ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

FAST RANDOMIZED ALGORITHM FOR CIRCLE DETECTION BY EFFICIENT SAMPLING

LIANYUAN JIANG, PEIHE TANG, HAOHAO YUAN, CHUNGUI LI, YALAN ZHANG

Department of Computer Engineering, Guangxi University of Technology, Liuzhou 545006, Guangxi, China

ABSTRACT

According to the randomized circle detection (RCD) algorithm, the probability that four randomly sampled points belong to the same circle is relatively low. A fast algorithm for circle detection in this paper is proposed. Three points will be randomly sampled in the edge image and the fourth point is then sampled within the margin of vertical circumscribed and inscribed squares of the circle determined by three previously sampled points, and it effectively increases the probability of four randomly sampled points belonging to the same circle. In judging whether these four points can determine a candidate circle, a threshold is defined to reduce the calculation times of circle parameters. It affirms the candidate circle for true circle by a fast evidence-collecting process. Experimental results demonstrate that the proposed algorithm is about an order of magnitude faster than RCD in the detection speed, and has a strong robustness. The detection precision for circles is better than half a pixel.

Keywords: *Circle Detection, Randomized Sampling, Circle Parameters, Evidence Collecting*

1. INTRODUCTION

Fast and accurate detection of circles is widely applied in the fields of image processing and computer vision [1-3]. The Hough transform (HT), as a basic method to detect circles, has its advantages in insensitivity to noise in images and easiness in parallel computing, which has attracted extensive research among researchers [4-6]. HT has the advantages of small storage and high speed if the parameter space is limited to two dimensions. When it is more than two, however, e.g. the circle with three dimensions, HT will only be theoretically applicable due to the sharp increase of computing time and storage. To solve this problem, Xu et al [7] presented the randomized Hough transform (RHT), which can significantly reduce computing time and storage requirements by random sampling in the edge image and establishing a link-list data structure for the parameters. In RHT, three points are firstly picked randomly in the edge image to determine a circle and then judge whether its parameters are approximately equal to some parameters in the link-list. If no qualified parameters are found in the link-list, a new unit for the parameters will be defined and appropriately inserted into the link-list; otherwise, the score of the qualified parameters will be increased by one. If the resultant score is not smaller than a given threshold nt, then a candidate circle is found and will be further affirmed for a true circle by evidence-collecting process.

Excellent performance of RHT can be showed in simple image detection, but when it comes to multi-circle complex images, numerous accumulations of useless parameters from aimless sampling will demand a huge amount of computation and storage. As a result, some improved RHT methods were proposed [8-12]. Chen and Chung [13] presented a method called randomized circle detection (RCD) to avoid the massive memory and computing time requirements caused by parameter accumulation. In RCD, four points are randomly picked from the edge image and the parameters of a circle can be calculated by any three points. If the fourth point is also on the circle, then the circle is affirmed for a true circle by an evidence-collecting process. Nevertheless, slight improvement was achieved with regards to the computing time. Chung and Huang [14] improved the computing speed of RCD by only making comparative operation for the points outside the vertical circumscribed of the candidate circle in the evidence-collecting process, it is, however, not so obvious. Chen et al [15] also improved the computation speed by tracking edge points to form a series of edge chains which enables the four randomly picked points to be from the same edge chain, but the tracking edge points reduced the robustness of the detection. Chung et al [16] made use of the multiple-evidence-based

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

sampling strategy and refinement strategy to improve both the execution-time performance and the detection accuracy for RCD, but the robustness of the detection was also affected as a result of the gradient information use.

To the noticeable improvement of the computing speed of RCD and to keep the high robustness of the detection, this paper presents the method of sampling the fourth point of RCD and effectively increase the rate of the four sample points belonging in the same circle. Meanwhile, the proposed algorithm reduce the computing times of the circle parameters by defining a threshold value and affirm whether the candidate circle is a true circle using fast evidence-collecting process.

2. RCD ALGORITHM FOR CIRCLE DETECTION

Let *P* denote the set of all edge points in the image, and $|P|$ denote the number of edge points retained in *P*. Suppose the minimum threshold of $|P|$ is $T_{\min}$ and the sampling failure counter is *f*. The number of failures that can be tolerated is assumed as T_f, and the distance between any two sampling points of the candidate circle should be larger than T_{d1}. Factors such as discretization will be used to judge whether the point lies on the candidate circle, with its distance threshold T_{d2}. If the number of the points that lie on the candidate circle is larger than $M_{\min}$ ($M_{\min} = \lambda \times 2\pi r$, where λ is a proportional coefficient, and *r* is the radius of the candidate circle), then the candidate circle is affirmed as a true one.

Steps for RCD algorithm can be described as follows:

Step 1. Store all points into the set *P* and initialize the failure counter *f* to be 0.

Step 2. If $f = T_f$ or $|P| < T_{\min}$, then stop; otherwise, we randomly pick four points $p_i (i = 1,2,3,4)$ and remove these four points out of *P*.

Step 3. From these four edge points, find out such a candidate circle that the distance between any two of the three sampling points is larger than T_{d1} and the fourth point also lies on the circle, and then a candidate circle is affirmed; go to Step 4. Otherwise, return these four points into *P*; perform $f = f + 1$; go to Step 2.

Step 4. Set the counter *C* to be 0. We put these four points back into *P* [17]. For each point in *P*, we check whether its distance to the boundary of the candidate circle is not larger than the given distance threshold T_{d2}; if yes, perform $C = C + 1$ and remove the point out of *P*.

Step 5. If $C \geq M_{\min}$, go to Step 6. Otherwise, regard the candidate circle as a false one and return these *C* edge points into *P*; perform $f = f + 1$, and go to Step 2.

Step 6. The candidate circle has been detected as a true circle. Reset *f* to be 0 and go to Step 2.

3. THE BASIC IDEA OF THE PROPOSED ALGORITHM

3.1 Method of Four Points Sampling

In RCD, the random selection of four points from the set of edge points results in the low probability of them belonging on the same circle. In this paper, we propose a new method in which we randomly select three points p_1, p_2, and p_3 with the distance between any two of them is larger than T_{d1}, and then we calculate the parameters $C(a, b, r)$ of the circle determined by these three points, the circle corresponding to the parameters calculated is marked as C_{123}. The fourth point $p_4(x_4, y_4)$ is then selected randomly within the margin of vertical circumscribed and inscribed squares of C_{123} in order to effectively improve the probability of the four randomly sampled points belonging to the same circle. In the gray area illustrated in Fig. 1, under ideal conditions, the point p_4 can only exist within the margin of the vertical circumscribed square (*ABCD*) and inscribed square (*EFGH*) of C_{123}. Both the circumscribed square (*ABCD*) and the inscribed square (*EFGH*) center on *O* and the side length of the former is 2*r* while that of the latter is $\sqrt{2}r$. Since the digital image consists of discrete pixels, p_4 is considered on the C_{123} if point $p_4(x_4, y_4)$ and parameters $C(a, b, r)$ meet $\left|\sqrt{(x_4 - a)^2 + (y_4 - b)^2} - r\right| < T_{d2}$ .Therefore, while selecting p_4, the side lengths of *ABCD* and *EFGH* should be equal to $2(r + T_{d2})$ and $\sqrt{2}(r - T_{d2})$, respectively. Therefore, $p_4(x_4, y_4)$ is possible to lie on the C_{123} only when Eqs. (1)~(3) hold at the same time.

$$T_1 < x_4 < T_2 \tag{1}$$

$$T_3 < y_4 < T_4 \tag{2}$$

Journal of Theoretical and Applied Information Technology
20th February 2013. Vol. 48 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$x_4 > T_5 or\ x_4 < T_6 or\ y_4 > T_7 or\ y_4 < T_8 \quad (3)$$

where $T_1 = a - r - T_{d2}$, $T_2 = a + r + T_{d2}$, $T_3 = b - r - T_{d2}$, $T_4 = b + r + T_{d2}$, $T_5 = a + \frac{\sqrt{2}(r - T_{d2})}{2}$, $T_6 = a - \frac{\sqrt{2}(r - T_{d2})}{2}$, $T_7 = b + \frac{\sqrt{2}(r - T_{d2})}{2}$, $T_8 = b - \frac{\sqrt{2}(r - T_{d2})}{2}$.

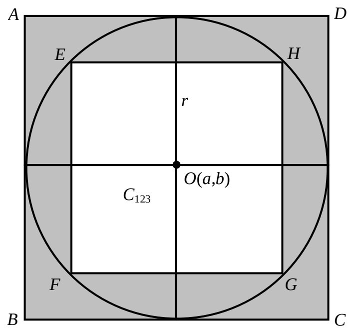


Figure 1: The Sampling Area Of P_4

Set m as the threshold of the failure counter of sampling P_4, and the method of selecting the fourth point P_4 we propose in this paper can be summarized as following:

Step 1. Let $i = 0$.

Step 2. Select randomly from the edge points a candidate p_4.

Step 3. If the p_4 satisfies the Eqs. (1)~(3) at the same time, then it is a true p_4 and stop. Otherwise, perform $i = i + 1$, and go to Step 4.

Step 4. If $i < m$, go to Step 2; otherwise, select new three points (p_1, p_2, and p_3) and go to Step 1.

3.2 Find the Candidate Circle

Suppose the distance of any one among the four sampling points, e.g. p_4, to the boundary of C_{123} determined by the other three points is marked as $d_{4\to123}$. To judge whether the four points can determine a candidate circle, we compare the $d_{4\to123}$ with T_{d3} and T_{d2} (the T_{d3} is slightly larger than T_{d2}). In this case, if $d_{4\to123}$ is larger than T_{d3}, then these four points cannot determine a candidate circle. If the $d_{4\to123}$, T_{d2}, and T_{d3} meet $T_{d2\le}d_{4\to123} \le T_{d3}$, we pick another one point out of these four and make the same comparisons. If some distance is smaller than T_{d2}, we confirm that the other three points determine a candidate circle [17].

Methods to judge whether the four points can determine a candidate circle in this paper can be described as follows:

Step 1. Assume the four points sampled as p_1, p_2, p_3, and p_4; as for p_4, we obtain it by the way mentioned in 3.1. The distance from p_4 to the boundary of C_{123} is marked as $d_{4\to123}$, and let $i = 4$.

Step 2. If $d_{4\to123} > T_{d3}$, then these four points cannot determine a candidate circle. If $T_{d2\le}d_{4\to123} \le T_{d3}$, go to Step 3; if $d_{4\to123} < T_{d2}$, a candidate circle is determined by p_1, p_2, and p_3.

Step 3. $i = i - 1$. If $i = 0$, then these four points cannot determine a candidate circle; otherwise, exchange the coordinate values of p_i and p_4, and go to Step 4.

Step 4. If the distance of p_1 to p_2, p_2 to p_3, and p_3 to p_1 is all larger than T_{d1}, then calculate the parameters of the circle determined by these three points, and go to Step 2; otherwise, go to Step 3.

4. ALGORITHM DESCRIPTION

Procedures of algorithm in this paper can be described as follows:

Step 1. Store all points into the set P and initialize the failure counter f to be 0.

Step 2. Select four different points from P according to the method described in 3.1.

Step 3. Judge whether these four points can determine a candidate circle according to the method described in 3.2. If yes, go to Step 4, otherwise, go to Step 5.

Step 4. Affirm whether the candidate circle is a true circle using the fast evidence-collecting process [12]. If yes, go to Step 6, otherwise, go to Step 5.

Step 5. Let $f = f + 1$. If $f > T_f$, then stop; otherwise, go to Step 2.

Step 6. Make sure whether the number of the circles detected is the same as the given one. If so, then stop; otherwise, remove the points lying on this circle out of P and reset $f = 0$, and go to Step 2.

5. EXPERIMENTAL RESULTS

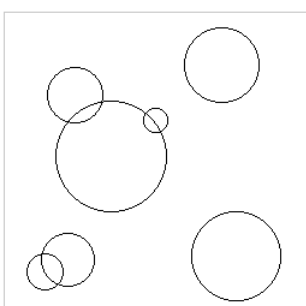
Many experiments have been carried out to validate the high speed and resolution of the

Journal of Theoretical and Applied Information Technology
20th February 2013. Vol. 48 No.2

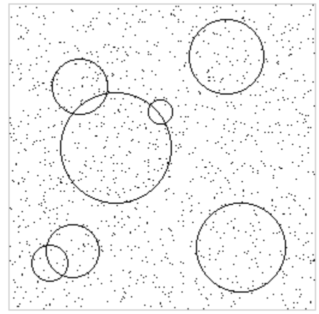
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

proposed algorithm. Here two of them are illustrated, compared with RHT, RCD, and PRCD [14]. In the first experiment, the values of λ, T_{d2}, and n_t are set to be 0.7, 0.5, and 2 when applying RHT, respectively. When applying RCD or PRCD, the values of λ, T_{d1}, and T_{d2} are set to be 0.7, 6, and 0.5, respectively. And the values of λ, T_{d1}, T_{d2}, T_{d3}, and m are set to be 0.7, 6, 0.5, 0.75, and 20 when applying the proposed algorithm, respectively. In the second experiment, all the values of parameters are the same as the first one except that all the values of λ are set to be 0.6. For fair competition, each algorithm will be stopped when the number of detected circles is equal to the given one. These two experiments are performed on a Pentium 2.66 GHz and memory 512 MB computer using C++ language.

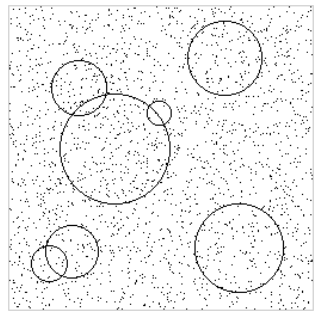
Refer to Ref. [13]. The first experiment is tested on the synthetic images that are created by adding noise to a 256×256 original image. The original synthetic image with 1 034 points is shown in Fig. 2(a), which consists of seven circles with different radii. In order to test the robustness, we randomly add different levels of noise to the original image. Here, the levels range from 30% (310 noises) to 210% (2 171 noises). The resulting images with 1 241 and 2 171 noises are shown in Fig. 2(b) and Fig. 2(c), respectively. The execution time required in each algorithm is obtained from the average of 50 simulations and illustrated in table 1. These four algorithms can correctly detect the seven circles to each synthetic image. The detected results are shown in Fig. 2(d) and table 2 with the proposed algorithm used in Fig. 2(c).



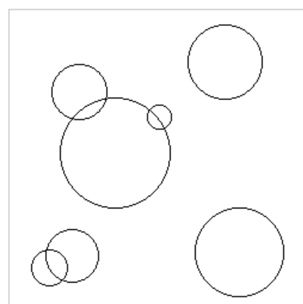
(a) The original image



(b) The image with 1 241 noises



(c) The image with 2 171 noises



(d) The detected circles with the proposed algorithm

Figure 2: The experiment on the synthetic images

Table 1: Time Performance Comparison

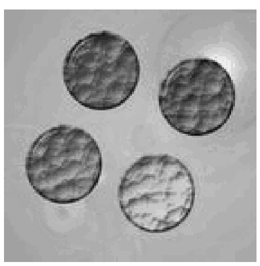
Noise ratio/%	RHT/s	RCD/s	PRCD/s	The proposed algorithm/s
0	0.0050	0.0056	0.0044	0.0009
30	0.0337	0.0591	0.0513	0.0059
60	0.3241	0.3178	0.2734	0.0309
90	1.1159	1.1794	1.0134	0.0950
120	2.9181	3.9959	3.4037	0.3009
150	11.1306	9.7959	8.2109	0.7519
180	27.0338	17.2028	14.2044	1.4009
210	49.7947	34.2716	27.9622	2.1362

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

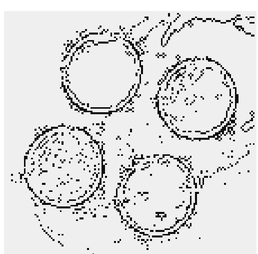
Table 2 : The Detected Results With The Proposed Algorithm Used In Fig. 2(C) (Unit: Pixel)

No.	The abscissa of the circle center		The ordinate of the circle center		The circle radius	
	Detected result	Real data	Detected result	Real data	Detected result	Real data
1	193.17	193	51.96	52	37.27	37
2	89.04	89	135.20	135	46.07	46
3	181.15	181	211.22	211	31.07	31
4	52.77	53	49.05	49	21.80	22
5	59.25	59	185.97	186	23.03	23
6	34.19	34	39.09	39	14.93	15
7	126.03	126	164.75	165	10.06	10

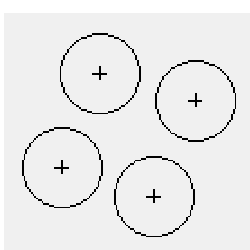
The second experiment is carried out on a 140×140 real image that is shown in Fig. 3(a). These four algorithms can correctly detect the four circles. As to the edge image including 1 688 points in Fig. 3(b), the average execution time of 50 simulations is 9.2372 s, 5.5212 s, 4.6619 s, and 0.9503 s, corresponding to RHT, RCD, PRCD and the proposed algorithm, respectively. The detected circles with the proposed algorithm are shown in Fig. 3(c).



(A) The Original Image



(B) The Edge Image



(C) The Detected Circles With The Proposed Algorithm

Figure 3: The Experiment On The Real Image

6. CONCLUSION

As the probability of four randomly sampled points belonging to the same circle in RCD is relatively low, this paper has presented a fast algorithm for circle detection, which effectively increases the probability of four randomly sampled points belonging to the same circle. When judging whether a candidate circle can be determined by the four points, we have reduced the number of calculations of the circle parameters by defining the threshold T_{d3}, which effectively improves the detecting speed. Besides, this method possesses the similar advantages to RCD, such as strong robustness and low demand for memory. Experimental results demonstrate that the proposed algorithm is about an order of magnitude faster than RCD in detection speed while the detection precision for circles is better than half a pixel.

In order to get the more precise parameters detected, the method can be further optimized by following the below: using the least square circle fitting method to fit the edge points lying on the circle whenever a true circle is detected in order to improve the detection accuracy of the circle parameters. The least-squares circle fitting method can be found in the literature [18]. Besides, the novelty in the proposed algorithm can be applied in other improved algorithms [15, 16] to effectively improve the detection speed, which is thus of potential application value. Further study will be carried on the combination aspects between RHT and the random sampling method of RCD.

ACKNOWLEDGEMENTS

This work was supported by the Guangxi Special Major and Curriculum Integration Project (No. GXTSZY217), the Science Research Program of Guangxi Education Department of China (No. 201010LX214), and the Science Research Foundation of Guangxi University of Technology (No. 1074009).

REFERENCES:

[1] I. Frosio, N.A. Borghese, "Real-time accurate circle fitting with occlusions", *Pattern Recognition*, Vol. 41, No. 3, 2008, pp. 1041-1055.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[2] E. Cuevas, V. Osuna-Enciso, F. Wario, D. Zaldívar, M. Pérez-Cisneros, "Automatic multiple circle detection based on artificial immune systems", *Expert Systems with Applications*, Vol. 39, No. 1, 2012, pp. 713-722.

[3] E. Cuevas, D. Oliva, D. Zaldivar, M. Pérez-Cisneros, H. Sossa, "Circle detection using electro-magnetism optimization", *Information Sciences*, Vol. 182, No. 1, 2012, pp.40-55.

[4] H.S. Kim, J.H. Kim, "A two-step circle detection algorithm from the intersecting chords", *Pattern Recognition Letters*, Vol. 22, No. 6-7, 2001, pp. 787-798.

[5] E. Duquenoy, A. Taleb-Ahmed, "Applying the Hough transform pseudo-linearity property to improve computing speed", *Pattern Recognition Letters*, Vol. 27, No. 16, 2006, pp. 1893-1904.

[6] H.D. Cheng, Y.H. Guo, Y.T. Zhang, "A novel Hough transform based on eliminating particle swarm optimization and its applications", *Pattern Recognition*, Vol. 42, No. 9, 2009, pp. 1959-1969.

[7] L. Xu, E. Oja, "Randomized Hough transform (RHT): basic mechanisms, algorithms, and computational complexities", *CVGIP: Image Understanding*, Vol. 57, No. 2, 1993, pp. 131-154.

[8] D. Walsh, A.E. Raftery, "Accurate and efficient curve detection in images: the importance sampling Hough transform", *Pattern Recognition*, Vol. 35, No. 7, 2002, pp. 1421-1431.

[9] Q. Ji, Y. Xie, "Randomized Hough transform with error propagation for line and circle detection", *Pattern Analysis and Applications*, Vol. 6, No. 1, 2003, pp. 55-64.

[10] L. Xu, "A unified perspective and new results on RHT computing, mixture based learning, and multi-learner based problem solving", *Pattern Recognition*, Vol. 40, No. 8, 2007, pp. 2129-2153.

[11] L.Y. Jiang, "Fast detection of multi-circle with randomized Hough transform", *Optoelectronics Letters*, Vol. 5, No. 5, 2009, pp. 397-400.

[12] L.Y. Jiang, "Efficient randomized Hough transform for circle detection using novel probability sampling and feature points", *Optik*, Vol. 123, No. 20, 2012, pp. 1834-1840.

[13] T.C. Chen, K.L. Chung, "An efficient randomized algorithm for detecting circles", *Computer Vision and Image Understanding*, Vol. 83, No. 2, 2001, pp. 172-191.

[14] K.L. Chung, Y.H. Huang, "A pruning-and-voting strategy to speed up the detection for lines, circles, and ellipses", *Journal of Information Science and Engineering*, Vol. 24, No. 2, 2008, pp. 503-520.

[15] A.J. Chen, J.Z. Li, D.D. Li, "Improved randomized algorithm for circle detection", *Opto-Electronic Engineering*, Vol. 33, No. 12, 2006, pp. 91-95.

[16] K.L. Chung, Y.H. Huang, S.M. Shen, A.S. Krylov, D.V. Yurin, E.V. Semeikina, "Efficient sampling strategy and refinement strategy for randomized circle detection", *Pattern Recognition*, Vol. 45, No. 1, 2012, pp. 252-263.

[17] L.Y. Jiang, "Fast randomized algorithm for detecting circles", *Opto-Electronic Engineering*, Vol. 37, No. 1, 2010, pp. 70-75.

[18] N. Chernov, C. Lesort, "Least squares fitting of circles", *Journal of Mathematical Imaging and Vision*, Vol. 23, No. 3, 2005, pp. 239-252.