AMSE JOURNALS-2016-Series: Advances B; Vol. 59; N°1; pp 177-188
Submitted July 2016; Revised Oct. 15, 2016, Accepted Dec. 10, 2016

An Improved Randomized Algorithm for Detecting Circles

* Lianyuan Jiang, ** Zhiwen Wang, *** Yiyi Xu, **** Meizhen Liu, ***** Jian He

* College of Computer Science and Communication Engineering, Guangxi University of Science and Technology, Liuzhou 545006, China
College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
Guangxi Experiment Center of Information Science, Guilin 541004, China
(The corresponding author: jly_jly_jly@163.com.)
** College of Computer Science and Communication Engineering, Guangxi University of Science and Technology, Liuzhou 545006, China
*** College of Computer Science and Communication Engineering, Guangxi University of Science and Technology, Liuzhou 545006, China
**** Video and Electronic Publishing House, Guangxi University of Science and Technology, Liuzhou 545006, China
***** College of Computer Science and Communication Engineering, Guangxi University of Science and Technology, Liuzhou 545006, China
Guangxi Experiment Center of Information Science, Guilin 541004, China

Abstract

In randomized circle detection algorithm, once a candidate circle is generated, the circle parameters will not be calculated for the 4 sampled points involved in the generation of the candidate circle. Utilizing this feature, we propose an improved randomized algorithm for detecting circles. Suppose that the candidate circle is judged as false circle and the number of points located on this candidate circle is above a certain threshold. If among the 4 sampled points, there are still 3 points for which the circle parameters are not yet calculated, then the circle parameters are calculated for these 3 points. Experiments with synthetic images and real images indicate that the probability that the circle represented by these circle parameters is a true circle is

very large. Therefore, the detection speed with the proposed algorithm is faster compared with randomized Hough transform and randomized circle detection algorithm. The same principle can be applied to other improved randomized circle detection algorithm.

Key words

Circle detection, candidate circle, circle parameter, randomized sampling, false circle.

1. Introduction

Circle detection algorithm is widely used in object location, robot technology and automated inspection [1-3]. As one of the common algorithms for detecting circles, Hough transform has two major advantages: first, it is insensitive to noises in the image; second, it allows parallel computing. Many improved Hough transform algorithms have been proposed targeting at the slow computing speed and large storage space required by the algorithm [4]. Xu et al. [5, 6] proposed the randomized Hough transform (RHT) as a major improvement of HT in terms of storage space and calculation speed. The former is also featured by unlimited parameter space and arbitrarily high detection precision. RHT displays excellent performance for simple images, but huge calculation load and storage space will be required for complex images. Therefore, RHT is also improved dramatically [7-9]. For example, Jiang [8] utilized probability sampling and feature points to optimize the method for selecting sampled points and finding candidate circle so as to increase the detection speed.

To reduce the storage space and search time needed for parameter accumulation in RHT, Chen and Chung [10] proposed the randomized circle detection algorithm (RCD). The principle is to choose 4 sampled points from the image and, by defining a distance criterion, to determine whether these 4 points constitute a candidate circle. If this candidate circle is eligible, the evidence-collecting process is used to determine whether the candidate circle is a true circle. Using RCD, there is no need to store an accumulator, which is used to store the relevant parameter information. This is the biggest difference with HT and RHT. Some improvements have been made based on RCD [11-13]. Chung and Huang [11] proposed a pruning-and-voting strategy to increase the speed of circle detection. This strategy also applies to the detection of lines and ellipses. Jiang [12] determined whether the 4 sampled points constituted a candidate circle by defining a new distance threshold, thus reducing the calculation of circle parameters and increasing the detection speed. Chung et al. [13] increased the calculation speed of RCD by the strategy of multiple-evidence-based sampling.

However, the probability of a circle obtained by circle parameters from randomly sampled points to become a true circle is very small. To increase the probability, we propose an improved randomized algorithm for detecting circles which examines whether the number of points on the candidate circle judged as the false circle is above a certain threshold. If yes, it is determined whether among the 4 sampled points for generating the candidate circle, there are still 3 points for which the circle parameters are not yet calculated. If yes, the circle parameters at these 3 points are calculated. The probability of the circle represented by the circle parameters at these 3 points to be a true circle is very large. The experiment provides the probability of this circle to be a true circle using the proposed algorithm, thus confirming the reliability of the algorithm.

2. Improved randomized algorithm for detecting circles

2.1. Improved method to find candidate circle

The meanings of the symbols are first defined. V is the set of edge points in the image; T_f is the maximum acceptable number of consecutive failure. Considering discretization of the images, among the 3 sampled points, the distance between any 2 points should be above T_a. T_d is the distance threshold used for determining whether a point is located on the circle. If the number of points on the candidate circle is no less than the threshold $M_{\min}$, the candidate circle is considered a true circle. Since the perimeter varies with radius, $M_{\min}=T_r\times 2\pi r$, where T_r is a proportionality factor and r is the radius of the candidate circle.

Suppose one circle parameter can be calculated from any 3 randomly sampled points and therefore 4 circle parameters are obtained from 4 randomly sampled points. Among the 4 circles determined by the 4 circle parameters, any 1 circle can be a true circle. In RCD algorithm, once 1 candidate circle obtained from the 4 points is judged a false circle, it is believed that no true circle can be obtained from these 4 points.

If a candidate circle is judged a false circle and the number of points located on the candidate circle is above a certain threshold, then this candidate circle may possibly be the approximation of a true circle. On this basis, the RCD is improved as follows: if a candidate circle is judged a false circle and the number of points located on the candidate circle is above the threshold $T_{\min}$ ($T_{\min}=T_n\times M_{\min}$, where $0<T_n<1$), then new circle parameters are calculated from the remaining 3 points.

As shown in Fig. 1, the 4 randomly sampled points are v_1, v_2, v_3 and v_4, respectively; v_1, v_2 and v_3 are concyclic points, whereas v_4 is closer to this circle. If the circle parameters are calculated from the points v_1, v_3 and v_4, the circle represented is C_{134} (dotted-line circle in Fig. 1).

If the distance from v_2 to C_{134} is smaller than T_d, then C_{134} is judged a candidate circle. In the evidence-collecting process, C_{134} will be judged a false circle. At this time another 4 points will be randomly sampled in conventional RCD. But with the improved RCD algorithm, the circle parameters will be calculated for other 3 points among v_1, v_2, v_3 and v_4, which may very likely result in a true circle. Therefore, the proposed algorithm takes less time to detect a true circle.

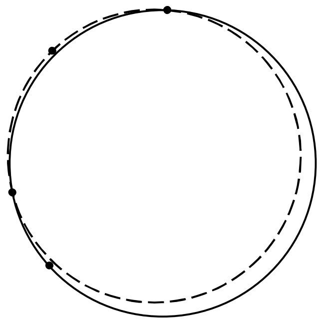


Fig. 1. The special case of randomly sampled points

The improved method to find candidate circle can be described as follows:

(1) The 4 randomly sampled points are v_1, v_2, v_3 and v_4. The circle determined by v_1, v_2 and v_3 is C_{123}; the distance from v_4 to the boundary of C_{123} is $d_{4\rightarrow 123}$, i=4.

(2) If the distance between any 2 points among v_1, v_2 and v_3 is larger than T_a, 1 circle parameter is obtained from these 3 points and move to (3); otherwise, move to (4).

(3) If $d_{4\rightarrow 123}>T_d$, move to (4); otherwise, v_1, v_2 and v_3 determine a candidate circle C_{123} and move to (5).

(4) i=i-1. If i=0, move to (7); otherwise, the coordinates of v_i and v_4 are swapped and move to (2).

(5) Determine whether the candidate circle C_{123} is a true circle using the method in section 2.2. If yes, a true circle C_{123} is detected; otherwise, C_{123} is a false circle and move to (6).

(6) Determine whether the number of points on C_{123} is larger than the threshold $T_{\min}$. If yes, move to (4); otherwise, move to (7).

(7) A new round of sampling begins and the 4 sampled points are v_1, v_2, v_3 and v_4 with i reset to 4 and move to (2).

The improved method has an additional step (6) compared with conventional RCD. As will be verified by the experiment in section 3, step (6) greatly improves the success rate of finding candidate circle and true circle and hence improves the detection speed.

2.2. Affirming the candidate circle for a true circle

Referring to Ref. [8], the method to determine whether the candidate circle is a true circle can be optimized in three aspects.

(1) In the evidence-collecting process, if any point is not located within the area corresponding to the difference of area enclosed by two vertical squares whose center is the center of the candidate circle and the length of sides is $2(r+T_d)$ and $\sqrt{2}(r-T_d)$, respectively (r is the radius of the candidate circle), then this point is not considered the point on the candidate circle. By this simple comparison, most points not on the candidate circle can be excluded, thereby increasing the speed of affirming the candidate circle for a true circle.

(2) When determining whether the distance from a point to the boundary of the circle is smaller than the threshold T_d, the formula can be simplified by taking the squared value and the like. This circumvents the operations of taking the square root and the absolute value, thereby increasing the calculation speed. Such simplification technique is also applied to the calculation of distance in step (2) and (3) in section 2.1.

(3) In the evidence-collecting process, if the sum of the number of points in the set of edge points not yet collected and that of points on the candidate circle already counted is smaller than $M_{\min}$, this candidate circle is considered a false circle. By this optimization, the calculation load in evidence collecting is reduced.

2.3. Algorithm description

The proposed algorithm is realized through the following steps:

(1) Store all edge points in the image into set V and initialize the number of sampling f=0.

(2) Four different points are randomly selected from V.

(3) Decide whether the 4 points can determine a candidate circle. If yes, move to (4), otherwise, move to (7).

(4) Determine whether the candidate circle is a true circle using the method in section 2.2. If yes, move to (8), otherwise, move to (5).

(5) Determine whether the number of points on the candidate circle is larger than the threshold $T_{\min}$. If yes, other candidate circles are sought using these 4 points according to the method in section 2.1, and move to (6); otherwise, move to (7).

(6) If true circle is found using the method in section 2.1, move to (8); otherwise, move to (7).

(7) f=f+1. If f>T_f, the detection is over; otherwise, move to (2).

(8) Determine whether the number of detected circles has reached the specified value. If yes, the detection is over; otherwise, the points located on the circle are removed from V with f reset to 0, and move to (2).

3. Experiment and analysis

A large number of synthetic images and real images are used for the verification experiments. It is found that the proposed algorithm has a higher detection speed. Due to limited space, only 4 experiments are described. The parameter configuration is as follows: for experiment 1, T_r, n_t [5] and T_d are 0.7, 2 and 0.5 in RHT and T_r, T_a and T_d are 0.7, 6 and 0.5 in RCD, respectively. For the proposed algorithm, T_r, T_a, T_d and T_n are 0.7, 6, 0.5 and 0.3, respectively. For experiment 2, 3 and 4, T_r is 0.6 for all algorithms, and other parameters are the same as in experiment 1.

Some elucidations of the experiment are given as follows:

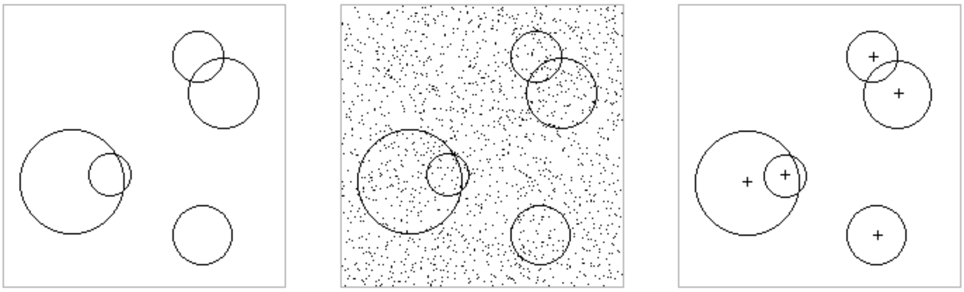
(1) Detection using RHT or RCD is over when all the circles in the image are detected.

(2) With the proposed algorithm, when a candidate circle is considered a false circle and the number of points on the candidate circle is larger than $T_{\min}$, the number of calculations of circle parameters for 3 out of the 4 points (not yet calculated) is p_1; it is the number of calculations of circle parameters in step (5) in section 2.3. The number of circles determined by p_1 circle parameters that will be the candidate circles is c_1; the number of the c_1 candidate circles that will be finally considered the true circles is r_1. With the proposed algorithm, the number of calculations of circle parameters is subtracted by p_1 value, and p_2 is obtained. The reason for calculating the p_2 circle parameters using the proposed algorithm is the same as with RCD. The number of circles determined by p_2 circle parameters that will be the candidate circles is c_2; the number of the c_2 candidate circles that will be finally considered the true circles is r_2.

(3) All programs are written in C++ language. The hardware configuration is Intel Core2 Duo processor (2.93GHz) and 2G memory.

(4) Since randomized sampling is adopted in all three algorithms, the detection time is calculated as the average time taken for doing 50 detections.

Experiment 1. See Ref. [10]. Fig. 2(a) is a 200×200 image containing 5 circles on which 647 edge points are located. Different levels of random noise are added into Fig. 2(a) with noise ratio of 50%-300%, i.e. the number of noise points is 324-1941. Fig. 2(b) is the image obtained by adding 1618 noise points into Fig. 2(a). The circles are detected using RHT, RCD and the proposed algorithm after adding noise into Fig. 2(a). The detection time of the three algorithms is shown in Table 1. The proposed algorithm is used to detect the circles for 100 times in Fig. 2(b), and the coordinates of circle center and radius of the circle are obtained accurately. Among the 100 detections, the results of 1 detection are shown in Table 2 and Fig. 2(c). The origin of coordinate in Fig. 2 is the point on the left lower corner of the square; the coordinate axes are parallel with the two adjacent sides of the square, respectively. Then the proposed algorithm is used to detect the circles for 50 times after adding noise to Fig. 2(a). The average values of p_1, c_1, r_1, r_1/p_1, p_2, c_2, r_2 and r_2/p_2 are shown in Table 3.



(a) (b) (c)

Fig. 2. The experiment on the synthetic images (I): (a) The original image, (b) The image with 1618 noises, (c) The detected circles with the proposed algorithm

Table 1. Time performance comparison for synthetic images (I)

Noise ratio (%)	RHT (s)	RCD (s)	The proposed algorithm (s)
50	0.0103	0.0094	0.0041
100	0.0702	0.0549	0.0209
150	0.3189	0.2040	0.0942
200	0.6552	0.4749	0.2050
250	2.0380	1.2037	0.4749
300	5.7988	2.7596	1.0115

Table 2. The detected results with the proposed algorithm used in Fig.2(b) (Unit: pixel)

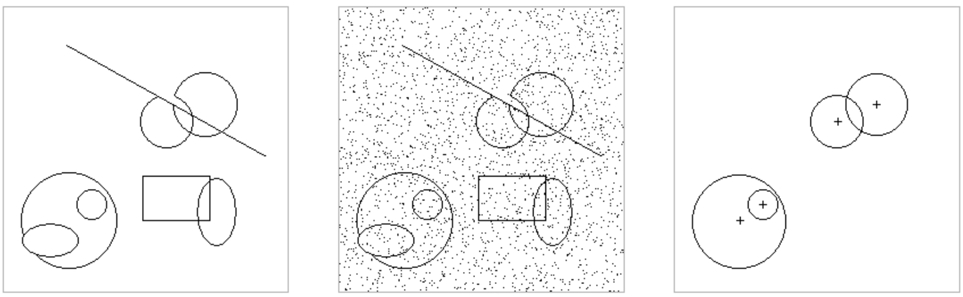
No.	The abscissa of the circle center		The ordinate of the circle center		The circle radius	
	Detected result	Real data	Detected result	Real data	Detected result	Real data
1	48.09	48	74.20	74	37.15	37
2	156.07	156	137.41	137	24.94	25

3	138.28	138	163.10	163	18.18	18
4	141.14	141	35.86	36	21.05	21
5	75.00	75	78.88	79	15.12	15

Table 3. The relevant statistical results for synthetic images (I)

Noise ratio (%)	p_1	c_1	r_1	r_1/p_1	p_2	c_2	r_2	r_2/p_2
50	7.88	6.22	1.10	0.139594	20969	155.34	3.90	0.000186
100	7.42	5.66	1.08	0.145553	110954	624.64	3.92	0.000035
150	8.46	6.36	1.44	0.170213	460326	2250.98	3.56	0.000008
200	10.30	7.52	0.90	0.087379	913473	4132.84	4.10	0.000004
250	13.60	9.80	1.22	0.089706	1964471	8412.38	3.78	0.000002
300	12.52	9.16	1.02	0.081470	3921987	16228.26	3.98	0.000001

Experiment 2. Fig. 3(a) is a 250×250 image containing 4 circles (3 circles are incomplete), 1 straight line, 1 rectangle and 2 ellipses, with a total of 1170 edge points. Different levels of random noise are added into Fig. 3(a) with noise ratio of 40%-200%, i.e. the number of noise points is 468-2340. Fig. 3(b) is the image obtained by adding 2340 noise points into Fig. 3(a). The circles are detected using RHT, RCD and the proposed algorithm after adding noise into Fig. 3(a). The detection time of the three algorithms is shown in Table 4. The proposed algorithm is used to detect the circles for 100 times in Fig. 3(b), and the coordinates of circle center and radius of the circle are obtained accurately. Among the 100 detections, the results of 1 detection are shown in Table 5 and Fig. 3(c). Then the proposed algorithm is used to detect the circles for 50 times after adding noise to Fig. 3(a). The average values of p_1, c_1, r_1, r_1/p_1, p_2, c_2, r_2 and r_2/p_2 are shown in Table 6.



(a) (b) (c)

Fig. 3. The experiment on the synthetic images (II): (a) The original image, (b) The image with 2340 noises, (c) The detected circles with the proposed algorithm

Table 4. Time performance comparison for synthetic images (II)

Noise ratio (%)	RHT (s)	RCD (s)	The proposed algorithm (s)
40	0.2172	0.2455	0.1204
80	0.9188	0.9207	0.5029
120	3.3415	2.6214	1.2751
160	7.8156	7.2113	3.2439
200	15.4478	14.8228	5.4541

Table 5. The detected results with the proposed algorithm used in Fig.3(b) (Unit: pixel)

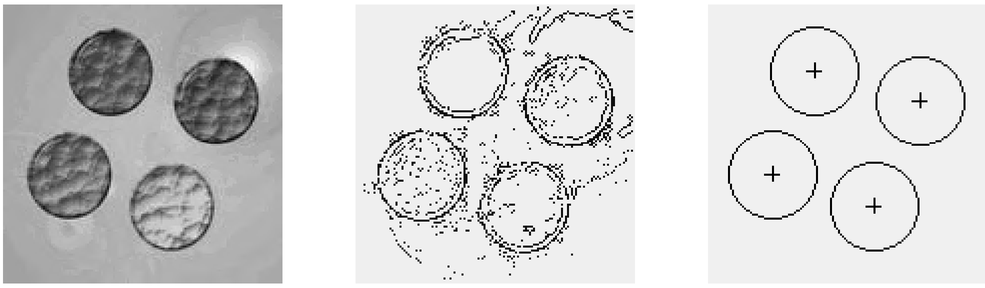
No.	The abscissa of the circle center		The ordinate of the circle center		The circle radius	
	Detected result	Real data	Detected result	Real data	Detected result	Real data
1	56.93	57	62.18	62	42.19	42
2	176.94	177	164.06	164	28.23	28
3	142.72	143	149.16	149	22.99	23
4	77.32	77	76.02	76	13.30	13

Table 6. The relevant statistical results for synthetic images (II)

Noise ratio (%)	p_1	c_1	r_1	r_1/p_1	p_2	c_2	r_2	r_2/p_2
40	15.76	11.92	0.62	0.039340	508638	2751.16	3.38	0.000007
80	23.56	17.62	0.52	0.022071	1999597	8640.66	3.48	0.000002
120	39.48	29.26	0.52	0.013171	4698798	18120.92	3.48	0.000001
160	67.68	50.26	0.64	0.009456	11249726	40023.12	3.36	0.000000
200	90.94	66.92	0.58	0.006378	17673536	59774.58	3.42	0.000000

Experiment 3. Fig. 4(a) is a 140×140 real image containing 4 circles. The process of circle detection is shown in Fig. 4. In the edge image Fig. 4(b) there are a total of 1688 edge points [14]. The three algorithms are used to detect the circles in Fig. 4(b), and the time taken for the detection was 5.4185 s, 2.1138 s and 0.4958 s, respectively. Of 50 detections, all three algorithms can correctly extract the coordinates of circle center and the radius of each circle. The average

values of p_1, c_1, r_1, r_1/p_1, p_2, c_2, r_2 and r_2/p_2 are 765.88, 561.76, 1.04, 0.001358, 2128706, 16487.58, 2.96 and 0.000001, respectively.



(a) (b) (c)

Fig. 4. The experiment on the real image (I): (a) The original image, (b) The edge image, (c) The detected circles with the proposed algorithm

Experiment 4. Fig. 5(a) is a 360×300 real image containing 4 circles. The process of circle detection is shown in Fig. 5. In the edge image Fig. 5(b) there are a total of 3991 edge points. The three algorithms are used to detect the circles in Fig. 5(b), and the time taken for the detection was 14.6197 s, 12.4142 s and 3.2067 s, respectively. Of 50 detections, all three algorithms can correctly extract the coordinates of circle center and the radius of each circle. The average values of p_1, c_1, r_1, r_1/p_1, p_2, c_2, r_2 and r_2/p_2 are 5141.70, 3575.44, 1.44, 0.000280, 8420882, 43815.06, 2.56 and 0.000000, respectively.

(a) (b) (c)

Fig. 5. The experiment on the real image (II): (a) The original image, (b) The edge image, (c) The detected circles with the proposed algorithm

4. Conclusion and discussion

For 4 randomly sampled points in RCD, if the circle determined by 3 points has the remaining point on its edge, the circle parameters will not be calculated for any other 3 points. The improved RCD algorithm is based on the situation where the candidate circle is considered a false circle and the number of points on the candidate circle is larger than the threshold $T_{\min}$. The proposed algorithm is faster than conventional RCD. It can be known from the experimental results that r_1/p_1 is far larger than r_2/p_2. That is, the probability of the calculated circle parameters which are determined by the step (6) in section 2.1 becoming the true circles is far greater than that using RCD. This explains the faster calculation speed of the proposed algorithm. Although the overall detection speed with the proposed algorithm is not much faster compared with the conventional RCD, the detection can be more considerably accelerated by combining the proposed algorithm with other improved RCD algorithms [12, 13].

The proposed algorithm has only an additional step (6) in section 2.1 compared with conventional RCD. Therefore, the proposed algorithm has the same robustness and precision as RCD. How to further increase the detection speed of RCD will be the focus of feature work.

Acknowledgements: This work was supported by the National Natural Science Foundation of China (61462008), the Program of Guangxi Experiment Center of Information Science (KF1403), the Natural Science Foundation of Guangxi (2013GXNSFAA019336, 2013GXNSFBA019268), the Program of Guangxi Education Department (KY2016YB256, YB2014208, LX2014201, 2013JGA194), the Innovation Team Project of Guangxi University of Science and Technology, and the Doctor Foundation of Guangxi University of Science and Technology(12Z14).

References

1. Cuevas, E., 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, pp. 713-722, 2012.
2. Radhakrishnan, P., A. Clementking, V. Murugesh, “Pattern Generation for Geometrical Objects using Morphological Transformations”, AMSE Journals, Advances B, Vol. 57, No 2, pp. 44-56, 2014.
3. Ok, A.O., “A New Approach for the Extraction of Aboveground Circular Structures from Near-Nadir VHR Satellite Imagery”, IEEE Transactions on Geoscience and Remote Sensing, Vol. 52, No 6, pp. 3125-3140, 2014.
4. Mukhopadhyay, P., B.B. Chaudhuri, “A Survey of Hough Transform”, Pattern Recognition, Vol. 48, No 3, pp. 993-1010, 2015.
5. Xu, L., E. Oja, P. Kultanen, “A New Curve Detection Method: Randomized Hough Transform (RHT)”, Pattern Recognition Letters, Vol. 11, No 5, pp. 331-338, 1990.

6. Xu, L., E. Oja, "Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities", CVGIP: Image Understanding, Vol. 57, No 2, pp. 131-154, 1993.
7. Xu, L., "A Unified Perspective and New Results on RHT Computing, Mixture Based Learning, and Multi-Learner Based Problem Solving", Pattern Recognition, Vol. 40, No 8, pp. 2129-2153, 2007.
8. Jiang, L.Y., "Efficient Randomized Hough Transform for Circle Detection Using Novel Probability Sampling and Feature Points", Optik, Vol. 123, No 20, pp. 1834-1840, 2012.
9. Jiang, L.Y., Y.Q. Ye, G.L. Xu, "An Efficient Curve Detection Algorithm", Optik, Vol. 127, No 1, pp. 232-238, 2016.
10. Chen, T.C., K.L. Chung, "An Efficient Randomized Algorithm for Detecting Circles", Computer Vision and Image Understanding, Vol. 83, No 2, pp. 172-191, 2001.
11. Chung, K.L., 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, pp. 503-520, 2008.
12. Jiang, L.Y., "Fast Randomized Algorithm for Detecting Circles", Opto-Electronic Engineering, Vol. 37, No 1, pp. 70-75, 2010.
13. Chung, K.L., 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, pp. 252-263, 2012.
14. Wang, Z.W., S.Z. Li, Y. Dai, "Adaptive Image De-Noising Using Multivariate Statistical Model of Fractal-Wavelet Encodes", Information, Vol. 15, No 1, pp. 41-49, 2012.