An Improved Hough Transform for Circle Detection using Circular Inscribed Direct Triangle

Qiang Li* Mingyun Wu

School of Computer Science and Technology
Hangzhou Dianzi University
Hangzhou, China
hzlee@hdu.edu.cn

Abstract—**Considering that the classical Hough transform for circle detection has poor real-time performance because of massive computation and memory consumption caused by accumulative voting in three-dimensional parameter space, this paper presents an improved Hough circle detection algorithm using circular inscribed direct triangle. The algorithm reduces the three-dimensional parameter space needed by classical Hough transform to two-dimensional parameter space by utilizing the geometrical characteristics of circle. By traversing an image's pixels only once, the coordinate points which may be centers of candidate circles are voted, and the radius corresponding to the coordinate points whose number of votes exceeds the user-defined threshold is taken out. Finally, the false circles are eliminated, and all real circles in the image are detected. The experimental results show the algorithm has fast speed, high accuracy and good anti-noise performance in contrast with the classical Hough circle detection method.**

Keywords-Hough transform; Circle detection; Image segmentation

I. INTRODUCTION

Circle detection is a very important task in computer vision. In the process of industrial production, circle detection can check whether the appearance and quantity of products meet the quality requirements. In daily life, human head contour can be regarded as a circular image, and circle detection can also realize crowd monitoring.

In short, circle detection is very common in industrial production and daily life. However, how to quickly and accurately detect the circles in an image is a problem. A frequently-used method is the Hough transform. Hough transform is an algorithm for circle detection proposed by Paul Hough in 1962 [1]. Its basic idea is to transform the image from original image space to parameter space. In the parameter space, a parameter form that most boundary points satisfied is used as the description of the curve in the image. It accumulates the parameters by setting an accumulator, and the point corresponding to the peak value is the required information. The biggest advantage of Hough transform is that it is insensitive to noise. Its main shortcomings are large amount of calculation, large amount of memory and poor real-time performance. Therefore, Hough transform is not suitable for the occasion where the real-time requirement of the system is high.

In order to improve the performance of the Hough transform, Zhang et al. proposed Chord Midpoint Randomized Hough Transform for the Cell Image Segmentation [2]. The algorithm performs circle detection by accumulating the midpoint of the chord based on the fact that the inscribed circle of all points on the circle must pass through the center of the circle. However, since all the circular information is obtained by calculating the midpoint of the chord, if the edge of circle to be detected is not completely enough, the calculation error of the midpoint of the chord is large. Luo et al. proposed a fast and robust circle detection method using perpendicular bisector of chords [3]. The method uses the vertical bisector pairs of two non-repeating chords to calculate the intersection of the centers of the circles. Because the position and radius of the center of the detected circle need to be separately determined, it is necessary to vote twice to detect the circle, so the algorithm needs to be improved in time performance. Makoto et al. proposed circle detection based on arc search using a table of virtual circle [4]. The method detects a circle on a virtual circle generated by a pair of edges on a horizontal or vertical scan line by detecting edge gradient information such as edge gradient size and edge gradient direction. Since the algorithm needs to establish a virtual circle table, it will cause large memory resources consumption and time consuming, and parallel computing is needed to improve the efficiency of the algorithm. Cuevas et al. proposed circle detection on images using learning automata [5]. The algorithm is based on a learning automaton and is a probabilistic optimization method that uses three non-collinear point codes as candidate circles on the edge image to detect the circle. Because the algorithm is a method based on probability optimization, if the image contains more circles to be detected, it may cause leak detection. Huang et al. proposed circle detection using a spiking neural network [6]. The algorithm trains the prediction model by constructing a multi-layered neural network to achieve circle detection. However, due to the different acceptance domains of different images, the algorithm is currently unable to switch between different accepted domains in response to different images. Xing Chen et al. proposed a new concentric circle detection method based on Hough transform [7]. The detection efficiency is enhanced by image discretization and reduced resolution ratio in the process of circle center detection, and proves that the circle center is on the gradient line of circle edge points. Meanwhile, the radius detection accuracy is improved by merging the similar radius

978-0-7381-0545-1/20/$31.00 ©2020 IEEE

in the process of radius detection. However, this algorithm is mainly used to detect concentric circles, and the detection efficiency for a single circle is inefficient. Cuevas et al. proposed fast algorithm for multiple-circle detection on images using learning automata[8]. The algorithm uses a combination of three edge points as parameters to determine circles candidates. A reinforcement signal determines whether such circle candidates are actually present at the image. Guided by the values of such reinforcement signal, the set of encoded candidate circles are evolved using the learning automata so that they can fit into actual circular shapes over the edge-only map of the image. The method proposed in this paper requires the using the learning automata, so it needs to use a large number of samples for training first, and the algorithm complexity is high. This paper presents an improved Hough transform algorithm for circle detection using circle inscribed direct triangle. The algorithm utilizes the geometric properties of circle and only needs to traverse the image once to get the centre points and radiuses of all circles in the image. It solves the problems of large computation, large memory consumption and poor real-time performance of classical Hough transform. At the same time, the verification is carried out with multiple images. The results show that the algorithm has a great improvement in the efficiency of circle detection.

II. HOUGH TRANSFORM

The classical circle detection algorithm is based on the Hough transform [9-10]. There are three unknown parameters in the circular Hough transform, namely the center coordinates *(a, b)* and the radius r. Therefore, it is necessary to triple-integrate all the edge points of the circle to solve a standard circle[11]. The problem of large computation of Hough transform becomes a bottleneck restricting the algorithm, which leads to the Hough transform being rarely used in the actual process[12]. In Cartesian coordinate system, the equation for the circle is shown in Equation (1).

$$(x-a)^2+(y-b)^2=r^2 \tag{1}$$

In the parameter space, the parameter equation of the circle is shown in Equation (2).

$$(x_i-a)^2+(y_i-b)^2=r^2 \tag{2}$$

The center coordinates of the circle are *(a, b)*, and the radius of the circle is *r*. At this time, each edge point (x_i, y_i) on the circumference corresponds to a three-dimensional cone in the parameter space *(a, b, r)*[13]. Assuming that *n* points are selected on the circle, there will be n three-dimensional cones in the parameter space, and the positions of these cone intersections are the center and radius *(a, b, r)* in the image space, as shown in Figure 1.

The parameter space is quantized to obtain a three-dimensional accumulation matrix for recording *(a, b, r)*[14]. Firstly, calculate all the points (a_0, b_0) whose distance from each point on the edge is *r*, and then record (a_0, b_0, r). Change the size of *r* value and then repeat the above calculation process. After calculating all edge points, all data's values are sorted accumulatively, and a set of data with the biggest value are considered to be the center coordinate and radius of the circle in the image space.

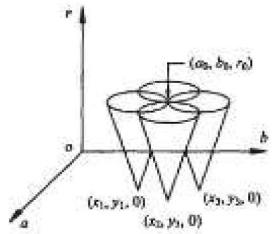

Figure 1. Parameter image of the circle

III. IMPROVED HOUGH TRANSFORM ALGORITHM FOR CIRCLE DETECTION

A. *Algorithm Thought*

Select the circle edge part in the second quadrant to enlarge, and the result is shown in Figure 2. The visible circle boundary is composed of a series of pixels, and the points in the second quadrant have a common feature, that is the gray value of the point is 0, and the gray value of the adjacent point at the upper left of the point is 255. This feature can determine whether a point is within the second quadrant of the target circle. The other quadrants are similar in point judgment method, which can quickly determine whether a point in the image is within a quadrant of the target circle.

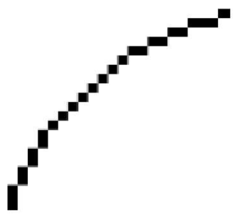

Figure 2. Enlarge figure of second quadrant internal circle edge

Using the above method, we firstly find a point $A(x_1, y_1)$ in the second quadrant, then find a point $B(x_2, y_2)$ in the horizontal right direction, find the point B and then find a point $C(x_3, y_3)$ in the vertical direction, as shown in Figure 3. The $\triangle ABC$ which is composed of three points *A,B,C* is the inscribed right triangle of the circle. From the nature of the circle, because the edge *AB* and the edge *BC* are perpendicular to each other, the edge *AC* must pass through the center of the circle and be the diameter of the circle, and the midpoint of the edge *AB* and the edge *BC* is the coordinate of the center of the circle. The solution formula for the center and radius is shown in Equation (3), *(a, b)* is the center coordinate, and *r* is the radius of the circle.

$$\begin{aligned} a &= (x_1+x_3)/2 \\ b &= (y_1+y_3)/2 \\ r &= \sqrt{(x_1-x_3)^2+(y_1-y_3)^2}\,/2 \end{aligned} \tag{3}$$

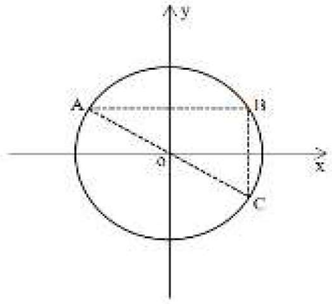

Figure 3. Circle inscribed direct triangle

Establish a two-dimensional voting matrix unit, and vote on the coordinate points that may be the center of the circle after traversing the image. If the value of votes of a certain coordinate point exceeds the set threshold, the coordinate point is regarded as the actual center coordinate, and then the radius of the circle corresponding to the coordinate point is extracted from the storage space.

B. Algorithm Steps

The algorithm requires three user-defined parameters to control its execution: validating threshold v of votes, minimum distance d between two detected circles, and radius range [r_{min}, r_{max}] of detected circles. The algorithm executes by following steps:

1) Define an accumulator unit Array(a, b) and radius storage unit Array(r).

2) Traverse the image from top to bottom from left to right. For every pixel.

a) Find a pixel with a gray value of 0 in the image, denoted as A(x_1, y_1), and determine whether the gray value of the pixel point (x_1-1, y_1-1) is 255.

b) If the condition is satisfied, search for a point B(x_2, y_2) in the horizontal direction to the right, which the gray value of the point is 0 and the grayscale value of the pixel (x_2+1, y_2-1) is 255.

c) If the point B exists, search downward from the point B for the point C(x_3, y_3) in the vertical direction, which the gray value of the point C is 0 and the pixel point (x_3+1, y_3+1) 's gray value is 255.

d) If the point C exists, consider there may be a circle, then calculate the center coordinates (a, b) and the radius r of the circle according to Equation (3).

e) Check whether r is in the interval [r_{min}, r_{max}], if the condition is satisfied, it is stored in the accumulator unit Array(a, b), and the accumulator unit value is added by 1.

3) Clean up the interference circles. Each center coordinate (a_i, b_i), with the accumulator unit value $v_i \geqslant v$ in Array(a, b), is considered to be an actual circle, and then all the center coordinates (a, b) of $d \geq \sqrt{(a-a_i)^2+(b-b_i)^2}$ *in the center coordinate range are deleted.*

4) Get the radius r corresponding to the center coordinate of each real circle from Array(r).

C. Algorithm Improvement and Performance Analysis

The algorithm uses the geometrical characteristics of a circle to find its inscribed direct triangle by finding the edge points of the circle, and then calculates the center coordinate and radius of the circle. A two-dimensional voting matrix is constructed. The real circles are obtained by cumulative voting on the center coordinates of the candidate circles. At the same time, the interference circles are eliminated according to the minimum distance condition. Finally, the radius corresponding to the center coordinate of the circle is taken out from the storage space. Since the classical Hough transform algorithm for circle detection needs to vote for both the center coordinate and the radius, the time complexity is $O(n^3)$. The improved algorithm proposed only needs to vote on the center coordinate of the circle, and the time complexity is $O(n^2)$, so the algorithm is more efficient. There is a great improvement. The disadvantage of this algorithm is that if the pre-processing effect is not ideal, it will lead to excessive missing edge of circle or more interference information, which will result in some errors in the missed detection or detected center and radius.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

All experiments in this paper were carried out on a PC. The configuration of the PC was Inter Core i5 2.2GHz dual core with 8GB memory. The resolution of the images selected in the experiment was 320×240 pixels.

A. Validity Experiment

In order to verify that the algorithm can effectively detect circles in an image, two images are first generated in the experiment, one image contains multiple circles with various radius, as shown in Figure 4a, and the other contains circles, straight lines and rectangles, as shown in Figure 4b. Then we use the algorithm in this paper to detect circles of this two images. The experimental results are shown in Figure 4c and Figure 4d, and illustrate that the algorithm can effectively detect all circles in the images.

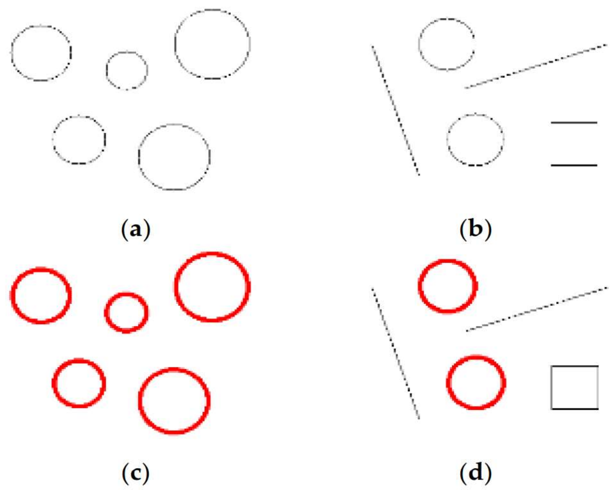

Figure 4. (a) Multiple circle image; (b) Mixed graphic image; (c) Multiple circle image detection result; (d) Mixed graphic image detection result.

B. *Anti-noise Experiment*

This experiment is to test the anti-noise performance of the algorithm. Add random salt and pepper noise to the image, as shown in Figure 5a. The experiment result is shown in Figure 5b, which shows that the proposed algorithm has good anti-noise performance and is robust to noise.

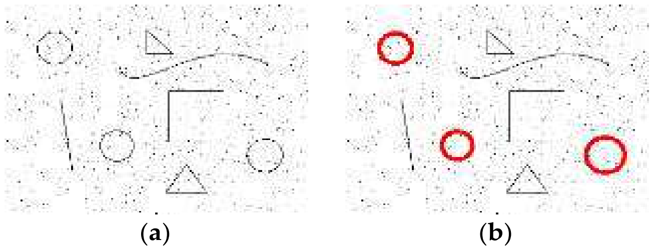

Figure 5. (a) Noise image; (b) Noise image detection results.

Figure 6.

C. *Practical Application Experiment*

The external appearance inspection of nylon package is a very important process in the nylon production, and image segmentation is the first step of the inspection. An image with several nylon packages can be divided into a number of images each containing only one nylon package for subsequent inspection.

In this paper, select three nylon package images for experiment, as shown in Figure 6a, 6b and 6c. Figure 6a contains white nylon packages with complete circle edge. Figure 6b contains black nylon packages with complete circle edge. Figure 6c contains some white nylon packages with partial circle edge.

For detecting outer contour of all the packages in Figure 6a, the parameters are set as v=15, d=80, $[r_{min}, r_{max}]=[40, 50]$. The experimental result is shown in Figure 6d, indicating the algorithm can accurately detect the position of the outer contour of all the packages in the image.

For detecting center axis of all the packages in Figure 6b, the parameters are set as v=40, d=50, $[r_{min}, r_{max}]=[20, 30]$. The experimental result is shown in Figure 6e, indicating the algorithm can accurately detect the position of the center axis of all the packages in the image.

For detecting outer contour of all the packages in Figure 6c, the parameters are set as v=30, d=60, $[r_{min}, r_{max}]=[25, 35]$. The experimental results are shown in Figure 6f, the algorithm can't detect the positions of the two leftmost packages because of partial missing of circle edge, and all the other packages with complete edges can be accurately detected.

D. *Performance Comparison Experiment*

Compared with the Classical Hough Transform(CHT), the experiment compared the time taken by the two algorithms to detect the circles in Figure 4a, Figure 4b, Figure 5a, Figure 6a, Figure 6b, Figure 6c. The experimental results are shown in Table I.

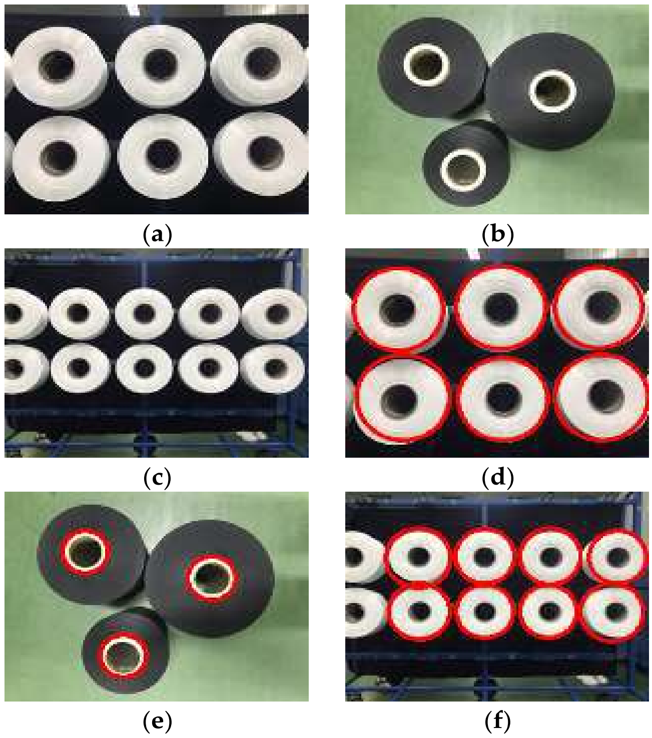

Figure 7. (a) White nylon package image with complete circle edge; (b) Black nylon package image with complete circle edge; (c) White nylon package image with partial circle edge; (d) Fig. 6(a) detection result; (e) Fig. 6(b) detection results; (f) Fig. 6(c) detection result.

TABLE I. TIME CONSUMPTION OF THE TWO ALGORITHMS

	CHT(s)	**This Paper(s)**
Figure 4a	4.323	0.262
Figure 4b	1.616	0.094
Figure 5a	1.401	0.086
Figure 6a	4.894	0.322
Figure 6b	2.184	0.134
Figure 6c	7.882	0.522

V. CONCLUSIONS

This paper focuses on the problems of classical Hough transform for circle detection, such as large time and memory computation, poor real-time performance, and proposed an improved Hough transform algorithm for circle detection using circular inscribed direct triangle. The algorithm utilizes the geometric properties of the circle, and the classical Hough transform needs to vote for the three-dimensional space to vote only in the two-dimensional space of the center coordinates. From the experimental results, the proposed algorithm greatly reduces the memory space and time consumption required by the algorithm. At the same time, the algorithm has good anti-noise performance and robustness to noise. However, it is found in the actual application process that the algorithm has high requirements on the pre-processing effect. If the pre-processing effect is not ideal, the missing edge of circle to be tested is too much or the interference information is too much, which may result in missed detection or detected center and the radius will have a certain error. The future research direction is to further improve the algorithm, so that the algorithm can accurately detect the circle to be measured when the pre-processing effect is not ideal.

ACKNOWLEDGMENT

This research is supported by 2017 MITT New Model Application Project of Intelligent Manufacturing for Zhejiang province, grant number 16.

REFERENCES

[1] Hough Paul, V.C. Method and Means for Recognizing Complex Patterns. U.S. Patent 3069654, 18 December 1962.

[2] Zhang, H.; Liang C.; Yan W. Chord midpoint randomized Hough transform for the cell image segmentation. Cross Strait Quad-regional Radio Science & Wireless Technology Conference. 2011, 2, 1446-1450.

[3] Luo, L.; Xu, D.; Zhang, Z. A Fast and Robust Circle Detection Method using Perpendicular Bisector of Chords. IEEE 2013 25th Chinese Control and Decision Conference. 2013, 2856-2860.

[4] Odagiri, M.; Onoguchi, K. Circle Detection Based on Arc Search Using a Table of Virtual Circle. 2016 International Conference on Digital Image Computing: Techniques and Applications. 2016.

[5] Cuevas, E.; Wario, F.; Zaldivar, D. Circle detection on images using learning automata. Iet Computer Vision. 2014, 6, 121-132.

[6] Huang, L.; Wu, Q.; Wang, X. Circle detection using a spiking neural network. Proceedings of the 2013 6th International Congress on Image and Signal Processing. 2013, 3, 1442-1446.

[7] Chen, X.; Lu, L.; Gao, Y. A new concentric circle detection method based on Hough transform. ICCSE 2012-Proceedings of 2012 7th International Conference on Computer Science and Education. 2012, 753-758.

[8] Cuevas, E.; Pérez-Cisneros, M.; Osuna-Enciso, V. Fast algorithm for multiple-circle detection on images using learning automata. IET Image Processing. 2012, 6, 1124-1135.

[9] Bosaghzadeh, A. Reducing circular hough transform parameters using morphological operations. 3rd International Conference on Pattern Analysis and Image Analysis. 2017, 56-61.

[10] Zhou, B. Using Vector Quantization of Hough Transform for Circle Detection. Proceedings-2015 IEEE 14th International Conference on Machine Learning and Applications. 2015, 447-450.

[11] Huang, S. Circular road sign detection and recognition based on hough transform. 2012 5th International congress on image and signal processing. 2012,1214-1218.

[12] Yadav. Approach to accurate circle detection: Circular Hough Transform and Local Maxima concept. 2014 International Conference on Electronics and Communication Systems.

[13] Zhou, B. Using Vector Quantization of Hough Transform for Circle Detection. IEEE International Conference on Machine Learning & Applications.

[14] Bosaghzadeh, A. Reducing circular hough transform parameters using morphological operations. 2017 3rd International Conference on Pattern Recognition and Image Analysis.