International Journal of Multimedia and Ubiquitous Engineering
Vol.11, No.1 (2016), pp.105-114
http://dx.doi.org/10.14257/ijmue.2016.11.1.11

Multiple Circles Detection in Complex Scenes

Chu Guangli[1], Wang Yanjie[2], Sun Honghai[2], Fan Bo[2] and Wen Zhuoman[2, 3]

[1]*Baicheng Normal University, Baicheng 137000, China*
[2]*Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences, Changchun 130033, China*
[3]*University of Chinese Academy of Sciences, Beijing 100039, China*
qibaobei2001@163.com

Abstract

To solve the problem that the conventional circle detection algorithm has low processing speed and poor identification accuracy in presence of overlap of objects, a novel multiple circles detection algorithm is proposed. Firstly, the consecutive edges with single pixel are obtained through an edge tracking method. Moreover, the edges of overlaid objects are extracted from the cross section according to the linear projection method. Secondly, a large number of non-circular edges are excluded by utilizing a coarse detection that decides whether four points on an edge are on the same circle or not. Lastly, the circles are recognized by the fine detection according to the geometrical feature which two subarcs belonging to one circle have the same center and radius. The experimental results demonstrate that the algorithm is superior in speed, precision and accuracy, comparing to the conventional ones, such as Hough transform circle detection and randomized circle detection. Furthermore, it can solve the issues from overlap of objects and accurately identify the multiple circular objects in the complex scenes as well. The algorithm is concise, fast and robust.

***Keywords**: image processing; object identification; circle detection; least square method*

1. Introduction

Circle detection plays an important place in the field of optical precision measurements [1-3]. In addition, it is widely used in automat testing, iris recognition products, intelligent transportation and other fields, and is always a hot research issue in image processing and machine vision. The classic circular Hough transform (CHT) converts the abstract circular detection in image space into accumulation of three dimensional in parameter space, the peak represents the circle parameters. The disadvantages of the algorithm include needing more memory, slow, low accuracy, so a lot of improved algorithms are put forward such as randomized Hough transform (RHT) [4], fuzzy Hough transform (FHT) [5], point Hough transform (PHT) [6], these algorithms overcome the disadvantages of CHT to a certain extent, but they all are based on transformation processing space and voting mechanism, the speed still can not meet the requirement of real-time processing, and the error detection rate is high. Ayala-Ramirez *et al.* [7] propose the circle recognition method based on genetic algorithm (GA), implement to identify multiple circles in natural images, but it is poor for recognizing incomplete, obscured circles; Cuevas *et al.* [8] propose an automatic learning (LA) recognition algorithm based on optimization model, although it can identify multiple circles, but the automatic learning process requires a large number of iterations, complex computation, slow processing, and the number of iterations has a great influence on the results; Frosio *et al.* [9] propose an circle recognition algorithm based on maximum likelihood, it can detect partial occlusion circles, but need the information to predict radius, is poor general; Shang Fei *et al.* [10] propose an circle

ISSN: 1975-0080 IJMUE

detection method based on right triangle inscribed, it converts the solution of the parameter into triangular parameter calculation, avoids the repeated power and root operation, and saves processing time, but is poor for partial occlusion circles detection; Chen *et al.* [11] put forward a random circle detection algorithm (RCD), it random picks up four points from the edge image, determines the candidate circle by the mechanism of hypothesis test, recognize the true circles from the candidate circle round by collecting evidence identification, the detection speed is faster than RHT [12] when the ratio of noise points and effective points is less than 170%; Yu Dan *et al.* [13] reduce invalid accumulation, improve efficiency by changing the taking point range of RCD; Chung *et al* [14] add the gradient condition before determining the candidate circles, reduce the error detection chance, and determine the detection results by statistical voting, improves the detection accuracy, but neither the RCD nor the algorithms based on RCD are all randomly selected from the initial point, and completes recognition through a lot of accumulation, not only time-consuming, and the detection result is random, not stable and great error.

In order to solve above problems, this article gives an improvement algorithm, resolves the partial occlusion. The algorithm is simple and robust, real-time and can accurately detect multiple circles in complex scenes.

2. Our Algorithm

The image is often interfered by the noise in the process of acquisition, conversion and transmission. It not only reduces the image visual effect, but also is not conducive to recognize. In order to reduce the influence by the noise, Gaussian filter is used to smooth the loaded gray images. Then we use the Canny detection operator to detect image edge, and obtain the binary edge images. But in the binary edge image, the points are isolated with each other, and need to find continuous edge points by edge tracking process. Edge tracking [15] is the foundation work for image segmentation, target recognition and other operations. To meet the needs of subsequent operations, we adopt corresponding tracking algorithm to improve the processing efficiency. The following tracking algorithm is good for circular recognition, especially in the case of partial occlusion.

2.1 Single Pixel Edge Tracking

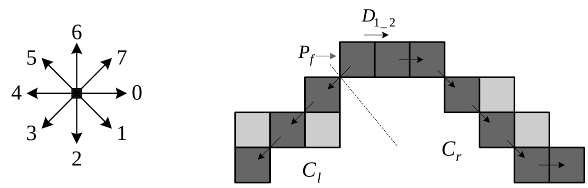
We search based on the eight neighborhood according to from top to bottom, from left to right in the binary edge image, the direction of the eight neighborhood code is shown in Figure 1 (a), the tracking process is shown in Figure 1 (b).

1) If there are not edge points which are not tracked in the binary image ,the tracking turn to 5), otherwise the found point P_f is put as a starting edge point ,turn to 2);
2) For the above and on the left side of the starting point P_f has been searched, we can just search the 0 ~ 3 four neighborhood, if there is no tracked target, the P_f is the isolated point and deleted, turn to 1), otherwise the found point is put as the second edge point and set to the current point, recorded as $D_{1\ 2}$ (the search direction from first to second, is one of the value of 0, 1, 2, 3), turn to 3;
3) Starting from the current point, search in clockwise order within the eight neighborhood to the current search direction D_c (from the last point to the current point on the search direction) as the initial direction, the direction code ($D_c + k$) is MOD8, the MOD represents the modulus operation, $k = 0, 1,... 7$; the found first point that is not to track as the next edge point, and at the same time all the points which are not tracked in the other direction within the eight neighborhood of the current point are deleted (the light color square represents the deleted point in Figure 1 (b)), the operation is used to ensure the edge is single pixel, then put the new found edge point as the current point, update D_c, turn to 3); If there does not exist points which are not tracked within eight neighborhood of the current point,

the tracked edge recorded as C_r, turn to 4);

4) If D_{1_2} is 2 or 3, then save C_r to the edge chain table, turn to 1); Or judge P_f in direction 3 to see if there is point that is not tracked, if there is (as shown in Figure 1 (b)), search in the same way along the direction 3, the edge is marked as C_l, and stored reversely in the edge list, namely the finally got point as the starting point of the whole edge, and then connect the C_r to the end of the C_l, combine an edge, turn to 1);

The tracking process is over, and got all the edge curves



(a) Direction Code (b) Tracking Procedure

Figure 1. Direction Code in Eight-Neighborhood and Tracking Procedure

Under the partial occlusion condition, the edges of the different objects exist intersection, if they are tracked as an edge, it will affect recognition, so we need to disconnect the edge at the intersection. Usually the edges in the intersection have a great change in the direction, which we can use the linear projection [16] to detect these points. The basic principle of linear projection method is that, put the edge that contains only a few pixels (4 pixels) as a straight line, using the local chain code in the x direction and y direction projection put the straight line into the form of vector, using the vector dot product formula to compute the angle between two straight lines.

Akinlar *et al.* [17] give a definition: under the condition that the premise is p, if the angle is less than $p \cdot \pi$, then the direction of the two straight lines is the same. According to this definition, p = 1/8 (because the edge tracking is based on the eight neighborhood search), when the angle θ is less than 22.5° ($0.125\pi = 22.5^\circ$), the direction of the two straight lines is the same. In the experiments, in order to increase the robustness, we set this value to 25°, if the θ of one point is more than nearby (taken the four points before the point and four points behind the point) average 25°, then it suggests that the edge direction has a great changes at that point, the both sides of the edge certainly does not belong to the same arc, then delete these point, make the edge off, to improve the occlusion resistance of the algorithm.

2.2 The Coarse Detection

The coarse detection principle is shown in Figure 2, determine the candidate circle by making use whether four points of edge are on the same circle.

When three points fit a circle, the fitting accuracy not only associated with the accuracy of fitting points, but also associated with the distribution of fitting points, and the high fitting precision occurs when three points and uniformly distributed on the circular. According to the conclusion and considering the edge may be closed, or points between at the first and at the end of the edge are close to, take the points at the starting edge point, a third and two-thirds, which are respectively p_1, p_2, p_3 in Figure 2. If the three points collinear, the edge is not circle, abandon; or they are fit to a circle (the dotted circle in the graph), the center is O, the radius is r_0, and take the point about 1/2 of edge as the fourth point p_4, then calculate the distance to the center of the circle. When $d \neq r_0$, we can determine that the edge is not circle, abandon; When $d = r_0$, determine the edge as a candidate circle, included two situations as in Figure 2 (a) and (b),but it also needs the subsequent fine detection, circle recognition. The adaptive taken point way in the coarse detection overcomes the randomness of RCD, and there's no need to take it over and over again, little computation and effect is remarkable. If the edge is short, such as less than 20 pixels, then skip the coarse detection process, directly into the fine detection stage, improve the ability of algorithm to identify small round.

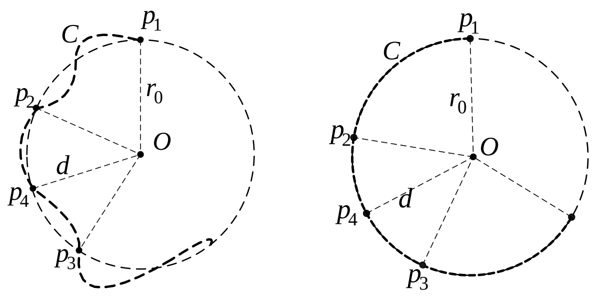


Figure 2. Principle of Coarse Detection

2.3 The Fine Detection

The theoretical basis of fine detection is: The two arcs on a circle have the same circle center and radius. As shown in Figure 3, assuming the edge curve M_1M_2 is part segment, the center point M of the arc divided it into two jokes M_1M and MM_2, and respectively fit through least square circle, get the circle center $O_1(x_{c1}, y_{c1})$ and $O_2(x_{c2}, y_{c2})$ respectively, and the radius R_{c1} and R_{c2} respectively.

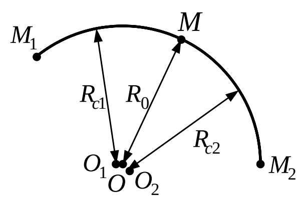


Figure 3. Principle of Fine Detection

The points on the segment circle M_1M have the equation:

$$(x - x_{c1})^2 + (y - y_{c1})^2 = R_{c1}^2 \tag{1}$$

We can get the equation(2) from equation(1):

$$x^2 + y^2 + Ax + By + C = 0 \tag{2}$$

Where $A = -2x_{c1}, B = -2y_{c1}, C = x_{c1}^2 + y_{c1}^2 - R_{c1}^2$.

The A, B and C as the independent variable to establish function:

$$F(A, B, C) = \sum (x_i^2 + y_i^2 + Ax_i + By_i + C)^2 \tag{3}$$

According to the principle of least squares fitting, in order to make F the minimum value, the partial derivatives of three independent variables, and set the partial derivatives to 0, get the equations, get the values A, B and C for the solutions of equations, and then have:

$$\begin{cases} x_{c1} = -\dfrac{A}{2} \\ y_{c1} = -\dfrac{B}{2} \\ R_{c1} = \dfrac{1}{2}\sqrt{A^2 + B^2 - 4C^2} \end{cases} \tag{4}$$

In the same way, the (x_{c2}, y_{c2}) and R_{c2} are obtained. Ideally, if the arcs M_1M and MM_2 are on the same circle, two groups of circle parameters should be accurately equal. But there are some errors in digital image processing, the center point and radius should be approximately equal respectively, the degree of approximately equal recorded as P (Precision):

$$P = (1 - \frac{|R_{c1} - R_{c2}|}{R_{c1} + R_{c2}})(1 - \frac{\sqrt{(x_{c1} - x_{c2})^2 + (y_{c1} - y_{c2})^2}}{R_{c1} + R_{c2}}) \tag{5}$$

The first factor reflects the relationship of radius, the radius value is $(0,1]$, the closer radius values are, the closer the factor value is to 1, whereas is to 0; The second factor reflects the relationship between the center of the circles, namely the position relationship between two circles. The position relationship between two circles in the plane including is from the outside, circumscribed, intersection, cut inside, inclusion and overlap. When it is from the outside, $\sqrt{(x_{c1} - x_{c2})^2 + (y_{c1} - y_{c2})^2} \geq R_{c1} + R_{c2}$,the second factor is $(-\infty, 0]$;When it is the intersection, cut inside, embedded and overlap, $\sqrt{(x_{c1} - x_{c2})^2 + (y_{c1} - y_{c2})^2} < R_{c1} + R_{c2}$, the second factor is $(0,1]$, and the closer the coincidence is, the factor value is close to 1. So the discriminant interval is $[k_1, 1]$, $k_1 \leq 1$, k_1 is set to 0.5 in this paper. If P is in discriminant interval, then do the least square circle fitting with N points on the edge of the whole article, get the center (x_0, y_0) , the radius R_0 . Calculate the relative standard deviation:

$$\delta = \frac{1}{R_0}\sqrt{\frac{\sum \left|(x_i - x_0)^2 + (y_i - y_0)^2 - R_0^2\right|}{N}} \tag{6}$$

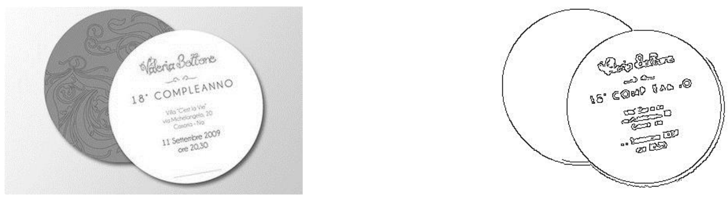
If δ meet the equation (7) it is a circle (k_2 is less than 1, this paper takes 0.1):

$$\delta < k_2 R_0 \tag{7}$$

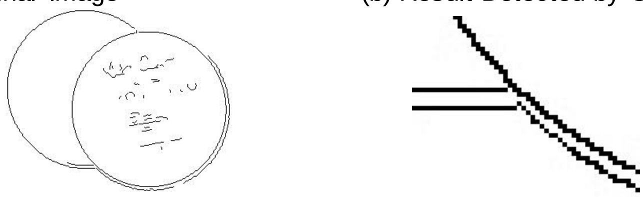
We should restructure the circles got, arc may be broken into multiple, fitted to multiple circles due to block or noise in the process of edge tracking. So we calculate the radius and center of the circle which meet the above conditions, according to the equation (5), then integrate the edges which are fell in the discriminant interval, and fit into a circle.

3. Experiments

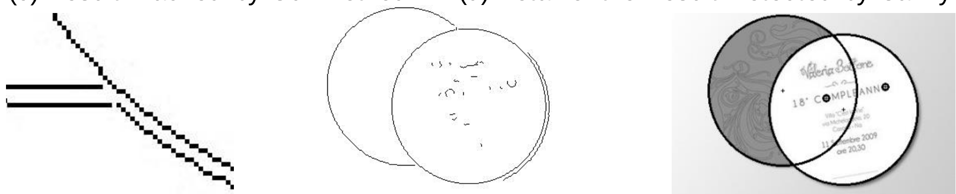
The experiments are performed on a 3.40GH CPU with 2G RAM using VC++6.0.In the algorithm, we get the images as shown in Figure 4, (a) the original image, size of 379×240. (b) the results of the Canny detection. (c)the tracked edges in this article , the edges whose length is less than 6 pixels without saving in the process of tracking because of a circle at least consists of six pixels, edges are refined compared with the results of (b), especially are more apparent in 45° and 135°. The Figures (d), (e) verify the effect of separating partial occlusion edges using the algorithm in this paper. (f) the edges of coarse detection, most of the edges are filtered after coarse detection, reduces the computational complexity in the later. (g) the identified results of this paper, detected 4 circles, two big partial occlusion circles and two letters O.



(a) Original Image (b) Result Detected by Canny



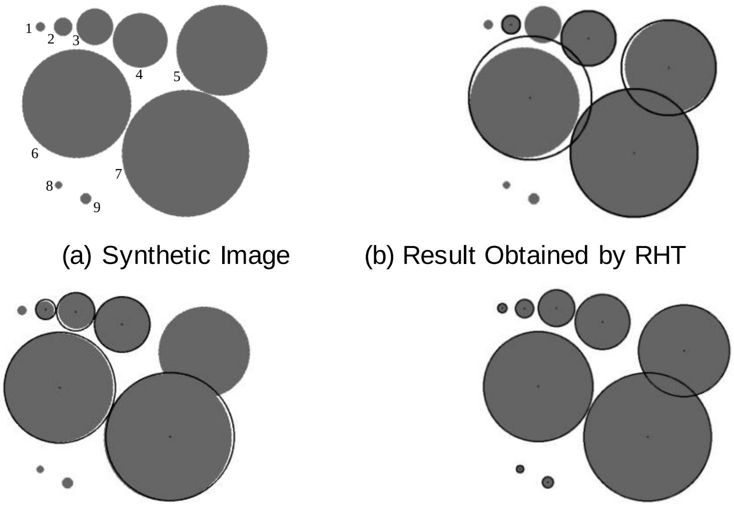
(c) Result Tracked by Our Method (d) Detail of the Result Detected by Canny



(e) Detail of the Result Tracked by Our Method (f) Edges after Coarse Detection (g) Result Recognized by Our Method

Figure 4. Processing Procedure of Our Method

The recognition accuracy comparison experiment by using synthetic image is shown in Figure 5. (a) Synthetic image, including 9 circles with known circle centers and radius, in order to contrast the recognition accuracy, the numbers in the Figures are the circles' number. (b) the recognition results of RHT algorithm in ref. [4], recognize five circles, numbers are 2, 4-7. (c) the recognition results of RCD algorithm in the ref. [11], recognize five circles, number are 2-7. The two algorithms both can not recognize small circles of numbers 1, 8, 9. Due to the randomness of RHT, RCD, we separately experiment many times with the two algorithms, and give the good recognition results in the paper. (d) the recognition result of our algorithm, accurately recognize the all circles.



(a) Synthetic Image (b) Result Obtained by RHT

(c) Result Obtained by RCD (d) Result Obtained by Our Method

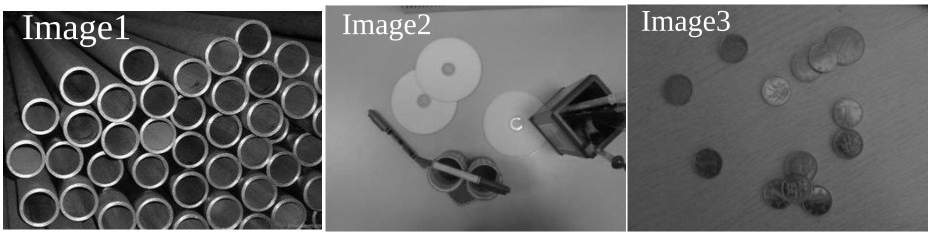
Figure 5. Comparison of Precision

The actual values of parameters of each circle and the detected values of each algorithm are listed in Table 1, "-" represents failing to identify. The data shows that our algorithm has higher recognition accuracy.

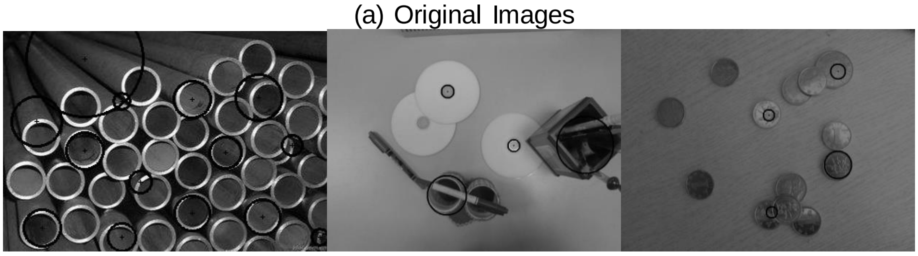
Table 1. Actual Parameters of Circles and Values Detected by Algorithms

No.	Actual values	RHT	RCD	Our method
1	(50,50) 5	-	-	(50,50) 5.150679
2	(100,50) 10	(100,50) 10.223749	(102,48) 11.267319	(100,50) 10.118223
3	(170,50) 20	-	(168,53) 21.568113	(170,50) 20.351082
4	(270,80) 30	(270,80) 30.299255	(270,81) 31.056372	(270,80) 30.177760
5	(450,140) 50	(446,144) 52.062485	-	(449,140) 50.261209
6	(130,220) 60	(142,210) 67.281189	(133,220) 62.382216	(130,220) 60.252412
7	(370,330) 70	(370,330) 71.246216	(375,328) 72.930857	(370,330) 70.165506
8	(90,400) 4	-	-	(89,400) 4.058953
9	(150,430) 6	-	-	(150,429) 6.187335

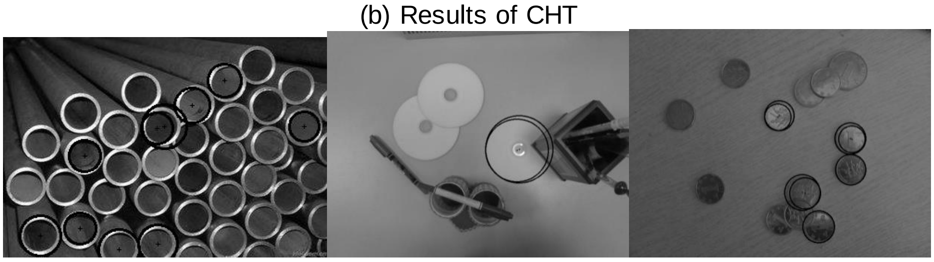
The result of circle detection performance comparison experiment using the natural images is shown in Figure 6, the contrast algorithm are respectively CHT and PHT in literature [6]. In image1, our algorithm detected 39 circles, only failed to detect two defect circles on the right side. The Image2 and Image3 all contain multiple circles and occur partial occlusion, this algorithm accurately detect all circles. The other two algorithms for three images are very serious in inspection and leak by mistake. This experiment shows that this algorithm has strong ability to recognize ability in complex scene, especially under the partial occlusion.



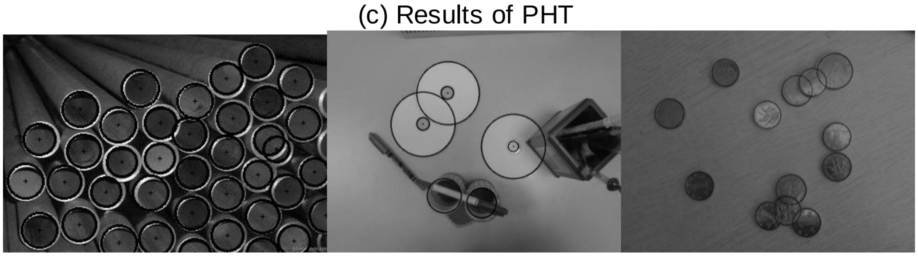
(a) Original Images



(b) Results of CHT



(c) Results of PHT



(d) Results of Our Method

Figure 6. Multiple Circles Detection

The time of circle detection part using three algorithms is listed in Table 2 , time unit is milliseconds (ms). Because do not need to accumulate and vote, simple operation, this algorithm's time is significantly less than the other two algorithms.

Table 2. Comparison of Time Spending on Circles Detection

Image(width×height)	CHT	PHT	Our method
Image1(350×240)	139	45	8
Image2(800×600)	172	51	10
Image3(800×600)	178	50	11

4. Conclusion

The paper proposes a multi-circle detection algorithm that is robust to partial occlusion, can recognize circles fast and accurately in complex scenes. The algorithm get single pixel edge by tracking based on the detection results of Canny, separate edges of the partial occlusion object using linear projection method, improves the recognition accuracy under the condition; the circle detection consists of coarse detection and fine detection, the operation is simple, fast, accurate and high precision. The algorithm is based on the results of Canny, but the Canny detection is time-consuming, if we can improve the edge detection speed, and get the continuous edges at the same time when detecting edge, can further increase the processing speed, and this is the next step that we shall research in the future.

References

[1] L. Jun, L. Yue and C. Weimin, "A Method for Multi-Circle Detection Based on Global Search and Evidence Accumulation", Acta Optica Sinica, vol. 9, no. 30, **(2010)**.
[2] A. Lei, T. Jiubin, C. Jiwen and K. Wenjing, "Fast and Precise Center Location for Circle Target of CCD Laser Auto Collimator", Acta Optica Sinica, vol. 2, no. 27, **(2007)**.
[3] Y. Ruixin, W. Shigang, G. Baiqing and W. Yingfengm "A Fast Interference-Resistant Deformable Template Method for Detecting the Circle", Acta Optica Sinica, vol. 8, no. 24, **(2004)**.
[4] L. Xu and E. Oja, "Randomized Hough Transform: Basic Mechanisms Algorithms and Computational Complexities", Computer Vision Graphic Image Process Image Understanding, vol. 2, no. 57, **(1993)**.
[5] K. P. Philip, E. L. Dove, D. D. McPherson, N. L. Cotteiner, W. Stanford and K. B. Chandran, "The Fuzzy Hough Transform-Feature Extraction in Medical Images", Medical Imaging, vol. 2, no. 13, **(1994)**.
[6] L. Jinlong and S. Qingyun, "Circle Recognition Through a Point Hough Transformation", Computer Engineering, vol. 11, no. 29, **(2003)**.
[7] V. A. Ramirez, H. Carlos, G. Capulin, A. P. Garcia, E. Raul and S. Yanez, "Circle Detection on Images Using Genetic Algorithms", Pattern Recognition, vol. 6, no. 27, **(2006)**.
[8] E. Cuevas, F. Wario, V. O. Enciso, D. Zaldivar and M. P. Cisneros, "Fast Algorithm for Multiple-Circle Detection on Images Using Learning Automata", Image Processing, vol. 8, no. 6, **(2012)**.
[9] I. Frosio and N. A. Borghese, "Real Time Accurate Circle Fitting", Pattern Recognition, vol. 3, no. 14, **(2008)**.
[10] S. Fei, W. Fenggui, T. Di and Z. Zhihui, "A Method for Circle Detection Based on Right Triangles in Scribed in a Circle", Acta Optica Sinica, vol. 4, no. 28, **(2008)**.
[11] T. C. Chen and K. L. Chung, "An Efficient Randomized Algorithm for Detecting Circles", Computer Vision and Image Understanding, vol. 2, no. 83, **(2001)**.
[12] Z. Yongliang, J. Yan, H. Ping and C. Qiang, "Accelerated Randomized Hough Transform for Circle Detection Using Effective Accumulation Strategy", Journal of Computer-Aided Design & Computer Graphics, vol. 4, no. 26, **(2014)**.
[13] Y. Dan and W. Wei, "Improved Randomized Circle Detection Algorithm", Journal of Inage and Graphics, vol. 8, no. 14, **(2009)**.
[14] C. Kuoliang, H. Yonghuai, S. Shiming, S. Andrey, Krylov, V. Dmitry, Yurin and V. Ekaterina, Semeikina, "Efficient Sampling Strategy and Refinement Strategy for Randomized Circle Detection", Pattern Recognition, vol. 1, no. 45, **(2012)**.
[15] W. Haibin, Y. Xiaoyang, G. Congrong, "Structured Light Encoding Stripe Edge Detection Based on Grey Curve Intersecting Point", Acta Optica Sinica, vol. 6, no. 28, **(2008)**.
[16] C. Jian, G. Chunxiao, L. Qiongxin and H. Yue, "Object Recognition Method Based on Corner Feature", J. Huazhong Univ. of Sci. & Tech. (Natural Science Edition), vol. 6, no. 38, **(2010)**.
[17] C. Akinlar and C. Topal, "ED Circles: A real-time Circle Detector with a False Detection Control", Pattern Recognition, vol. 3, no. 46, **(2013)**.
[18] H. Chauris, I. Karoui, P. Garreau, H. Wackernagel, P. Craneguy and L. Bertino, "The Circlet Transform: A Robust Tool for Detecting Features with Circular Shapes", Comput.Geosci., vol. 3, no. 37, **(2011)**.
[19] A. O. Ok, "A New Approach for the Extraction of Aboveground Circular Structures from Near-nadir VHR Satellite Image", IEEE Transactions on Geoscience and Remote sensing, vol. 6, no. 52, **(2014)**.