

An Improved Randomized Circle Detection Algorithm Using in Printed Circuit Board Locating Mark

Jingkun Liu¹, Qi Fan^{2*}

¹Chengyi University College, Jimei University, Xiamen, China

²Xiamen Sinic-Tek Intelligent Technology Co., Ltd., Xiamen, China

Email: liujingkun@126.com, *qi.fan@sinictek.com

How to cite this paper: Liu, J.K. and Fan, Q. (2019) An Improved Randomized Circle Detection Algorithm Using in Printed Circuit Board Locating Mark. *Applied Mathematics*, 10, 848-861.

<https://doi.org/10.4236/am.2019.1010061>

Received: September 14, 2019

Accepted: October 20, 2019

Published: October 23, 2019

Copyright © 2019 by author(s) and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

This paper presents an improved Randomized Circle Detection (RCD) algorithm with the characteristic of circularity to detect randomized circle in images with complex background, which is not based on the Hough Transform. The experimental results denote that this algorithm can locate the circular mark of Printed Circuit Board (PCB).

Keywords

Circle Detection, Randomized Algorithm, Characteristic of Circularity, Printed Circuit Board

1. Introduction

Detecting circles from a digital image is very important in image processing [1] and computer vision [2]. This problem has extensive application value in engineering such as product inspection and assembly, traffic monitoring, robot vision, face recognition, vectorization of hand-sketched drawing. For example, it plays an important role in the production of Printed Circuit Board (PCB) to achieve automatic positioning of PCB and to locate the reference point named as fiducial mark. Fiducial marks are used to locate the position of all features on PCB. In recent years, two main research directions are how to improve the accuracy and reduce the computation performance.

As the most well-known approach for circle detection, Hough Transform (TH) [3] has gained the widespread interest from researchers. Standard Circle Hough Transform (CHT) [4] has been shown with possessing robustness in dealing with noisy images. Set (x, y) be an edge pixel on a circle with center

coordinates (a, b) and radius r , the circle can be represented by

$$(x-a)^2 + (y-b)^2 = r^2. \quad (1)$$

Each edge pixel (x, y) in the image can be mapped into a conic surface in the three-dimensional (a, b, r) -parameter space. Although the technique can regularly achieve a relatively high degree of accuracy, there are some influencing factors such as huge demand of memory space for the accumulator and high computational complexity due to the time-consuming voting procedure. Several HT-based methods have been developed to overcome these problems. Randomized Hough Transform (RHT) [5] relative to CHT has faster processing speed and smaller storage requirement. RHT uses three noncollinear edges to deal with the three parameters (a, b, r) of Equation (1). This method randomly selects three edge pixels in the image with equiprobability every time, and the corresponding mapped pixel in the three-dimensional parameter space is collected. RHT performs well in high quality image, but low performance in noisy image.

In place of creating an accumulator array for mapping the extracted edge pixels in images to the circle parameters in HT-based method, Randomized Circle Detection (RCD) [6] does not use an accumulator for saving the information of related parameters in Randomized Sample Consensus (RANSAC) [7] based method. The main concept is that the algorithm randomly chooses four edge pixels from the image first, and then uses a distance criterion to determine whether they belong to a possible circle in the image. After finding a possible circle, RCD uses an evidence-collecting step to further determine whether the candidate circle is a real-circle. Since RCD does not need extra accumulator storage, the memory requirements needed in RCD are only a few variables, and the method has some other advantages such as real-time speed and more robust to noise. However, sampling for RCD randomly happens on all edge pixels of the whole image and verification of the hypothetical circles also use all the edge pixels, which both occupy a mass of time and obtain uncertainty of results [8]. To solve these problems, an improved randomized circle detection algorithm in the complex background image is proposed in the paper, which uses improved RCD algorithm and the characteristic of circularity. Firstly, the improved RCD based on connected contours is applied to detect possible circles. Then, the characteristic of circularity is used to discard some inaccurate possible circles. The algorithm is faster than RCD for it samples only on the connected curve [8]. The experimental results show that the refined algorithm has good detection performance.

In this paper, we search circles with an improved RCD algorithm in the complex back ground image, and use the characteristic of circularity to eliminate incorrect circles. Simulation results show that the proposed algorithm can locate circle mark effectively in PCB. The rest of this paper is organized as follows: Section 2 discusses the theory about normal Randomized Circle Detection (RCD). Section 3 introduces the improved Randomized Circle Detection in detail. Section 4 states the experimental results. Finally, some conclusions are given in Section 5.

2. Randomized Circle Detection (RCD)

This section describes the algorithm of standard RCD. Store all edge pixels $v_i = (x_i, y_i)$ to the set V . Any three noncollinear pixels $v_i = (x_i, y_i), i = 1, 2, 3$ can exactly determine one circle C_{123} . By Equation (1), we have

$$2x_i a_{123} + 2y_i b_{123} + r_{123}^2 - a_{123}^2 - b_{123}^2 = x_i^2 + y_i^2, \quad i = 1, 2, 3. \quad (2)$$

A representation of Equation (2) in terms of matrix form

$$\begin{pmatrix} 2x_1 & 2y_1 & 1 \\ 2x_2 & 2y_2 & 1 \\ 2x_3 & 2y_3 & 1 \end{pmatrix} \begin{pmatrix} a_{123} \\ b_{123} \\ r_{123}^2 - a_{123}^2 - b_{123}^2 \end{pmatrix} = \begin{pmatrix} x_1^2 + y_1^2 \\ x_2^2 + y_2^2 \\ x_3^2 + y_3^2 \end{pmatrix} \quad (3)$$

Applying Gaussian elimination and Cramer's rule, the center (a_{123}, b_{123}) can be obtained by

$$a_{123} = \frac{\begin{vmatrix} x_2^2 + y_2^2 - x_1^2 - y_1^2 & y_2 - y_1 \\ x_3^2 + y_3^2 - x_1^2 - y_1^2 & y_3 - y_1 \end{vmatrix}}{2(x_2 - x_1)(y_3 - y_1) - 2(x_3 - x_1)(y_2 - y_1)}, \quad (4)$$

$$b_{123} = \frac{\begin{vmatrix} x_2 - x_1 & x_2^2 + y_2^2 - x_1^2 - y_1^2 \\ x_3 - x_1 & x_3^2 + y_3^2 - x_1^2 - y_1^2 \end{vmatrix}}{2(x_2 - x_1)(y_3 - y_1) - 2(x_3 - x_1)(y_2 - y_1)}. \quad (5)$$

After obtaining the center (a_{123}, b_{123}) , the radius can be calculated by

$$r_{123} = \sqrt{(x_i - a_{123})^2 + (y_i - b_{123})^2}, \quad i = 1, 2, 3. \quad (6)$$

Let $v_4 = (x_4, y_4)$ be the fourth edge pixel, then the distance between v_4 and the boundary of the circle C_{123} denoted by

$$d_{4 \rightarrow 123} := \left| \sqrt{(x_4 - a_{123})^2 + (y_4 - b_{123})^2} - r_{123} \right|, \quad (7)$$

where $|z|$ denotes the absolute value of z . The four edge pixels $v_i = (x_i, y_i), i = 1, 2, 3, 4$ can obtain four circles $C_{123}, C_{124}, C_{134}, C_{234}$ with respect to four distances $d_{4 \rightarrow 123}, d_{3 \rightarrow 124}, d_{2 \rightarrow 134}, d_{1 \rightarrow 234}$. If $d_{i \rightarrow jk}$ is smaller than the given threshold T_d , these four edge pixels are co-circular and the circle C_{ijk} is a possible circle. If the four distances are all larger than T_d , RCD algorithm chooses the other four edge pixels.

Set a counter $C = 0$ for this possible circle in order to count how many edge pixels lie on the possible circle. For each edge pixel v_n in V , if $d_{n \rightarrow jk} \leq T_d$, the counter C will be incremented by one and v_n should be taken out of V ; otherwise the next edge pixel will be proceed to. Continue above process until all the edge pixels in V have been examined. In the evidence-collecting process, the final value of C is equal to n_p which is the number of edge pixels on the possible circle. If n_p is larger than the given global threshold T_g , the possible circle is claimed to be true. Otherwise, we discard the false circle and return those n_p edge pixels back into V .

In practical application, the following thresholds are used to search circle in

image:

T_f	The number of failures that can be tolerated. The running time must be finite, regardless of whether the correct results are obtained.
T_r	The ratio threshold of co-circle pixels. It is used to replace the global threshold
T_g	The number of pixels at the edge of circle, $T_g = 2\pi r T_r$, r is the radius of possible circle.
T_a	The threshold of distance. The distance between any two agent pixels of the possible circle should be larger than T_a .
T_d	The distance threshold for co-circle. It means that the fourth selected pixel is closed to the possible circle, which is determined by the other three pixels.
T_c	The number of circles that we want to detect in image.
T_{rl}	The minimal radius that the possible circle should fit.
T_{ru}	The maximal radius that the possible circle should fit.

The standard RCD algorithm has some advantages such as fewer memory requirements, faster speed and simpler algorithm than CHT or RHT.

3. The Improved RCD Algorithm

This section presents the improved algorithm consisting of the following. The general idea is to reduce the computational complexity and enhance the accuracy by extracting connected contours. It can discard some inaccurate possible circles by using characteristic of circularity.

3.1. Noise Suppression

In order to obtain higher recognition quality, an image preprocessing is used to suppress and eliminate noise from image. A Gaussian smoothing is a widely used to reduce image noise, which is represented by Gaussian filter [9]. Gaussian filter is a windowed filter of linear class with weighted nature, which is calculated according to Gaussian distribution. Gaussian filter in the most general function form of

$$G_{\sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}}, \quad (8)$$

in one-dimensional is able to fulfill this criterion optimally, where σ is the standard deviation of the Gaussian curve. In two-dimensions the Gaussian filter is defined by

$$G_{\sigma}(x, y) = \frac{1}{2\sigma^2\pi} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (9)$$

This filter can be separable by

$$G_{\sigma}(x, y) = G_{\sigma}(x)G_{\sigma}(y). \quad (10)$$

In order to remove small details from the image and fill small gaps in lines or

curves, the Gaussian filter should be used before other operations. The above mentioned Gaussian filter is used in continuous domain. In one discrete digital image, a discrete Gaussian low-pass filter should be used. The value of every pixel in the image will be replaced by the average of the intensity levels in the neighborhood defined by the filter mask. Because the smoothing process leads to “un-sharp” transition in intensities, smoothing filters have the undesirable side effect, which blur edges of object. In order to keep the balance between eliminating random noise and preserving sharpness of edges, the Gaussian filter with mask size 3×3

$$\frac{1}{9} \times \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix} \tag{11}$$

should be used in our experiment. It can be best seen by substituting the coefficient of the mask into the characteristic response R [10] with the sum of products as

$$R = \sum_{k=1}^9 w_k z_k = W^T Z, \tag{12}$$

where W and Z are nine-dimensional vectors formed from the coefficients of a 3×3 filter mark and the image intensities are encompassed by the mask respectively. Then we obtain

$$R = \frac{1}{9} \sum_{i=1}^9 z_i. \tag{13}$$

Figure 1 shows the results of applying a Gaussian smoothing in digital image. As expected from the Gaussian filter with mask size 3×3 , the detail of noise is blurred.

3.2. Edge Detection

The purpose of edge detection is to identify points in digital image at which the image brightness changes sharply. In the past 30 years, there are many methods for edge detection, but most of them can be classified into two categories, the Template Matching (TM) and the Differential Gradient (DG) [11]. In either

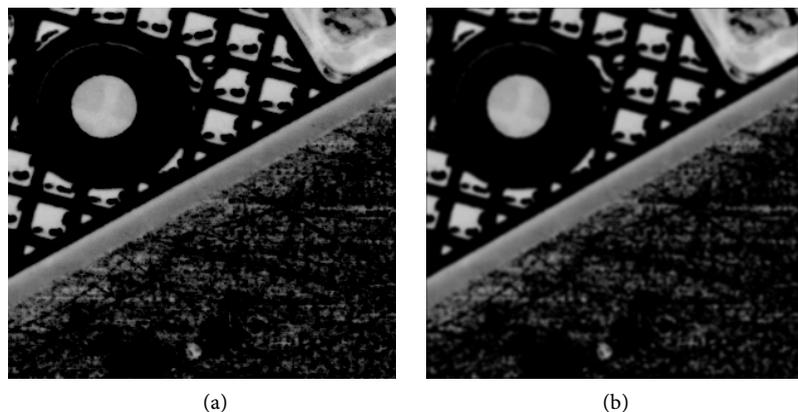


Figure 1. Gaussian smoothing. (a) Original image of size 307×307 pixels; (b) Result of smoothing using Gaussian filter.

case, the aim is to find where the intensity gradient magnitude g is large enough to be taken as a reliable indicator of the edge. In the DG approach, the gradient information is estimated vectorially. There are some well-known edge operators, such as Sobel edge operator, Canny edge operator, Robbers edge operator, Prewitt edge operator and so on. In practice, Sobel gradient operator is most frequently used according to its simplicity and effectiveness. The local edge magnitude may be calculated vectorially using the nonlinear transformation [11]

$$g = \sqrt{g_x^2 + g_y^2}, \quad (14)$$

In order to save computational effort, the approximate formula

$$g = |g_x| + |g_y|, \quad (15)$$

could be used in practice. **Figure 2** illustrates the edge segmentation, which are obtained from input original image **Figure 1(a)** and smoothed image **Figure 1(b)**. It is clear that **Figure 2(b)** shows fewer noisy in the segmentation.

3.3. Extracting Contours

In order to reduce the computation running time and enhance the accuracy, the contours are obtained by using software OpenCV. RCD will be executed in the connected contour to find possible circles. The computational complexity is reduced from all pixels to contours.

As illustrated in **Figure 3(a)**, an image with all contours from **Figure 2(b)** is generated. **Figure 3(b)** shows another more complex image with a different fiducial mark. All contours extracted from **Figure 3(b)** are shown in **Figure 3(c)**. The circle is divided into some broken edges. Every connected contour is drawn with its own particular color in images.

3.4. The Improved RCD Algorithm

As above mentioned, we describe how to determine possible circles according to

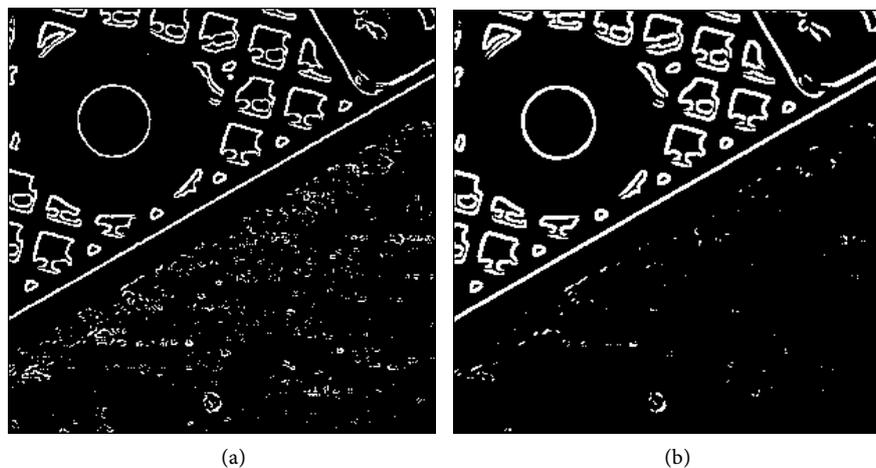


Figure 2. Edge segmentation. (a) Segmentation of **Figure 1(a)**; (b) Segmentation of **Figure 1(b)**.

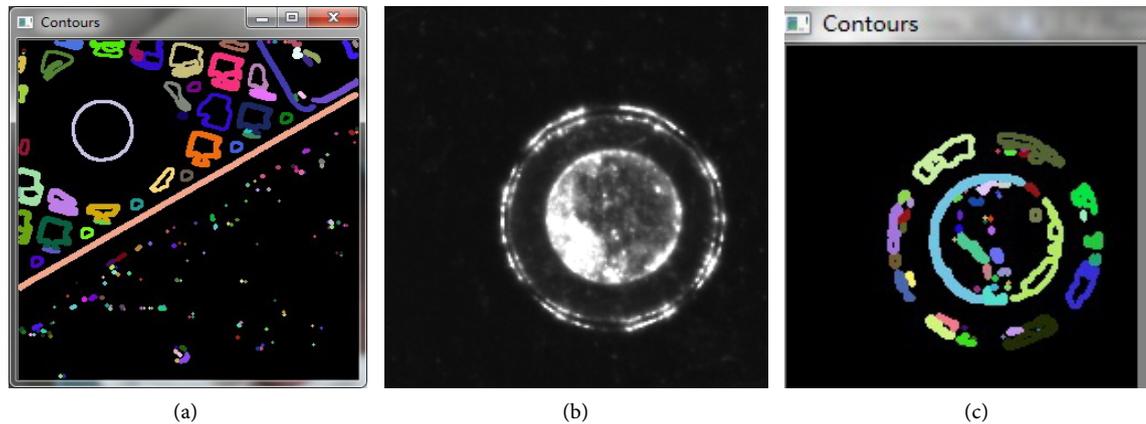


Figure 3. Connected contours. (a) Contours from **Figure 2(b)**; (b) Original image with size 195×195 pixels; (c) Contours obtained from (b).

the extracted contours. Let V denote the set of all edge contours in the image. The procedure randomly picks four pixels from each contour to make a decision whether the contour belongs to a possible circle based on the RCD algorithm. If the four pixels meet Equation (6) and Equation (7), all pixels from all contours will be collected. We continue the above process until the contour is determined as a possible circle or the global threshold T_f has been reached. The procedure proceeds to next contour and the global Threshold T_f is reset at the beginning.

Because the proposed algorithm samples only on the connected contours, it can overcome the interference between different objects and improve the accuracy of detection. The computational complexities are reduced and the execution-time is saved.

3.5. The Characteristic of Circularity

The above technique can detect all possible circles around the true fiducial mark. There exists a bias problem, because of noise and nonstandard circular circle. The circularity will be calculated to determine the best matching circle.

Usually the used form features can be grouped into two categories, the contour characteristic and the regional characteristic. The contour characteristic of image is mainly aimed at the boundary of image, and the regional characteristic of image is related to the whole shape region. The circular degree could be used to describe the circle. There are four circular degree measures [12], such as circularity, boundary energy, density and ratio of area to average distance quadratic sum. The characteristic of circularity is easy to be realized and more suitable for engineering application. It is needed to explain that the extraction of characteristic of circularity is based on image pre-processing and edge detection.

The characteristic of circularity is defined by all boundary pixels of the region D . An image may be defined as one two-dimensional function $f(x, y)$, and the amplitude of f at any pair of coordinates (x, y) is called the gray level of the image at the pixel. Suppose that the image is of size $N \times M$ elements. Set a pair of coordinates (x_k, y_k) of any pixel in the image. The barycentric coordinates

are defined as

$$\bar{x} := \frac{\sum_{i=1}^N \sum_{j=1}^M i \cdot f(i, j)}{\sum_{i=1}^N \sum_{j=1}^M f(i, j)}, \quad \bar{y} := \frac{\sum_{i=1}^N \sum_{j=1}^M j \cdot f(i, j)}{\sum_{i=1}^N \sum_{j=1}^M f(i, j)}. \quad (16)$$

The average distance from the regional barycentric to the boundary pixel is in the form of

$$\mu_D := \frac{1}{K} \sum_{k=0}^{K-1} \|(x_k, y_k) - (\bar{x}, \bar{y})\|, \quad (17)$$

where $\|(x_1, y_1) - (x_2, y_2)\| := \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$, and the mean square deviation of distance from the regional barycentric to the boundary pixel is in the form of

$$\delta_D := \frac{1}{K} \sum_{k=0}^{K-1} [\|(x_k, y_k) - (\bar{x}, \bar{y})\| - \mu_D]^2 \quad (18)$$

can be simplified to

$$\delta_D := \frac{1}{K} \sum_{k=0}^{K-1} [(x_k - \bar{x})^2 + (y_k - \bar{y})^2]^2 - \mu_D^2. \quad (19)$$

The calculation formula for the characteristic of circularity is defined by

$$C := \frac{\delta_D}{\mu_D}. \quad (20)$$

When the regional D tends to be a circle, the characteristic of circularity C is monotonically decreasing and tends to be infinitesimal. It's not affected by regional translation, rotation and variation of scales. It means that we can determine the position of the circle in the image by the minimum point of characteristics of circularity in local region.

3.6. Stability, Accuracy and Time Speed

There are still some shortcomings in the standard RCD. Because the standard RCD randomly picks four edge pixels each time. The search process ended when it found the true circle or reached the number of failures T_f . It means that the searching result may be quite different according to the different four starting pixels. If the starting pixels are close to the target circle, the circle may be detected rapidly. Sometimes, the start four pixels are too far from the target circle and the input image contains too many useful and useless pixels. The RCD algorithm may end without finding the true circle. In order to find possible circle rapidly, these thresholds T_d , T_r and T_a should not be too small. Then many incorrect circles would be found.

By suppressing noise and finding of the connected contours, which contains at least a certain number of points, it will reduce the calculation time and find fewer possible circles. At last, after verifying the circularity, the real circle will be selected. So the improved RCD provides the better stability and accuracy than

standard RCD. As the image grows larger and more complex, the time will become more longer. But the improved RCD will significantly reduce the time. The following experiment will be used to verify these conclusions.

4. Experimental Results

In this section, some experimental results are demonstrated to show the execution-time and accuracy advantages of our improved RCD algorithm, compared with standard RCD algorithm. Four original images are used to evaluate the performance of the proposed algorithm. The first three images with the same mark are shown in **Figures 4(a)-(c)**, which have different size 131×124 , 262×153 and 307×307 pixels. The last image **Figure 4(d)** has another different mark. The radius of the first mark R is 26.5 pixels and the radius of the second mark R is 36.5 pixels. **Table 1** describes all parameters used in our experiments. Some

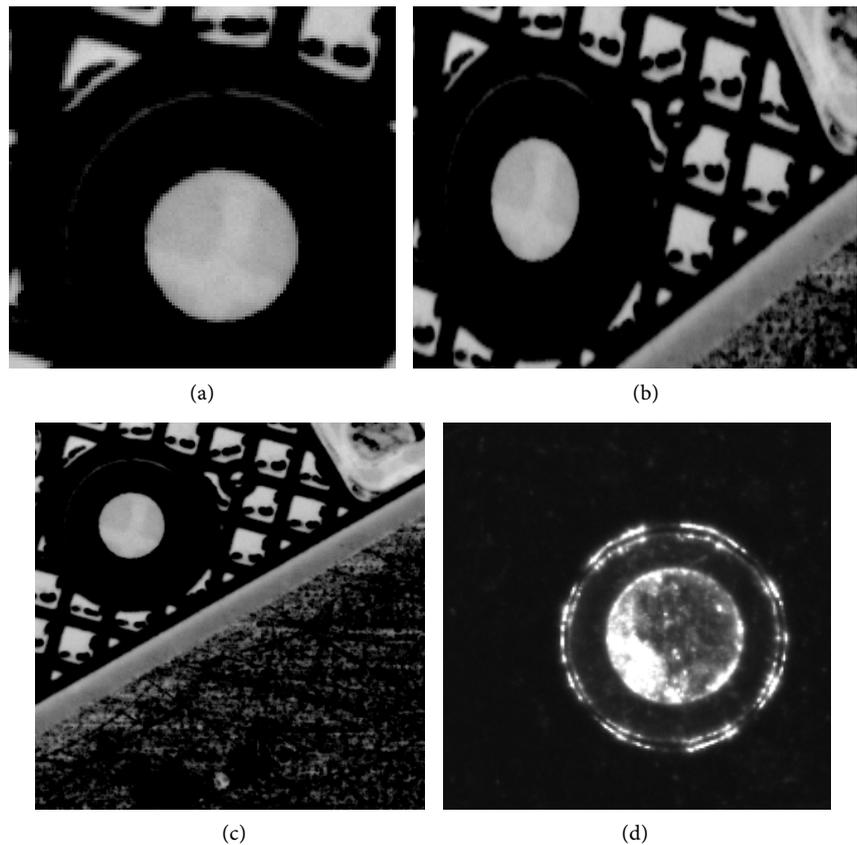


Figure 4. Four original images with different size and mark. (a)-(c) Original images with the same mark; (d) Original image with another different mark.

Table 1. All thresholds used in experiments.

	T_f	T_r	T_a	T_d	T_c	T_{rl}	T_{ru}	T_{cc}
Pure RCD	100,000	0.6	0.25R	1.5	20	$R - 5$	$R + 5$	1
Improved RCD	Square of the length of contour, which is different in searching circle in each chain	0.6	0.25R	1.5	20	$R - 5$	$R + 5$	1

additional variables are defined as following: P denotes the center pixel (x, y) of detected circle. R is the pre-input radius and r is the radius of detected circle. PN is the number of pixels on the contour of detected circle. C is the circularity of detected circle. Avg T (ms) represents the average running time in millisecond. T_{cc} is the threshold of circularity.

4.1. Locating Mark with Standard RCD Algorithm

In this part the marks will be detected by using standard RCD algorithm. As shown in **Figure 5** there are many correct and incorrect circles in images. In **Figure 5(c)** the true mark circle is not included in the detected possible circles. There is no way to discard the incorrect circles and find the true mark. According to the threshold T_c , there are twenty possible detected circles at most, which are drawn in **Figure 5**. In **Tables 2-5** only ten possible circles are described.

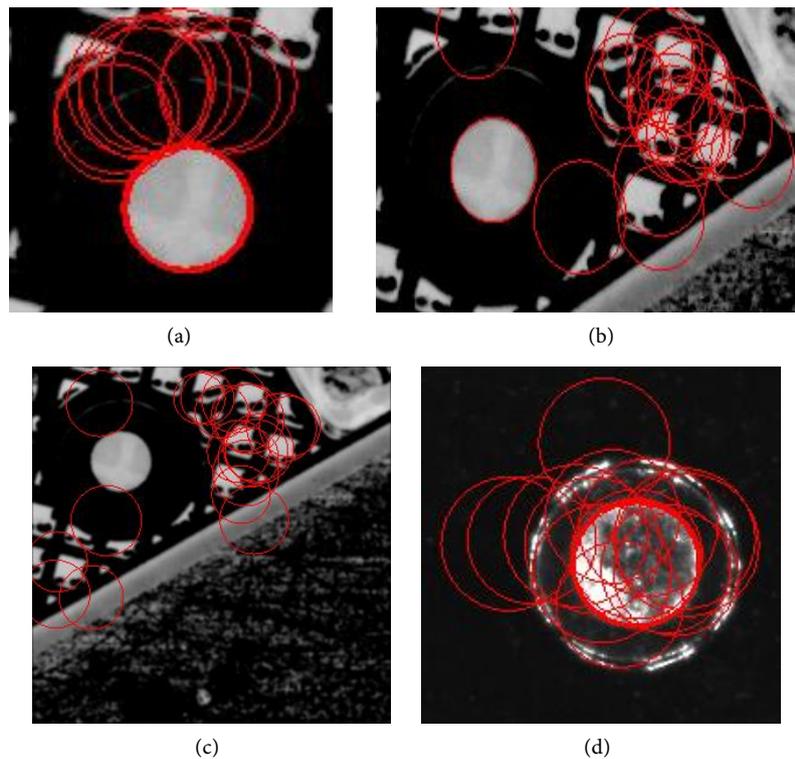


Figure 5. Locating results using standard RCD algorithm. (a)-(c) The detected results, obtained from **Figures 4(a)-(c)**; (d) The result obtained from **Figure 4(d)**.

Table 2. Ten results shown in **Figure 5(a)**.

P	r	PN	P	r	PN
(71.039, 81.479)	25.911	404	(70.902, 80.576)	26.225	369
(90.787, 30.734)	28.790	133	(75.500, 83.700)	28.452	138
(70.930, 79.679)	26.190	302	(45.153, 11.418)	27.853	58
(72.391, 81.334)	25.750	383	(71.071, 80.071)	25.014	320.000
(71.982, 81.718)	25.382	379	(62.561, 16.912)	23.763	66

Table 3. Ten results shown in **Figure 5(b)**.

P	r	PN	P	r	PN
(189.352, 66.432)	26.473	175	(157.550, 47.532)	26.822	170
(194.103, 72.685)	24.486	116	(195.472, 52.751)	24.591	166
(157.036, 36.878)	24.956	151	(223.636, 50.818)	26.623	106
(201.467, 86.570)	23.407	89	(185.626, 41.030)	23.896	143
(169.820, 65.472)	27.184	158	(206.284, 80.524)	27.742	67
(71.982, 81.718)	25.382	379	(62.561, 16.912)	23.763	66

Table 4. Ten results shown in **Figure 5(c)**.

P	r	PN	P	r	PN
(34.786, 163.67)	27.257	150	(252.645, 48.271)	28.241	122
(177.100, 43.500)	28.500	148	(103.150, 122.58)	23.413	74
(69.098, 179.443)	25.349	103	(205.563, 55.500)	26.300	137
(145.883, 32.260)	27.740	151	(167.003, 42.453)	26.043	204
(216.647, 44.956)	25.610	97	(148.500, 13.571)	27.472	119
(71.982, 81.718)	25.382	379	(62.561, 16.912)	23.763	66

Table 5. Ten results shown in **Figure 5(d)**.

P	r	PN	P	r	PN
(116.345, 107.928)	33.594	286	(115.135, 107.68)	34.296	318
(116.459, 106.745)	34.481	348	(137.832, 112.68)	33.834	102
(116.271, 107.794)	34.310	337	(116.256, 106.40)	33.951	309
(132.030, 109.909)	35.091	83	(107.256, 129.41)	34.519	101
(116.398, 101.856)	35.538	108	(116.355, 106.31)	34.913	325

4.2. Locating Mark with Improved RCD Algorithm

In the following experiments, the fiducial marks will be detected by our improved RCD algorithm and corresponding circularity. As illustrated in **Figure 6**, the left four images (a)-(d) denote the results, which are obtained without calculating minimal circularity. Because of unsharpness of edge, the correct and bias circles are detected, which have similar radius and position. It is difficult to determine the best true circle without additional decision condition. After calculating the minimal circularity, the true marks are located in the right four images (e)-(h). The detected possible circles are shown in **Tables 6-9**. Compared to the former standard RCD algorithm, the number of possible circles is fewer and the results are close to the true mark. By calculating minimal circularity, in **Table 6** the true circle with center pixel (71.500, 80.500) and radius 26.163 pixels is detected. In **Table 7** the true circle with center pixel (71.369, 81.622) and radius 27.132 pixels is detected. In **Table 8** the true circle with center pixel (75.601, 81.013) and

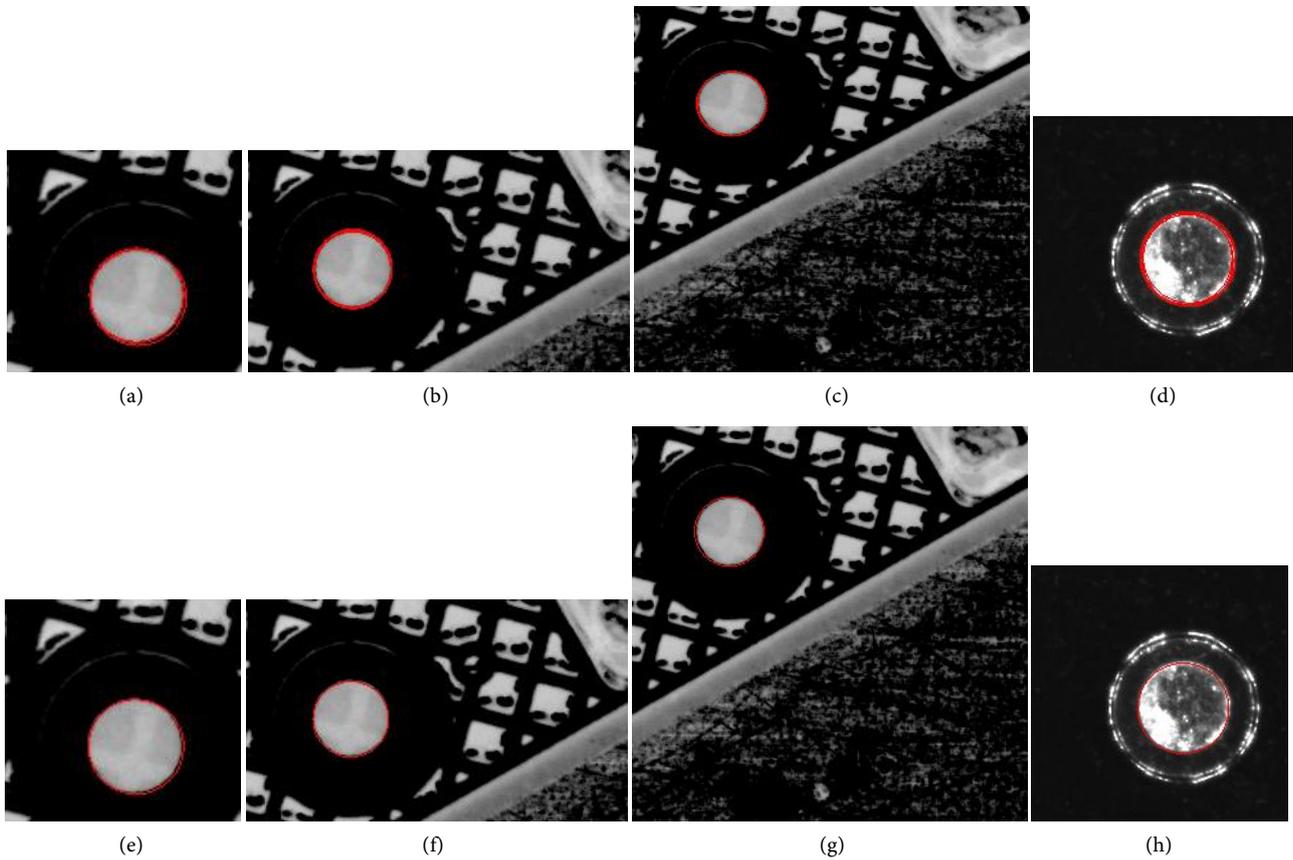


Figure 6. Results obtained by using our improved RCD algorithm and circularity. (a)-(d) Results obtained without calculating minimal circularity; (e)-(h) Results obtained by calculating the minimal circularity.

Table 6. Results shown in **Figure 6(a)**.

P	r	PB	c	
(72.313, 81.737)	26.795	309	0.8077	
(71.500, 80.500)	26.163	392	0.1288	Selected As true

Table 7. Results shown in **Figure 6(b)**.

P	r	PN	c	
(71.404, 82.005)	26.291	356	0.8077	
(71.369, 81.622)	27.132	253	0.1288	Selected As true
(71.194, 81.403)	27.016	272	0.2732	
(71.570, 80.779)	26.903	290	0.1548	

Table 8. Results shown in **Figure 6(c)**.

P	r	PN	c	
(74.985, 81.435)	27.104	257	0.6047	
(75.601, 81.013)	26.951	253	0.0339	Selected As true

Table 9. Results shown in **Figure 6(d)**.

P	r	PN	c	
(114.992, 108.471)	34.629	356	0.0595	Selected As true
(14.192, 107.192)	34.529	376	0.0985	
(116.826, 106.174)	34.699	461	0.2000	
(116.500, 108.500)	34.821	383	0.1547	

Table 10. Time performance comparison between standard RCD and the improved RCD in terms of milliseconds.

Original Images	Avg T (RCD)	Avg T (improved RCD)
Figure 4(a)	77.8 ms	121 ms
Figure 4(b)	466.8 ms	478.2 ms
Figure 4(c)	1172.8 ms	710 ms
Figure 4(d)	148.2 ms	145 ms

radius 26.951 pixels is detected. In **Table 9** the true circle with center pixel (114.992, 108.471) and radius 34.629 pixels is detected.

4.3. Comparing the Execution-Time

In order to compare the execution-time, all concerned experiments are performed on the Intel i5-5200U CPU with 2.20 GHz and 8 GB RAM. The adopted operating system is MS-Windows 7 and the programming environment is VS2010. In order to accurately evaluate the execution-time, we run each image 100 times and calculate the average time in **Table 10**. For locating mark in small images, the execution-time is almost the same for both algorithms. The time will be significantly reduced, when the mark is detected in a larger and complex image.

5. Conclusions

This paper has presented the proposed improved RCD strategy to improve the performance of execution-time and the accuracy of detection. The refined method can suppress the interference between different objects significantly. After calculating minimal circularity, the true mark will be located efficiently and accurately. The experimental results demonstrate that the proposed improved RCD improves significantly the accuracy of detection. Experimental results also demonstrate that the proposed algorithm provides a considerable execution-time improvement. The algorithm has significant execution-time superiority in large and complex image.

At the current stage, the quality of the image, such as the stability of the light source, the degree of image damage, clarity, etc., has a great influence on the calculation results. In the future, some normalization methods will be studied to reduce these interferences, such as combining with pattern recognition.

Founding

This article is supported by Science and Technology Project of Fujian Provincial Department of Education under contract JAT170917 and Youth Science and Research Foundation of Chengyi College Jimei University under contract C16005.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Gonzalez, R.C. and Woods, R.E. (1992) Digital Image Processing. Addison Wesley, New York.
- [2] Forsyth, D.A. (2002) Computer Vision: A Modern Approach. Prentice-Hall, New Jersey.
- [3] Leavers, V.F. (1993) Survey: Which Hough Transform. *CVGIP: Image Understanding*, **58**, 250-264. <https://doi.org/10.1006/ciun.1993.1041>
- [4] Duda, R.O. and Hart, P.E. (1972) Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Communications of the ACM*, **15**, 11-15. <https://doi.org/10.1145/361237.361242>
- [5] Xu, L. and Oja, E. (1993) Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities. *CVGIP: Image Understanding*, **57**, 131-154. <https://doi.org/10.1006/ciun.1993.1009>
- [6] Chen, T.C. and Chung, K.L. (2001) An Efficient Randomized Algorithm for Detecting Circles. *Computer Vision and Image Understanding*, **83**, 172-191. <https://doi.org/10.1006/cviu.2001.0923>
- [7] Fischer, M.A. and Bolles, R.C. (1981) Random Sample Consensus: A Paradigm to Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, **24**, 381-395. <https://doi.org/10.1145/358669.358692>
- [8] Jia, L.Q., Peng, C.Z., Liu, H.M. and Wang, Z.H. (2011) A Fast Randomized Circle Detection Algorithm. *International Congress on Image and Signal Processing*, **4**, 835-838. <https://doi.org/10.1109/CISP.2011.6100372>
- [9] Gomes, J. and Velho, L. (1997) Image Processing for Computer Graphics. Springer-Verlag, New York. <https://doi.org/10.1007/978-1-4757-2745-6>
- [10] Gonzalez, R.C. and Woods, R.E. (2008) Digital Image Processing. Pearson Prentice Hall, New Jersey.
- [11] Davies, E.R. (2005) Machine Vision: Theory Algorithms Practicalities. Morgan Kaufmann, San Francisco.
- [12] Fan, Y. (2007) Digital Image Processing and Analysis. Beijing University of Aeronautics and Astronautics, Beijing.