. However, the two dimensional slice images acquired by the 3rd generation ICT device can't be edited in the existing computer aided design (CAD) system. And most industrial CT images are the workpieces' dislocation images, which have many geometrical elements, such as lines, circular arcs and circles. Several conventional vectorizaion methods, applied in engineering drawings and maps, come with disadvantages, including noise sensitivity, computational complexity and low accuracy. These shortcomings make it difficult to accurately measure the size of workpieces' inner construction. In order to overcome these problems, in this paper we investigate methods of industrial CT images precise vectorization.

Since the research of image vectorization started in the early 1970s, numerous approaches have been developed [3, 4]. Broadly, these algorithms can be divided into several groups: thinning algorithm, contour tracking, Hough Transform (HT), dynamic window method and global recognition algorithm. The main advantage of thinning algorithm is that it holds a better reservation of image topology information, and its processing data points are decreased. Nevertheless, it is sensitive to noise. As a solution of this problem, the contour tracking method is provided. Its processing rate is high, and the impact of the air bubble and burr defects on the vectorization effect is reduced. However, the recognition accuracy is low since the vector image is discontinuous. In terms of accuracy and robustness, the HT and its variants have been successfully used in image vectorization. These methods are robust against outlier and occlusion, but computationally expensive. Dynamic window vectorization algorithm was proposed as a relatively novel method. A major advantage of this method presents at its high recognition accuracy to crossing point. Unfortunately, the window is changing continually, so the robustness is not enough. With the vectorization technology improving, some researchers have presented global recognition method. In process of vectorization, the line width can be obtained. Meanwhile, the corresponding tracking mode is adopted for reducing the impact of line fracture and missing data. But the accuracy of crossing point should be further improved.

In view of the disadvantages of the current image vectorization methods, the key contribution of this paper is to investigate a vectorization system for industrial CT image. In this system, the sub-pixel edge of an industrial CT image is extracted based on facet model. Its execution time is reduced significantly by removing invalid data points. Then, the circles, lines and circular arcs are recognized respectively with the corresponding recognition algorithms. These algorithms improve recognition accuracy. Also, in the process of recognition, the obtained elements parameters are dynamically

*Address correspondence to this author at the ICT Research Center, Chongqing University, Chongqing, 400030, China; Tel: +86-23-86394817(office), +86-13983075766 (Mobile); Fax: +86-23-65103562; E-mails: ict_lfl@yahoo.com.cn, hebing_ict@126.com saved in the corresponding linked lists. Hence, the required computation storage space is much cheaper than the previous methods.

2. SYSTEM OVERVIEW

In the process of industrial CT image vectorization, it is critical to detect edges in the image which is an important stage as the quantity and quality of edge data will affect greatly the vectorization performance of the system. The sub-pixel edge contour is extracted by the edge detection algorithm based on facet model, after the pre-processing as Gaussian filter, binarization. For this contour, the recognition algorithms of graphic elements are implemented. Firstly, determine whether the contour is a circle by calculating the center's probability. If it is a circle, the improved fast algorithm of circle detection based on probability of existence map is utilized to recognize it. The obtained circle parameters by the least squares function (LSF) are saved in the circle chained list. Reversely, if it is not a circle, the contour curve is fitted into many short line segments by adopting the improved set intersection algorithm of fitting a straight line. Then, merge the short line segments into the long line segments. Finally, the perpendicular bisector tracing algorithm is adopted to find the long line segments which can be fitted into circular arcs. And the merged circular arcs' parameters are derived from the least squares function. These parameters are saved in the circular arc linked list. Another long line segments' parameters are saved in the line linked list. After recognizing these elements, their parameters are outputted for vector graph. The system architecture is summarized in Fig. (**1**).

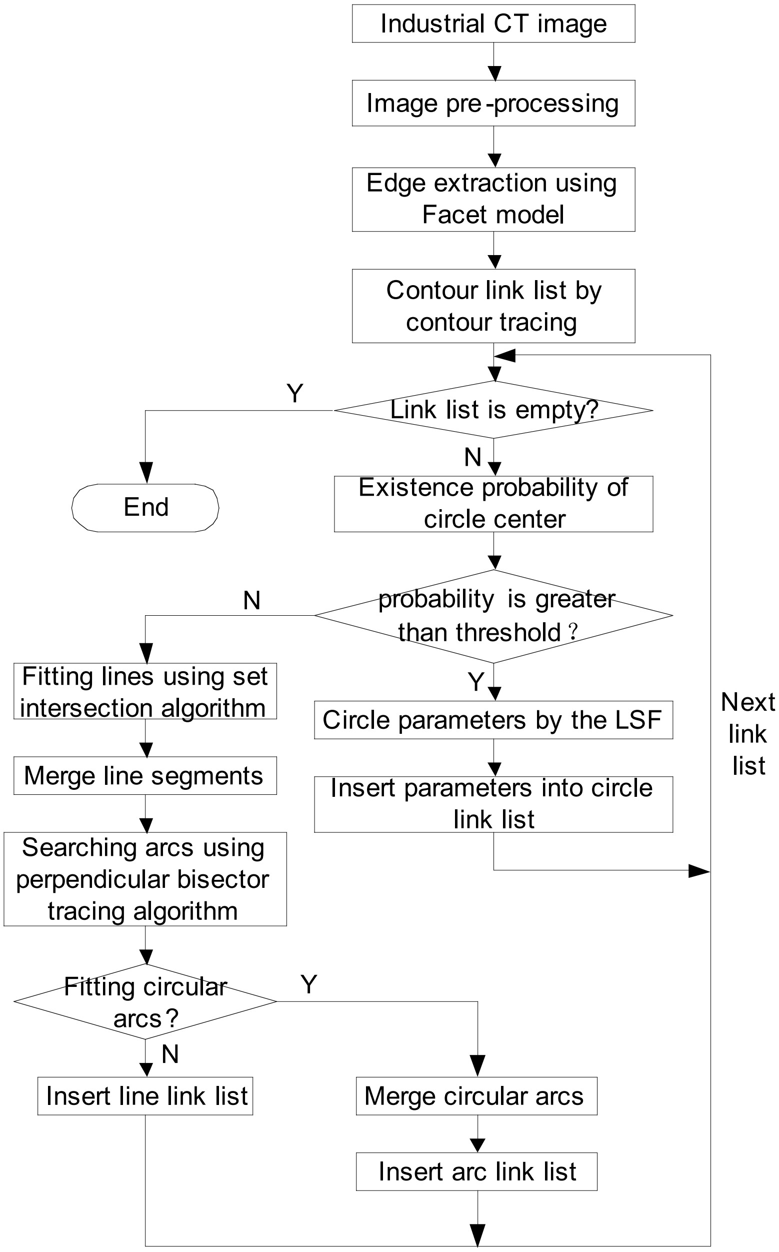
Industrial CT image
Image pre -processing
Edge extraction using Facet model
Contour link list by contour tracing
Link list is empty?
Y
End
N
Existence probability of circle center
probability is greater than threshold？
N
Y
Fitting lines using set intersection algorithm
Circle parameters by the LSF
Merge line segments
Insert parameters into circle link list
Next link list
Searching arcs using perpendicular bisector tracing algorithm
Fitting circular arcs?
Y
N
Insert line link list
Merge circular arcs
Insert arc link list

Fig. (1). Architecture of system.

3. EDGE DETECTING ALGORITHM BASED ON FACET MODEL

The exact location of edge is obtained by using the edge detection algorithm based on facet model. Through calculating the second-ordered directional derivative, edge point is defined as the zero-crossing point of facet model along the gray gradient direction [5]. Third-ordered facet model of the pixel (x, y) in 5×5 regions can be expressed as follows:

$$f(r,c) = k_1 + k_2 r + k_3 c + k_4 r^2 + k_5 rc + k_6 c^2 + k_7 r^3 + k_8 r^2 c + k_9 rc^2 + k_{10} c^3 \quad (1)$$

$$r, c \in [-2, -1, 0, 1, 2] .$$

Where (r, c) is the coordinate of any point in this pixel's neighborhood; $k_1 \sim k_{10}$ are the coefficients which are obtained by the least square method [6]. So

$$\sin\alpha = k_2 \Big/ \sqrt{k_2^2 + k_3^2} \, , \cos\alpha = k_3 \Big/ \sqrt{k_2^2 + k_3^2} \quad (2)$$

Define $r = \rho \sin\alpha$, $c = \rho\cos\alpha$ (ρ is the distance between edge point and the center of pixel), the first-ordered, second-ordered and third-ordered partial derivatives of the model can be expressed as follows:

$$\begin{cases} f'(\rho) = m_1 + 2m_2\rho + 3m_3\rho^2 \\ f''(\rho) = 2m_2 + 6m_3\rho \\ f'''(\rho) = 6m_3 \end{cases} \quad (3)$$

The edge point is the location of local extremum of the first-ordered derivative in local fitting surface; meanwhile it is the zero-crossing point of the second-ordered derivative as well. (r, c) is the edge point, if we can obtain ρ_0 (ρ_0 is within the 5×5 sub-region whose center is (r, c)) which satisfies (4), and the third-ordered partial derivative negative, as in (5).

$$f''(\rho) = 2m_2 + 6m_3\rho = 0 \Rightarrow \left| m_2 / 3m_3 \right| = \rho_0 \quad (4)$$

$$f'''(\rho) < 0 \Rightarrow m_3 < 0 \quad (5)$$

Adopting this algorithm, the subtle image detail is located. The edge detection accuracy is high enough, and the image noise is restrained to some extent.

4. RECOGNITIION OF GRAPHIC ELEMENTS

4.1. Recognition of Circle

4.1.1. Existence Probability of Circle

Circles in the X-Y plane can be completely represented by the following quadratic equation with three parameters (u, v, r) as other 2nd order curves can be.

$$(x-u)^2 + (y-v)^2 = r^2 \quad (6)$$

In order to obtain the size and position of circles in industrial CT image, the method based on probability of existence map is applied [7]. P_e is the existence probability of the circle whose center is (u, v) and radius is r. It is given by:

$$P_e = A(r) / 2\pi r \quad (7)$$

Where $A(r)$ is the amount of edge points, whose distance from the center equals r; $2\pi r$ is the amount of discrete pixels of the circle with radius r.

Every pixel (x, y) in the image is regarded as the center, the existence probability P_e is calculated and saved in the two dimensional array $\{P(x, y)\}$. The probability of existence map is produced, and the radius r is saved in the two dimensional array $\{R(x, y)\}$. Each peak in probability of existence map represents that a circle may exist in this image, taking the peak coordinates as the center and r as the radius. P_e is the probability of such existence. Fig. (**2**) is an industrial CT image with a circle. The existence probability of the center point is calculated, and the probability map is shown in Fig. (**3**). The following information can be obtained: Center point (139, 139), r =42.4946, P_e =0.8996.

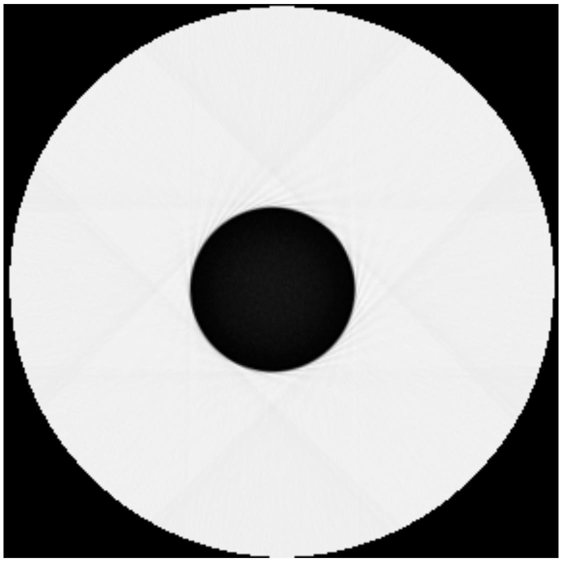

Fig. (2). Industrial CT image.

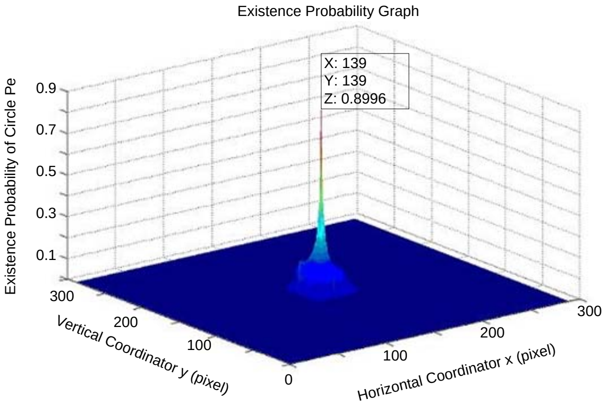

Fig. (3). Probability map of circle.

4.1.2. Improved Algorithm of Existence Probability

The above algorithm regards each pixel in the image as the center of a circle, but most pixels in the image are not the center of the circle actually. As a result, the computation is complicated and the required computational storage is expensive. The center of the circle should lie in a mini-realm in the target region, thus, we improve the above algorithm by decreasing the select range of circle center to the square *C* [8]. If a closed curve has been recognized as a circle, the maximum distance of two points on the circle is defined as diameter, and the middle point of the diameter is the circle center. As illustrated in Fig. (**4**), for every closed contour, we first have to find the nethermost point *A*, then, calculate all distances from *A* to the other points. The point *B* that has the maximum distance *D* is the endpoint of the diameter. The midpoint *O* of the diameter is approximately regarded as the centroid of the closed contour. Therefore, a square *C* is obtained, whose center and sides are the centroid *O* and $D \times w$ ($D = 100$, $w = 0.1$), respectively. The square *C* is the area within which the potential circle center will be sought. In this way, the improved method constrains the search for potential circle center only in the specified small range and thereby reduces the computational cost.

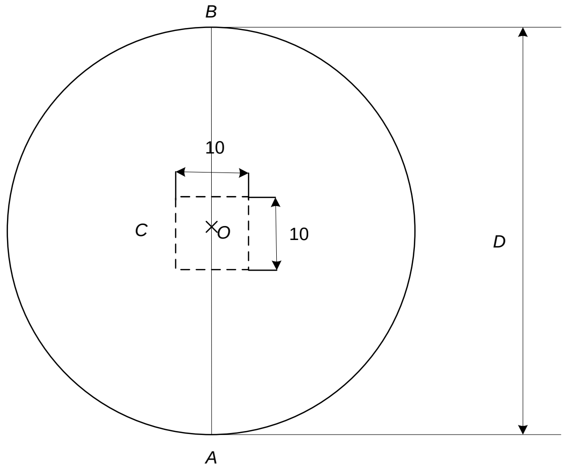

Fig. (4). Select range of circle center.

Besides, we improved the storage mode for saving memory space as well. Circle parameters, such as the coordinates of the circle center, radius and existence probability, are stored by a series of two dimensional arrays in the previous algorithm, which has to allocate memory beforehand. Thus, different types of images require different memory size. If the allocated memory is oversize, the memory space will be wasted; on the contrary, if undersized, memory space will cause data overflow. In this paper, we store the above parameters using the chained-list, referred to as the mode of allocating memory is dynamic, not beforehand. There is no need for this rule to redefine the memory size, even though the image size is different. As described in the following, the constructed list structure is effective in that it influences not only memory allocation, but also the recognition efficiency of a circle.

```
typedef struct
    {int x; // x-coordinate of circle center
    int y; // y-coordinate of circle center} Point;
struct NODE
    {Point point; // coordinates of circle center
    float r; // radius of circle
    int k; // number of edge points
    float p; // existence probability of circle
    NODE *next; // the pointer of next node};
```

4.2. Recognition of Line

After recognizing the circles in industrial CT image, the remaining contours are fitted into short line segments by an improved method of fitting a straight line. Based on the set intersection algorithm of fitting a straight line [9], the method makes use of least square technique [10] to get line segment parameters. Generally, the procedure of the method is as follows:

As shown in Fig. (**5a**). For a contour, the starting point is P_0, the current point is P_c. At the beginning of fitting, P_c=P_0. First, determine whether the next field of P_c is empty. If the next field is empty, end tracking the current line, get the line segment P_0P_c. If the next field isn't empty, judge whether the following point P_n belongs to the current line. If P_n is in the current line, then $P_n \rightarrow P_c$, and continue determining the following one. Or else, terminate tracking the current line, and get the line segment P_0P_c. Next, let P_n be the new starting point, and continue recognizing other line segments of the contour. After finishing a contour recognition, the above process is repeated for another contour of industrial CT image.

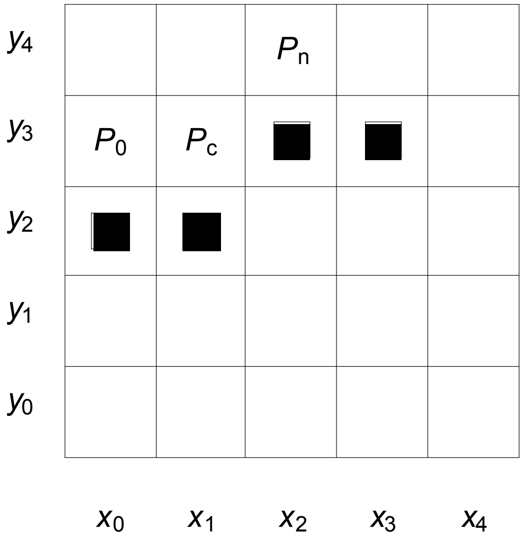

Fig. (5) (a). Sketch map of straight line fitting.

Due to low image quality and improper image preprocessing, the straight line may be mistaken as a series of short line segments, which we should merge into the long segments. The merging condition of two adjacent short line segments: the difference of slopes between two line segments must fall within the prescribed limits; and the distance between the endpoint of line segment and the merged long segment should be less than the specified value. As illustrated in Fig. (**5b**), if the angular difference between the adjacent line segments *AB* and *CD* is less than the prescribed value (using the angle value to represent the slope of line segment; here we choose 10°); and the distance from the point *B* to the line segment *AC* is less than the specified value (here 3 is appropriate), we will merge the short line segments *AB* and *BC* into the long segment *AC*. Then, determine whether the distance between the point *C* and the line segment *AD* is less than 3, and determine whether the angle difference between the line segment *CD* and the line segment *AC* is less than 10°. If the above two conditions are both met, get the long line segment *AD*. Or else, insert the merged line segment *AC* into the merged line link list, then, let *CD* be the new starting line segment, and continue merging the subsequent line segments.

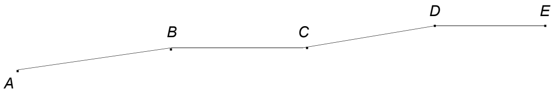

Fig. (5) (b). Sketch map of merging line segment.

4.3. Recognition of Circular Arc

The obtained short line segments by implementing the above strategy are merged into the long line segments. These long line segments are fitted into circular arcs by means of the perpendicular bisector tracing algorithm [11]. The basic idea of the algorithm is that the perpendicular bisector of any chord on a circle passes through the circle center, and every endpoint of the chord is at an equal distance from the circle center.

As illustrated in Fig. (**6a**), any obtained line segment may be an approximation of an arc. We first consider the obtained line segment L_0 as seed arc, its perpendicular bisector P_0 is constructed. For every line segments L_i (i =1, 2…N), we then construct its perpendicular bisector P_i, and calculate the intersection C_i of P_i and P_0. Due to errors, the intersection C_i is not unique. However, all C_i are located along the line P_0. We define the point C_{min} is the nearest point from L_0, and the point C_{max} is the farthest point from L_0. Therefore, a rectangle *C* is obtained, whose length is the distance between C_{min} and C_{max}, and whose width is equal to the length of L_0 and is bisected by P_0. The rectangle *C* is the area within which the potential arc center is sought.

Having determined the potential center area, we check every pixel within it as a center candidate $c(x, y)$, the distances $R_i(x, y)$ between $c(x, y)$ and endpoints of line segments are gained. Then, we calculate the average radius $R(x, y)$ and the mean square error $ASD(x, y)$, using (8) and (9), respectively.

$$R(x,y)=\frac{1}{N+1}\sum_{i=0}^{N}R_i(x,y) \tag{8}$$

$$ASD(x,y)=\frac{1}{N+1}\sum_{i=0}^{N}(R(x,y)-R_i(x,y))^2 \tag{9}$$

The point whose $ASD(x, y)$ is minimal is taken as the circle center of circular arc, and the corresponding $R(x, y)$ is the radius of circular arc.

The long circular arc may be recognized as some short circular arcs by using the perpendicular bisector tracing algorithm, so we merge these short circular arcs into the long circular arc. As shown in Fig. (**6b**), the merging condition of two adjacent short circular arcs: the coordinate difference between the circle centers O_1 and O_2 must fall within the prescribed limits (5 is fine); the radius difference between the radius R_1 and the radius R_2 should be less than the specified value (also 5 is enough); and the angular difference between the ending angleα_2 of the first circular arc and the starting angle α_3 of the second circular arc should be less than the threshold value (the threshold value is 3°). If there are two circular arcs meeting the above-mentioned three conditions, they can be merged into a new longer circular arc. The new circle center coordinates of this longer arc can be defined by the average value of the circle center coordinates of the two shorter arcs, and the radius of this new longer arc can be replaced by the average value of the radius of the two shorter ones. Also, the starting angle and ending angle of this new arc are substituted with the starting angleα_1 of the first arc and the ending angleα_4 of the second arc, respectively.

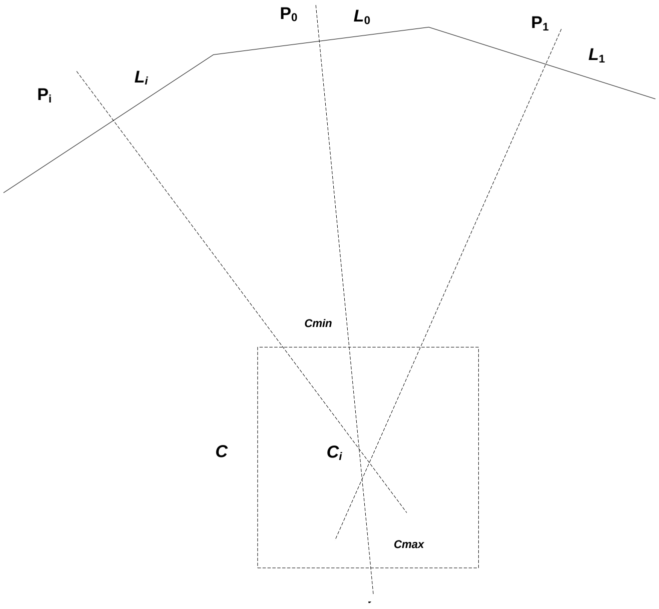

Fig. (6) (a). Sketch map of circular arc fitting.

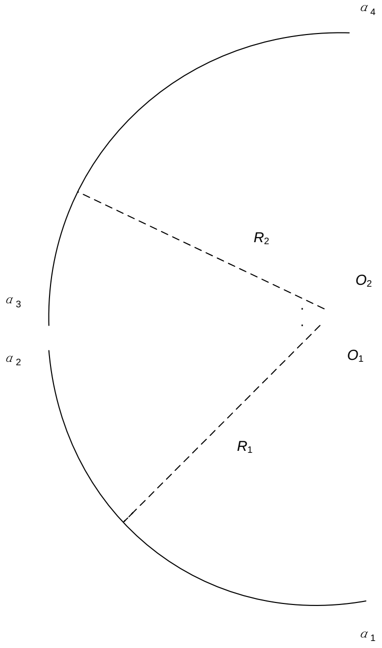

Fig. (6) (b). Sketch map of merging circular arc.

5. EXPERIMENTAL RESULTS AND EVALUATION

5.1. Recognition of Circle

In this section, the performance of the proposed algorithms has been tested with the real industrial CT images. The algorithms are implemented in C++ with MS Visual C++ 6.0 complier on a desktop PC with AMD 64 processor 1.81GHZ and 512MB RAM. The vectorization system for industrial CT image is designed, which provides the platform for our experiments. Fig. (**7a**) is an original industrial CT image of a socket set by the ICT system of CD-650BX in Chongqing University ICT Research Center. The technical indexes of the ICT system are as follows: the ray source is a 6/9MeV electron linear accelerator, the spatial resolution is 2.0lp/mm, the density resolution is better than 0.5%, the diameter of field of view is 398.872mm, the size of image is 800pixel*800pixel. Fig. (**7b**) is the sub-pixel edge image of circles in Fig. (**7a**); Fig. (**7c**) is the probability map of circles in Fig. (**7a**).

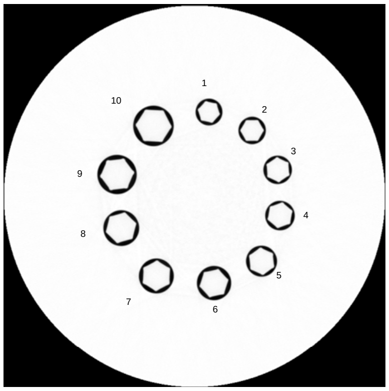

Fig. (7) (a). Industrial CT image of a socket set.

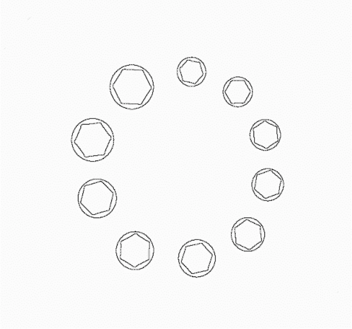

Fig. (7) (b). Sub-pixel edge image.

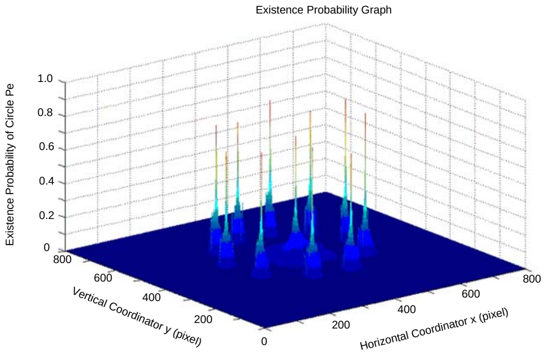

Fig. (7) (c). Probability map of circles.

5.2. Measurement of Circle Parameters

The results of our algorithms on the industrial CT image of the socket set are summarized in Table **1**. While in Table **2**, we list the absolute error *e* between the true and measured values of a parameter, and we evaluate the relative error e_r as well. As can be seen from Table **2**, our algorithms are able to achieve good accuracy. The absolute errors *e* of radius are less than 0.10mm, and the relative errors e_r are less than 0.5%.

Table 1. Measurement Results of Circles

Number	Circle Center (pixel)	Radius (pixel)	Radius (mm)	P_e
1	(430, 576)	27.798	13.860	0.867
2	(520, 538)	28.452	14.186	0.919
3	(573, 455)	29.520	14.718	0.908
4	(578, 360)	30.635	15.274	0.774
5	(540, 264)	32.172	16.040	0.870
6	(440, 219)	35.642	17.770	0.785
7	(320, 233)	36.254	18.076	0.847
8	(246, 333)	37.170	18.533	0.729
9	(237, 445)	40.871	20.378	0.841
10	(314, 547)	42.133	21.007	0.877

Table 2. Measurement Errors of Circles

Number	True Value (mm)	Measured Value (mm)	Error (mm)	Relative Error (%)
1	13.925	13.860	-0.065	0.467
2	14.240	14.186	-0.049	0.379
3	14.780	14.718	-0.062	0.419
4	15.350	15.274	-0.076	0.495
5	16.075	16.040	-0.035	0.218
6	17.810	17.770	-0.040	0.225
7	18.125	18.076	-0.049	0.270
8	18.590	18.533	-0.057	0.307
9	20.420	20.378	-0.042	0.206
10	21.230	21.007	-0.023	0.109

5.3. Vectorization Results of Whole industrial CT image

In addition, we evaluate the vectorization performance on two complicated images shown in Figs. (**8a** and **9a**). Fig. (**8a**) displays an original industrial CT image of a car engine. Based on facet model, its edge image is extracted and shown in Fig. (**8b**). As seen from the edge image, the contour is clear and intact. Also, all small contours are extracted correctly. Fig. (**8c**) is the vector graph of the car engine by our algorithms; it is edited and perfected in AutoCAD2008. In this experiment, a threshold is set for determining whether the closed graphic element is circle. As observed in Fig. (**8c**), setting the threshold is 0.7, our algorithms not only detect these four large circles correctly, but also fit them with good accuracy. The sizes of circles are marked in AutoCAD2008. Nevertheless, the small circles and slight contours can't be detected completely. If we decrease the threshold value appropriately, these small circles and slight contours will be detected. But the vectorization accuracy may be dropped.

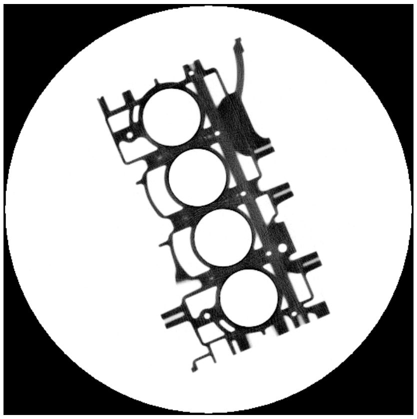

Fig. (8) (a). Industrial CT image of car engine.

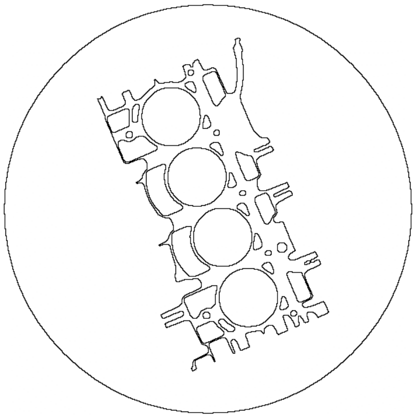

Fig. (8) (b). Sub-pixel edge image of car engine.

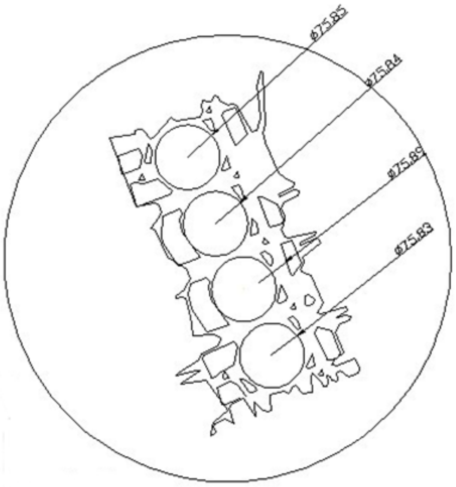

Fig. (8) (c). Vector graph of car engine.

Fig. (**9a**) shows an original industrial CT image of a gearbox cover. Unlike the previous industrial CT image of the car engine, this image has relative complex contours. In the top left corner are several small contours, which are difficult to be detected correctly. Despite this, its edge map is extracted completely and shown in Fig. (**9b**). Of particular interest is the vector graph of the gearbox cover in Fig. (**9c**) which evaluates our algorithms, and it is edited in AutoCAD2008. As shown in this picture, all circles are detected. Furthermore, several small circles are also detected and fitted correctly. Notice that the left isolated circle, which has the inner and outer concentric contours, is detected correctly. Also, the line contour can be fitted with ideal precision. However, Fig. (**9c**) also shows the limitation of our algorithms in the slight and complicated details of image. The circle with diameter of 16.55mm is nonexistent, it should not be found. And some contours are not with respect to the original contours ideally. For instance, the circular arc is represented by the irregular arc segments and curves. The following work will aim to resolve these problems.

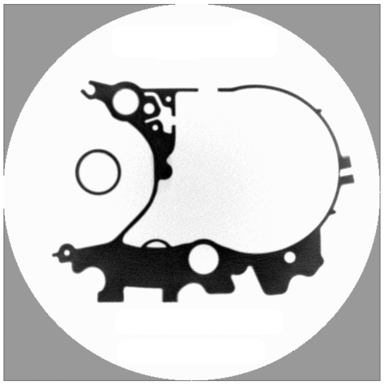

Fig. (9) (a). Industrial CT image of gear box cover.

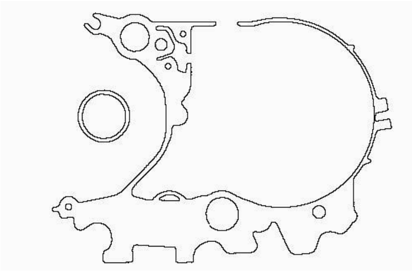

Fig. (9) (b). Industrial CT image of gear box cover.

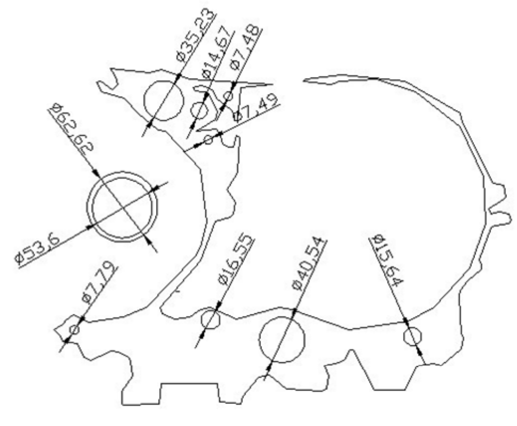

Fig. (9) (c). Industrial CT image of gear box cover.

6. CONCLUSIONS

In this paper, an algorithm for sub-pixel edge detecting based on facet model is adopted for the pre-processed industrial CT image. For the obtained contour, we can calculate the parameters of circle using the improved fast algorithm based on probability of existence map. Then, a set intersection algorithm of fitting a straight line is applied for recognizing the line. Finally, use the perpendicular bisector tracing algorithm and the least squares function to recognize circular arc. Thus, a vectorization system of industrial CT image is designed, which provides the platform for our experiments.

Experimental results show indeed that our algorithms are capable of recognizing the circle, line and circular arc with an excellent accuracy. Furthermore, the vectorization performance for the whole image is preferable. It can satisfy the industrial CT image vectorization requirements of higher precision, rapid speed and non-contact. In the future, we will focus on extending this work by recognizing the relative complex graphic elements as ellipse, regular polygon.

ACKNOWLEDGMENTS

This research has been supported by the National High Technology Research and Development Program 863 (Project No. 2006AA04Z104). All industrial CT images are provided by ICT Research Center of Chongqing University.

REFERENCES

[1] F. L. Liu, G. F. Qiao and B. Zou, “Precise measurement of circles in industrial computed tomographic images”, *Optics Precision Eng.*, vol. 17, no. 11, pp. 2842-2848, 2009.

[2] N. Y. Lee, S. H. Jung and J. B. Kim, “Evaluation of the measurement geometries and data processing algorithms”, *Appl. Radiat. Isot.*, vol. 67, pp. 1441-1444, 2009.

[3] B. Cheng, S. S. Zhang and Y. F. Shi, “Research on Vectorization and Recognition for Engineering Drawings: a Survey”, *J. Eng. Graph.*, vol. 29, no. 6, pp. 7-12, 2008.

[4] V. Nagasamy and N. A. Langrana, “Engineering drawing Processing and vectorization system”, *Comput. Vis. Graph. Image Proc.*, vol. 49, no. 3, pp. 379-397, 1990.

[5] J. W. Tian, Y. Shu, D. P. Tian and Y. X. Huang, “Image sub-pixel feature location algorithm on vision measurement system”, *Chin. J. Mech. Eng.*, vol. 42, no. 11, pp. 75-80, 2006.

[6] S. Zheng, J. W. Tian and J. Liu, “Efficient facet-based edge detection approach”, *Opt. Eng.*, vol. 44, no. 4, pp. 1-10, 1990.

[7] Y. C. Zhang, H. M. Wang, Z. Z. Liang, M. Tan, W. B. Ye and B. Lian, “Existence Probability Map Based Circle Detection Method”, *Comput. Eng. Appl.*, no. 29, pp. 49-51, 2006.

[8] F. L. Liu and G. F. Qiao, “A Fast Algorithm of Circle Detection for Industrial CT Images Based on Probability of Existence”, *China Mech. Eng.*, vol. 19, no. 19, pp. 2335-2339, 2008.

[9] Q. H. Li, D. Q. Wang and Z. Y. Chen, “Research on Image Vectorization Methods and Algorithms”, *Appl. Res. Comput.*, no. 6, pp. 9-12, 1995.

[10] H. Y. Guan, J. Q. Zhang, Q. Hu and L. Zhong, “Line Extraction of industrial parts based on least square template matching,” *Proc. - Int. Congr. Image and Signal Processing (CISP 08)*, IEEE Press, July 2008, pp. 696-700.

[11] D. Dori and W. Y. Liu, “Stepwise Recovery of Arc Segmentation in Complex Line Environment”, *Int. J. Doc. Anal. Recog.*, vol. 1, no. 1, pp. 62-71, 1998.

Received: November 26, 2010 Revised: December 29, 2010 Accepted: January 07, 2011