A Fast Randomized Circle Detection Algorithm

Li-qin JIA[1], Cheng-zhang PENG[2], Hong-min LIU[1], Zhi-heng WANG[1]
[1]School of Computer Science and Technique, Henan Polytechnic University, Jiaozuo 454000, China
[2]School of Computer Science and Technology, Northwestern Polytechnical University, Xi'an 710072, China
E-MAIL: lqjia@hpu.edu.cn, abelard2009@mail.nwpu.edu.cn, hongminliu@hpu.edu.cn, wzhenry@eyou.com

***Abstract*—In this paper, we present a fast randomized circle detection algorithm applied to determine the centers and radii of circular components. Firstly, the gradient of each pixel in the image is computed using Gaussian template. Then, the edge map of the image, obtained by applying canny edge detector, is tackled to acquire the curves consisting of 8-adjacency connected edge points. Subsequently, for the detection of the center, N edge points for each curve are picked up, and the point, passed through by the most gradient lines of the edge points, corresponds to a center. The radius can be received by computing the distance between the center and the corresponding edge points. The algorithm performs much better in terms of efficiency compared to randomized circle detection algorithm (RCD), in which a mass of accumulations are done by random sampling. Synthetic images and natural images are used to test the capability of the proposed algorithm. The experimental results indicate that the presented algorithm consumes less computing resources, has excellent performance for detection of single circle, multiple circles, concentric circles, partial circles and overlapped circles, and also has good accuracy despite the presence of different noises and interference.**

Keywords-randomized circle detection; 8-adjacency connectivity; multi-circles detection; concentric circle detection; partial circle detection

I. INTRODUCTION

In the field of pattern recognition, circle detection is important and has been extensively studied and applied, such as automatic inspection and assembly, target locating and medical images processing in the past decades. Hough transform (HT) [1] is the most straightforward algorithm for the circle detection, which provides enough robust but requires massive computation and memory. In order to overcome the limitations of HT, some new approaches based on HT were proposed, e.g., the probabilistic HT [2], the randomized HT (RHT) [3] and the fuzzy HT [4]. In RHT, edge pixels are randomly sampled and mapped into one point, which reduces the storage requirement and the computing time. However, RHT accumulates in the parameter space, which results comparatively large computation and memory. Moreover, the accuracy of the detection results is directly related with the level of the parameter space quantized. For this reason, circles are hardly located well in an image. Chen proposed the randomized circle detection (RCD) [5] in 2001. Different from RHT, RCD detects the potential circles by making use of hypothesis-verification algorithm. The efficiency and accuracy of the detection results are enhanced compared with RHT for it does not need an accumulator in the parameter space. 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.

To solve these problems, a fast randomized circle detection algorithm is proposed in the paper. Firstly, Gaussian template is applied to compute the gradient of the image. Then, the edge pixels with 8-adjacency connectivity are connected. Further, N edge pixels are randomly picked up in the same connected curve which can exactly determine a possible circle with center and radius. The algorithm is faster than RCD for it samples only on the connected curve and also does not need evidence-collecting. The detection results show that the proposed algorithm has good detection performance and can detect many kinds of circular components well. We present the application results of the provided algorithm to both synthetic and natural images.

The remainder of this paper is organized as follows. Section 2 introduces the principle of the algorithm. Section 3 gives a detailed description for this algorithm in steps. The application results of the algorithm on both synthetic and natural images are displayed in Section 4. Section 5 presented the conclusions.

II. PRINCIPLE OF THE PROPOSED ALGORITHM

For a circle, the gradient vector of an edge pixel points at the center of the circle. Thus, define the line, which is coincident with the gradient vector of the edge pixel and also passes through the pixel, as gradient line of the edge pixel. Obviously, the gradient lines of all the edge pixels lying on the edge of a circle can intersect at the center of the circle. The radius of the circle can be determined by computing the distances between the edge pixels and the estimated center. Fig.1 shows the principle the proposed algorithm. N edge pixels (indicated as A, B, C, $\cdots$) are randomly picked up in one connected curve, and the corresponding gradient lines are denoted respectively as $l_A, l_B, l_C, \cdots$. The gradient lines yield several intersection points, in which the point passed through by most lines is the center of the circle (indicated as O). Subsequently, the distances of the center to the edge points whose gradient lines intersect at the center, denoted as $d_{A\to O}, d_{B\to O}, d_{C\to O}, d_{D\to O}$ as Fig.1 shows, are computed, and the radius is received as the mean of the distances.

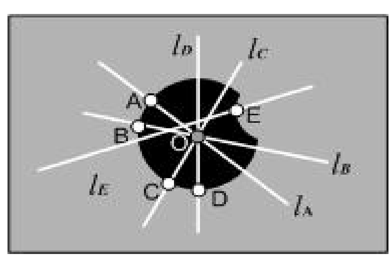


Figure 1. Principle the proposed algorithm

III. PROPOSED ALGORITHM

Based on the principle of the algorithm, we propose an algorithm to detect the centers and radii of the circles as follows:

The image has $r \times c$ pixels.

(1) The gradient of each pixel in the image is computed by using Gaussian template.

(2) Canny detector is used to get the edge map of the image and K curves consisting of the edge pixels with 8-adjacency connectivity are connected.

(3) For the kth connected curve ($k = 1, 2, \cdots, K$) :

All edge pixels in the kth connected curve construct a set V , and No_V denotes the number of pixels in V . If $No_V > T_{\min}$, do

a) N edge pixels on the curve are picked up by homogeneous distribution and $P_i (i = 1, 2, \cdots, N)$ denotes an edge pixel.

b) The gradient line of edge pixel P_i is computed and indicated as $L_i (i = 1, 2, \cdots, N)$.

c) The C_N^2 intersection points are yielded by the lines in which the point passed through by the most lines is indicated as O . The accumulation of the point O is denoted as O_{num} . If $O_{num} > T_{num}$, the point O is considered as the center of a circle, else turn to step (3) to deal with the following curve.

d) The distances values d_i between O and the edge points whose gradient lines passing through O are computed, and radius r is solved by computing the average of the distance values.

end

(4) For all K candidate circles, $C_1, C_2, \cdots$ denotes their centers. If distance between arbitrary two centers $| C_i - C_j | < 3, (i, j = 1, 2, \cdots K)$, the two centers will be considered as one center and the coordinate values are the average of the two centers.

end

Considering the number N effects the computation of the algorithm greatly, it can be 5~10. The factor $T_{\min}$ can be set small (e.g. 50) for ordinary images without noise but big (e.g. 120) for images with noise. For a circular curve, when $N = 10$, O_{num} and d_{num} will be 25~30 and 10 respectively, so T_{num} is set as 5 in general which can exclude triangles, rectangles, polygons and non-circular curves effectively. Step (4) is designed to determine the center of concentric circles.

We discuss the computational complexities of the proposed algorithm and RCD simply. Considering an image containing 88×145 pixels, there are 288 edge pixels in which edge pixels of two circles are 106 and 182 respectively. According to RCD, the probability of 4-tuple of randomly chosen pixels that come from the 106-edge-pixels circle is

$$p = \frac{106 \times 105 \times 104 \times 103}{288 \times 287 \times 286 \times 285} \approx 0.0184 ,$$

that is to say, $1 / 0.0184 \approx 54$ times should be run at most for finding a circle. While in the proposed algorithm, the probability of finding a circle is 1 given a circular curve. After finding out a candidate circle, the two circles need 284 (288-4) and 178 (182-4) times verifying computing in RCD but there is no step like this in proposed algorithm because only circular curves can product intersection point passed by mostly lines. Certainly, the proposed algorithm need to compute gradient of the image and 8-adjacency connectivity curves, but the computation is less. For the 88×145 image, the computational complexities of RCD and the proposed algorithm are 1000ms and 85ms respectively approximately, that is to say, the proposed algorithm is a fast randomized circle detection method.

IV. EXPERIMENT RESULTS

Experiment tests have been developed both on synthetic and natural images in order to evaluate the performance of the provided algorithm. N and T_{num} are 10 and 5 respectively in all tests.

A Synthetic images

As shown in Figure 2 (a), an image with two different circles of size 88×145 is generated in order to evaluate the performance of the proposed algorithm. There are two connected curves indicated on Figure 2 (b). For the first curve, N edge points are picked up by homogeneous distribution and the gradient lines of the edge pixels are almost intersected on one point (Figure 2 (c)), and then the center and radius detected are accurate as shown in Figure 2 (c). Figure 2 (d) indicates the detection on the second curve and the final detection results. The results are displayed in Figure 2(e), in which the centers and radii are detected accurately.

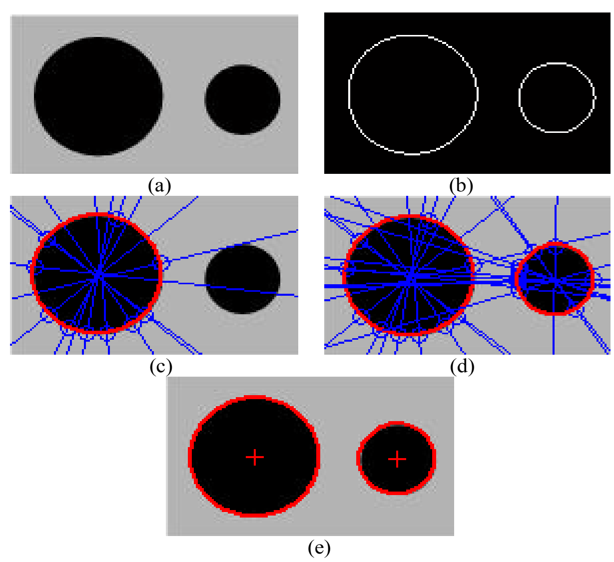
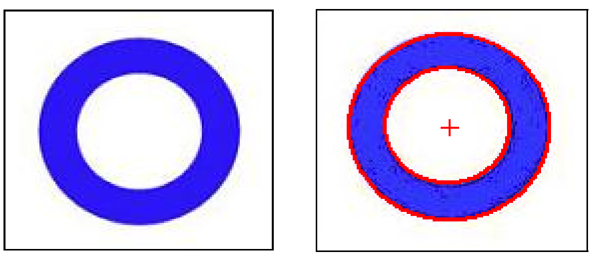


Figure 2. Detection of two circles. (a) is the original image, (b) is the edge detection, (c) and (d) is the circle detection on the first and the second connected curve, (e) is the detection result

Figure 3 shows the detection result of concentric circles. A 130×135 original image with concentric circles is shown in Figure 3 (a). For the outer curve, the detection center is (64, 67) and the radius is 50, and the center (63, 66) and radius 31 determine the inner curve. According to step (4), the two centers are considered as one center (63, 66) for they are too closed by computing average using method fix. The detection result is shown in Figure 3 (b), which indicates the centers and radii of concentric circles accurately.

The result of proposed algorithm on partial circles is shown in Figure 4. The original image and edge detection are shown in Figure 4 (a) and (b) respectively. The detection result is given as (c). The experiment indicates, if the edge of the circle is gotten, even a circular curve, the circle will be accurately determined.

Figure 5 shows the noise resisting of the algorithm. Figure 5 (a)-(c) display the original image without noise and the edge and circle detected results individually. When Gaussian noise ($\sigma^2 = 0.002$) is added, the edge image and the detected results is shown in Figure 5 (e) and (f) respectively. The factor $T_{\min}$ is set as 120 for there is too many connected curve in the image and the shorter curves should be not considered. Figure 5 (h)-(j) display the results of adding Salt & Pepper noise ($\sigma^2 =$ 0.03). Here, the most connected curves are short, so $T_{\min}$ is set as 50. The results show that the proposed algorithm has good resisting for the noise effects.



(a) (b)

Figure 3. Detection of concentric circles

Figure 4. Detection of partial circles

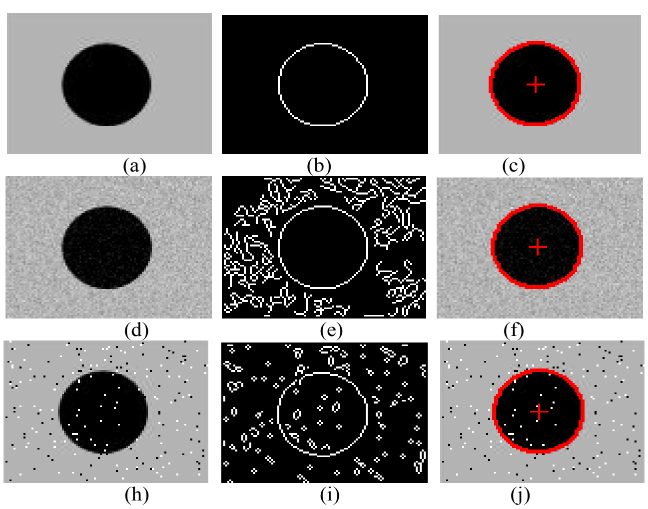


Figure 5. Detection of circles with noise

B *Natural images*

In this section the proposed algorithm is tested upon natural images.

$T_{\min}$ is set as 50 in Figure6-Figure 10. Figure 6 and Figure 7 show the detection of moon and golf balls. The results demonstrate the proposed algorithm have excellent detection performance on both single and multiple circles. Figure 8, Figure 9 and Figure 10 display the detection of concentric circles, partial circle and overlapped circles, and the results show the proposed algorithm can detect their centers and radii accurately.

Figure 11 displays the detection of a globe. The original image (a) is computed to get the edge by canny detector shown as (b) in which many connected curves are received. The result is displayed as (c) and indicates the proposed algorithm has excellent performance against interference.

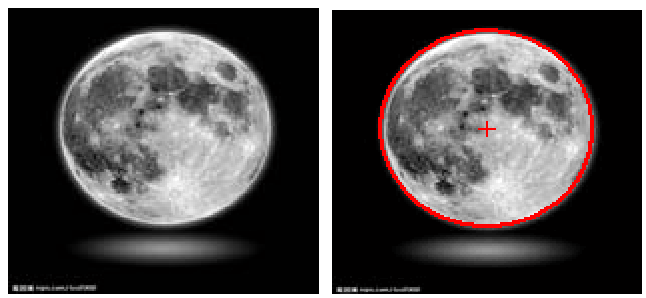


Figure 6. Detection of moon

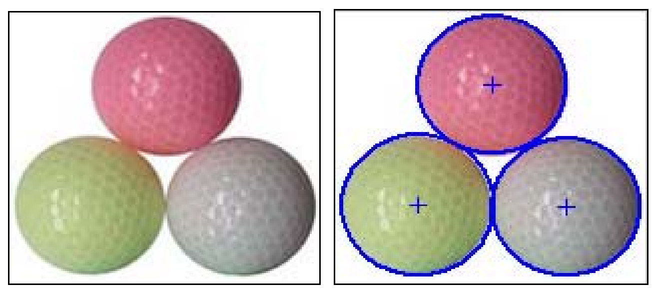


Figure 7. Detection of golf balls

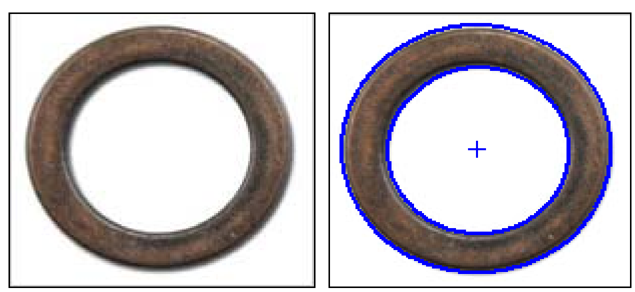


Figure 8. Detection of partial moon

Figure 9. Detection of partial moon

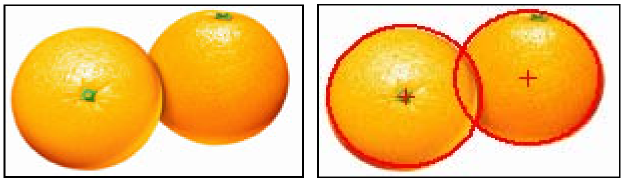
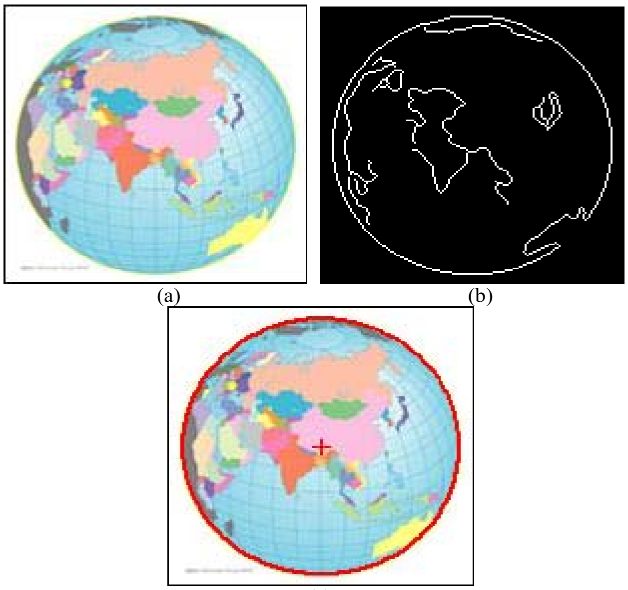


Figure 10. Detection of oranges



(a) (b)

(c)

Figure 11. Detection of globe

V. CONCLUSIONS

In this paper, a fast randomized circle detection algorithm is proposed. On the 8-adjacency connectivity edge curve, several points are picked up whose gradient lines intersect at the center. The radius is obtained by computing the distance between the edge points and the center. The algorithm consumes less computing resources for it avoids useless computations of random sampling and evidence-collecting compared to RCD algorithm. The experimental results indicated that our algorithm can be effectively applied to detect the centers and radii of single circle, multiple circles, concentric circles and partial circles, and has good accuracy despite of the presence of different noises and interference.

ACKNOWLEDGEMENTS

This work is supported by National Natural Science Foundation of China (61005033), Open Projects Program of National Laboratory of Pattern Recognition (20090018) and Doctor Degree Foundation of Henan Polytechnic University (B2010-39, B2010-68).

REFERENCES

[1] Illingworth J, Kittler J. A survey of the Hough transform. Computer Vision Graphics Image Process, 1988, pp. 87-116.

[2] Shaked, D., Yaron, O., Kiryati, N., “Deriving stopping rules for the probabilistic Hough transform bysequential analysis”, Computer Vision Image Understanding, 1996, pp. 512–526..

[3] Xu, L., Oja, E., and Kultanen, P., “A new curve detection algorithm: Randomized Hough Transform (RHT)”, Pattern Recognition Letters, 1990, pp. 331–338

[4] Han, J.H., Koczy, L.T., Poston, T., “Fuzzy Hough Transform”, In Proc. 2nd Int. Conf. on Fuzzy Systems, 1993, pp. 803–808.

[5] T.C. Chen, K.L. Chung, “An efficient randomized algorithm for detecting circles”, Computer Vision and Image Understanding, 2001, pp. 172–191.

[6] Xu L, Oja E, Kultanen P., “Randomized Hough Transform (RHT): basic mechanisms, algorithms, and computational complexities”, Comput Vision Graphics Image Process: Image Understanding, 1993, pp. 131- 154.

[7] Han, J.H., Koczy, L.T., Poston, T., “Fuzzy Hough Transform”, In Proc. 2nd Int. Conf. on Fuzzy Systems, 1993, pp. 803–808.

[8] V. A. Ramirez, C. H. G. Capulin, A. P. Garcia, R. E. S. Yanez, “Circle detection on images using genetic algorithms”, Pattern Recognition, 2006, pp. 652-657.

[9] K. L. Chung, Y. H. Huang, J. P. Wang, T. C. Chang, H. Y. M. Liao, “Fast randomized algorithm for center-detection”, Pattern Recognition, 2010, pp. 2659-2665.