A Suggested Point Search Algorithm for Circle Detection in Binary Images

Sundus Kh. Ebraheem
College of Computer Sciences and Mathematics
University of Mosul

Received on: 16/8/2009 **Accepted on: 16/5/2010**

الملخص

نظرا لأهمية اكتشاف الدائرة في تحديد الأهداف في تطبيقات معالجة الصور، فقد تم في هذا البحث اقتراح خوارزمية لاكتشاف الدائرة، سميت الخوارزمية "بطريقة البحث النقطي لاكتشاف الدائرة" والتي تعتمد مبدأ إيجاد نقاط وافتراضها نقاطاً تقع على محيط الدائرة واعتمادها كنقاط تحري لتكوين نموذج لدائرة افتراضية باستخدام الصفات الهندسية للدائرة ليتم مطابقتها مع الصورة الأصلية، وتم استخدام عملية المطابقة في هذا البحث للتقليل من العمليات الحسابية والتقليل من درجة تعقيد الخوارزمية ووقت تنفيذ البرنامج. تتميز الخوارزمية المقترحة بالدقة والسرعة العالية والخزن القليل مقارنة بغيرها من طرق اكتشاف الدوائر. وبإمكان الخوارزمية المقترحة اكتشاف الدوائر ذات القياسات المختلفة وكذلك الدوائر المتقاطعة والمتداخلة بدقة في الصور الثنائية.

تم تنفيذ الخوارزمية المقترحة وطريقة Hough Transform (HT) لاكتشاف الدائرة على عدة صور بأحجام مختلفة تضم دوائر بأحجام وأعداد مختلفة للمقارنة وإظهار كفاءة الخوارزمية المقترحة، وقد أظهرت الخوارزمية المقترحة كفاءةً أكثر حيث كان معدل الزمن اللازم للتنفيذ بالخوارزمية المقترحة إلى طريقة HT1:646 وأثبتت الخوارزمية المقترحة دقة 100% في اكتشاف الدوائر. وكلا الخوارزميتين نفذتا بلغة Matlab 7.2 باستخدام حاسبة Pentium IV ومعالج 1.8MHz وذاكرة 512MB .

ABSTRACT

Detecting circles is very important in the application of image processing especially in determining the object locations. In this paper, a new algorithm is proposed for circle detection, called Point Search Circle Detection (PSCD), which detects points and assumes them as inspection points on the circle circumference by using them to create a virtual circle to match it with the original image. Using matching operation leads to reduce computational operations and reduce the complexity and the running time of the algorithm. The proposed algorithm is highly accurate, has high speed and low storage requirements in comparing with other related algorithms. The proposed algorithm can precisely detect circles with various scales, crossed and nested circles in the binary images.

The proposed algorithm was compared with Hough Transform (HT) method for circle detection by using many images with different numbers and radius of circles and different image dimensions. The proposed algorithm was more efficient, where the average ratio of the running time for the proposed algorithm to HT method was 1:646, and the accuracy of the proposed algorithm was 100% for circles detection. Both the proposed and HT algorithms are applied by using Matlab 7.2 language, PC equipment with 1.8MHz Pentium IV processor and 512MB RAM.

1- Introduction:

Extraction of features can be implemented through several different techniques; however, the choice of the feature, as well as of the technique to be used should take into account the contribution in terms of information that can be obtained from it. In other words, the choice of certain feature depends on its capacity for separating patterns [9]. Edge detection has been one of the most active fields in computer vision. Most of previous search on edge detection has been based on discrete approximation to differential operations [8]. Detecting lines and circles in an image is a fundamental issue in image processing applications. Extracting circles from digital images has received more attention for several decades because an extracted circle can be used to yield the location of circular object in many industrial applications. So far many circle-extraction methods have been developed [11].

One of the key issues image processing is to extract interested objects from an image. Hough transform (HT) is a classical algorithm for extracting line and circle from a binary image [6]. The Circle Hough transformation (CHT) has become a common method for circle detection in numerous image processing applications. The CHT has some disadvantages when it works on discrete images. The large amount of storage and computing power required by the CHT are the major disadvantages of using it in real time applications [11].

Many approaches have been presented to speed up the computation and reported to increase the HT performance so far and another improved algorithm is the generalized HT (GHT) which can detect arbitrary object in a grayscale image [6]. There are many methods for circle detection depending on (HT) and several methods utilize randomized selection of edge points and geometrical properties of circle. Many modifications have been reported to increase the CHT performance so far. Tsuji and Matsumoto decomposed the parameter space and used the parallel property of circle [12].

Xu and et. al presented an approach that randomly selects three pixels. The method selects three no collinear edge pixels and votes for the circle parameters which are found by using the circle equation [13]. Chen and Chung improved Xu et al.'s method by using the randomized selection of four pixels [3].

Olson considers methods to improve curve detection by decomposing the HT into many small sub-problems. He used randomized to limit the number of sub-problem that he must examine and he carefully propagate effects of location error in the sub-problems that he do examine, which concentrated on curve detection, similar HT techniques can be applied to surface detection and a number of other problems [10]. since the computational complexity of HT is very large many approaches has been presented to speed up the computation such as randomized HT (RHT) [2].

Fung and et. al were presented an algorithm for object detection. It combines the advantages of both GHT and RHT, hence it was named Randomized Generalized Hough Transform (RGHT) [6].

Ji and Xie were introduced a new probabilistic HT that was aimed at improving the accuracy and robustness of the HT by explicitly accounting for the errors with the image pixels and error with the estimated curves parameters [7].

Yu and Bajaj described an automatic method for circle detection, their method was based on shape (circle) matching between the edge map for original image and the template shapes (circles) of the true particles (called virtual circles) [15].

Chiu and Liaw were presented a voting method to reduce the computations and the storage requirements of standard HT for circle detection [4]. This method improves

the efficiency of circle detection by letting each pixel only belong to one candidate of circle parameters. Though this approaches reduced heavy computational burden, other problems have still remained [11].

2- Overview proposed algorithm:

Concerned predefinitions and procedures for the some steps in proposed algorithm are declared in this section.

2-1 Inspection point:

Inspection point is the first white pixel detected during the main scanning (horizontally scanning) and it assumed to be the top point on a circle circumference called (P1) and then find its couple point (P2) on the opposite side of the circumference which will be declared in section (3).

2-2 Finding the radius and the center coordinates:

Any circle has two opposite points on its circumference , like the inspection point P1(x1,y1) and its couple P2(x2,y2), we assume that the straight line between P1 and P2 is passing through the center of this circle, then the diameter of this circle is equal to the distance (d) between P1 and P2, and can be find according to the following equation:

$$d = \sqrt{(x1 - x2)^2 + (y1 - y2)^2} \quad \ldots(1)$$

Then the radius (r) = d/2

To find the center of the circle assume that C(xc,yc) is the center coordinates for the virtual circle.

If P1 and P2 are on the same column then the values of y1, y2 are equal, therefore:

xc=x1+radius.

yc=y1.

If P1 and P2 are on the same row then the values of x1,x2 will be equal, then:

xc=x1.

yc=y1+radius.

2-3 Template of virtual circle:

Depending on the radius and the center coordinates which obtained in section (2-2), begin making the template of the virtual circle by determining some of distributed points on the circumference of the circle by using the equations (2)&(3) below, see figure (1).

xi=xc+r*cos(*q* i) **…(2)**

yi=yc+r*sin(*q i*) **…(3)**

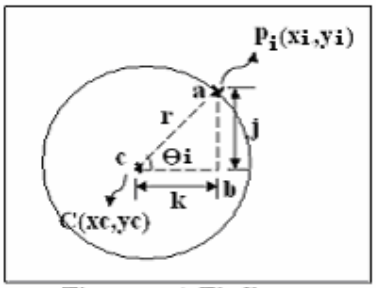


Figure (1):Finding p_i

In this section, the circumference of the virtual circle will be completed as a sample which will be used to match with the corresponding circle in the original image.

2-4 Matching process:

Template matching is a common image processing method to determine the location of an object in an image using a template [14], or template matching is the process of finding the location of sub-image, called a template, inside an image. Template matching involves comparing a given template with windows of the same size in an image and identifying the window that is similar to the template [5].

To compare the template points (virtual circle) with the original image, compare each point of the template with the corresponding point in the original image and its eighth neighbors to avoid noise of shifting by one pixel of some points if exist. The mask (3*3) as shown in the figure (2) is used for this purpose.

1	1	1
1	1	1
1	1	1

Figure(2): The Mask.

Any point from the original image match the corresponding point in the template or its eighth neighbors (the mask) it will be account, and so for the rest points of the template.

2-5 Removing the circle:

If the virtual circle match the circle in original image, remove the detected circle from the original image by delete the circumference's point and eighth neighbors of each point to insure that the hole points of the detected circle are deleted, and to reduce the unneeded process (i.e. the circles are detected one time only which reduces the running time by reducing the matching processes and avoid detecting same circles many times).

3- Approach of proposed algorithm:

The steps of the new algorithm (PSCD) for circle detection are declared in the figure (3) and stated below:-

1. Begin main scanning horizontally pixel by pixel from top left corner of the image until getting the inspection point. else go to the step (11).
2. With getting the inspection point, begin the vertical scanning pixel by pixel from the inspection point **P1(x1,y1)** until getting the first white pixel **P2(x2,y2)**, else go to the step (1) and resume the horizontal scanning.
3. Find the distance (d) between P1&P2 by using the equation (1).
4. Find the radius (r) of the virtual circle (assumed).
5. Find the center coordinates C(xc,yc).
6. Check the pixels P3(xc-r,yc) and P4(xc+r,yc) if one or both of them is white then continue. else neglect p2 and go to the step (2) and resume the vertical scanning.
7. Find the template of the virtual circle -circumference's points.
8. Matching:
   Compare the template points with the original image as in section (2-4), If the total matching is (50%) or more (upon user) the virtual circle will consider as real and

exist circle in the original image, and will store its radius and center coordinates in a file as a detected circle.

Else for total matching less than (50%) neglect p2 and go to step (2) and resume vertical scanning.

9. Remove the detected circle from the original image.
10. Go to the step (1) and resume the horizontal scanning.
11. Terminate.

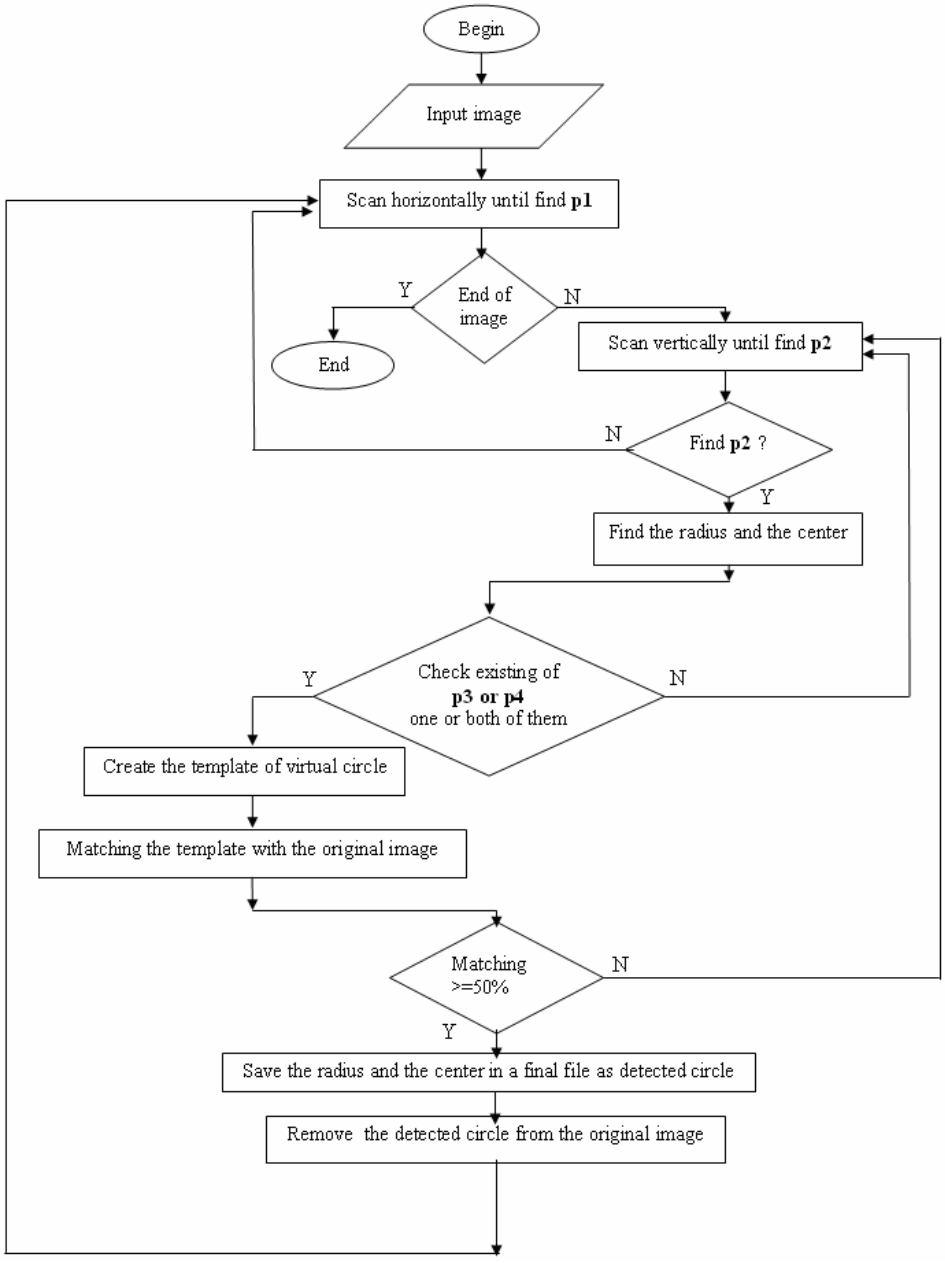


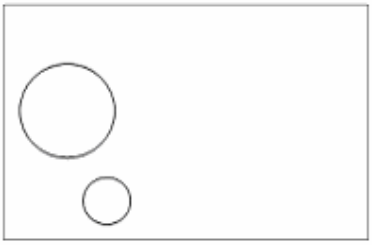
Figure (3): Flow chart of proposed algorithm

4- Discussion the proposed algorithm by examples:

In this section the proposed algorithm will be declared by applying many examples according to the algorithm steps.

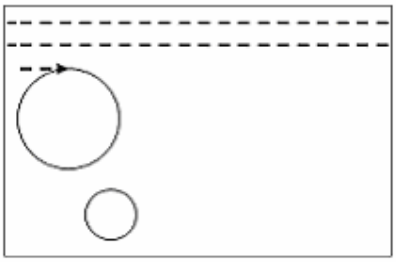
Example (4.1):-

The original image is shown in figure (4)



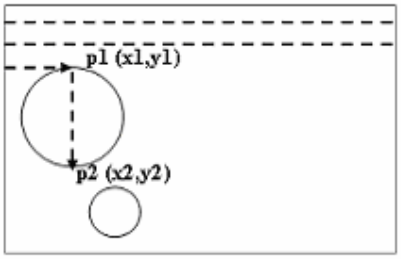
Figure(4)

In figure (5) the horizontal scanning until finding inspection point **p1** is shown.



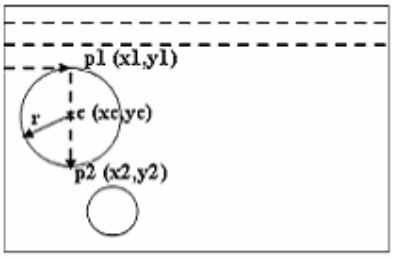
Figure(5)

See the beginning of vertical scanning until finding **p2** in figure (6).



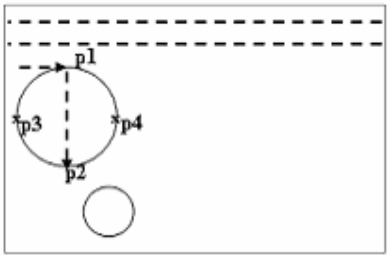
Figure(6)

Figure (7) explain finding the radius (r) and the center coordinates c(xc,yc).



Figure(7)

In figure (8) checking of existing **p3** or **p4** is shown.



Figure(8)

Figure (9) shows creating of the template of virtual circle and matching process between the template and the circle in the original image.

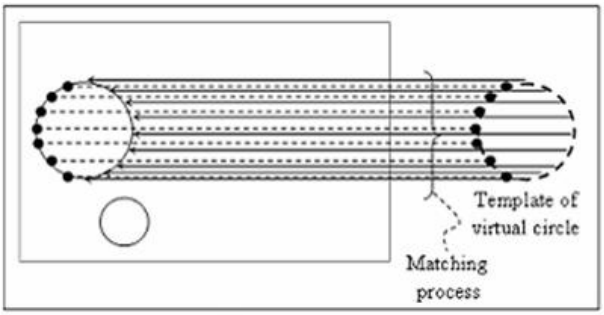


Figure (9)

In figure (10) the original image shown after removing the first detected circle.

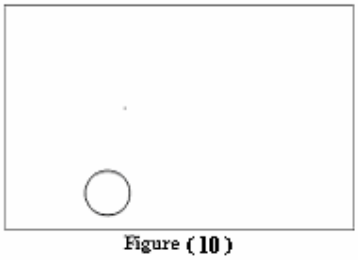
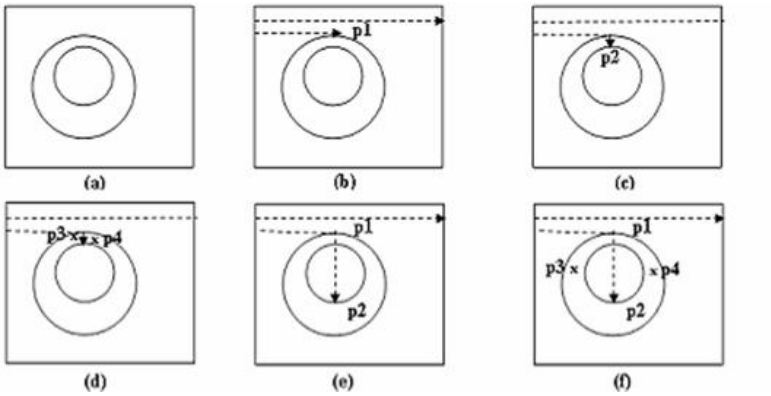


Figure (10)

The algorithm will continue on the image in figure (10) to detect the other circle.

Example (4.2):-

In this example nested circles will be used to apply the algorithm as follows:-



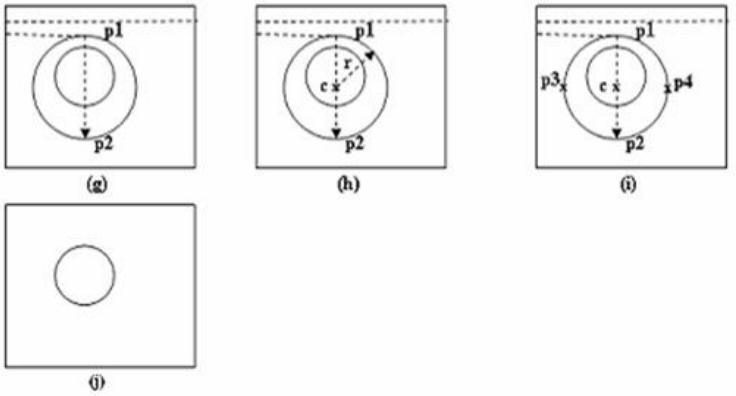


Figure (11): example (4.2)

Figure (11) shows the steps of the algorithm when applying on this example where (a) represents the original image, (b) declares the horizontal scanning to find inspection point p1(x1,y1), (c) explains the vertical scanning to find p2(x2,y2).Then after finding the radius (r) and the center coordinates c(xc,yc) for the expected circle find p3(xc-r,yc) & p4(xc+r,yc) and check there existences as shown in (d) but the points p3&p4 are not exist in the original image, in this case neglect p2,p3, and p4and resume the vertical scanning to find new p2 as shown in (e). Again find the radius and the center coordinates then find new p3&p4 and check there existence (f) shows that, p3&p4 are not exist, therefore neglect p2,p3, and p4. Resume vertical scanning to find new p2 as shown in (g), then find the radius and the center coordinates for the new expected circle see (h). In (i) shows checking existence of p3 or p4 for the new expected circle, they are both exist, this existence indicates that this expected circle is more promising to be exist. Therefore the algorithm will go on for this virtual circle to find the template and matching it with the circle in the original image, then remove the detected circle as shown in (j), and algorithm will continue to find the other circles if exist.

Example (4.3):-

In this example there are crossed circles, the steps of application of the algorithm are declared in figure (12), the original image is shown in (a), where (b) shows the horizontal scanning to find p1, (c) shows the vertical scanning to find p2, finding the radius and the center coordinates are seen in (d), in (e) finding p3&p4 and checking there existence, they are exists. Therefore the algorithm steps will continue to find the template of the virtual circle and matching it with the circle in the original image, in this example matching occur more than 50% and the circle will be detect and will removed as shown in (f). Then the algorithm will continue to detect another circle.

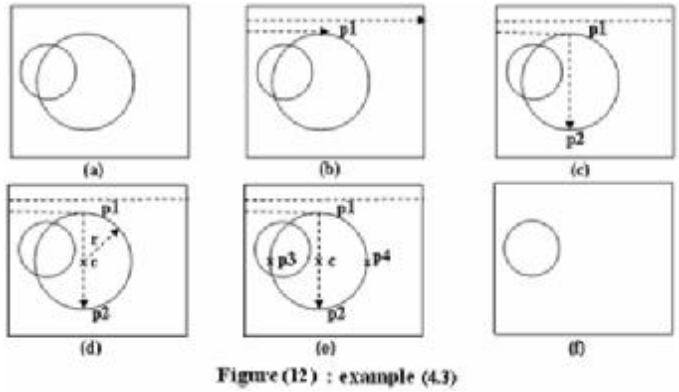


Figure (12) : example (4.3)

5- Efficiency of the Proposed Algorithm and Asymptotic:

The main factors in measuring the efficiency of an algorithm are the running time and the space required for the program. There are many factors that affect the running time of the program, among these are the program itself, the input data, and the computer system used to run the program. By fixing the computer factors focus should concentrated on the input data and the program. Some input data distributed so that program runs faster (called Best case), other distributed so that the program runs slowest (called Worst case), the concentration is on the determining the running time for the worst case data.

Now for the program role, the focus should be on the operations that affect on the running time which they are the loops. The concept of (Big-O) rise here to determine the complexity of the program, which simplify the counting of the operations by dropping the all lower order terms, eliminate the constants, and the remaining term is the (Big-O). [18]. The time complexity of a program is the number of steppes that it takes to solve an instance of the problem as a function of the size of the input, using the most efficient known algorithm. If an instance has length n and can be solved in n^2 steps we can say the problem has a time complexity of n^2 [17]. Asymptotic analysis is based on the idea that as the problem size grows, the complexity will eventually settle down to a simple proportionality to some known function [1].

To compare the proposed method with another related common method we apply Hough Transform (HT) method for circle detection by using improved and efficient program for HT [16] , a database of about 40 different images was used, each image contains one or more circles of different radius and some cases contain other shapes and noise. As it known HT reserves 3D matrix [R*L*Z] to store circumference points of the detected circles. HT detected a large number of pseudo circles for each figure.

Table (1): comparing between HT & PSCD

Fig.* No.	Dimensions Of Image	No. of White pixels	HT			PSCD			Running Time***** Ratio
			Memory** required (Location)	No. of Detected Circles (Radius>2)	Running*** Time (Second)	Memory required**** (Location)	No. of Detected Circles (Radius>2)	Running Time *** (Second)	
1	221*286	610	221*286*286	906201	37.624	2*37	1	0.084	1:447
2	254*255	771	254*255*255	1291345	44.855	2*37	2	0.091	1:493
3	242*241	1321	242*241*242	2845282	70.888	2*37	7	0.155	1:457
4	221*143	1700	221*143*221	1958472	79.252	2*37	20	0.166	1:477
5	221*286	3526	221*286*286	8233920	207.231	2*37	41	0.402	1:515
6	221*286	3647	221*286*286	8556385	215.512	2*37	40	0.637	1:338
7	254*292	4416	254*292*292	11079840	447.153	2*37	72	0.660	1:678
8	442*286	7052	442*286*442	23884570	1940.734	2*37	82	2.670	1:726
9	302*480	7194	302*480*480	Out of memory	2347.936	2*37	100	1.077	1:2180
10	442*286	7880	442*286*442	25692514	2586.872	2*37	74	16.975	1:152

* See appendix A.
** 3D matrix [R*L*Z].
*** Using tic-toc instruction in Matlab.
**** To save the points(x, y)of the template for virtual circle=37 point.
***** Running Time of PSCD : Running Time of HT.

From the table (1) it seen that the shortcoming points of HT are: it required a large amount of space for its 3D matrix, it detects a large number of circles, HT needs a large amount of running time because a large number of computational processes should apply on each white pixel in the image (i.e. HT running time depends on number of white pixels see figure (13)), to extract the real circles from the produced matrix it needs to know previously the number of circles in the original image to choose them and neglect the rest large number of detected circles, this point makes HT to lose its reliability and its efficiency in the practical application field, it detect the other shapes

and the noise as circles, beside all that HT needs to extra efficient method to extract the real circles from the produced matrix.

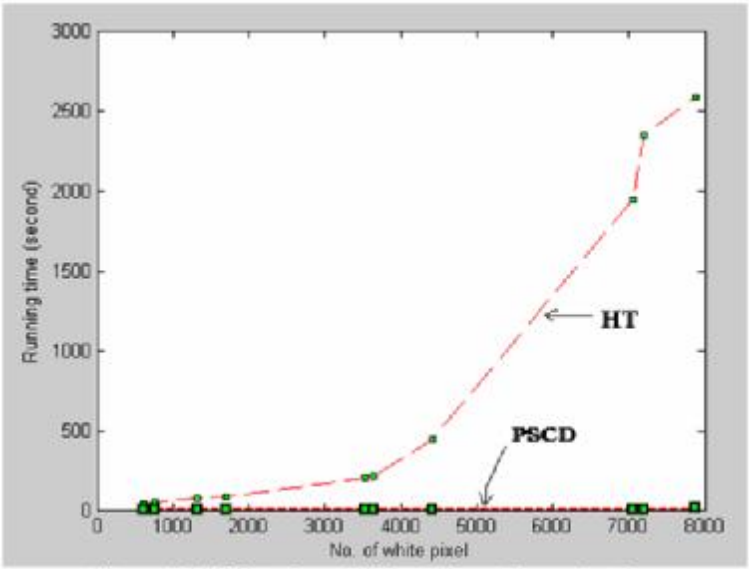


Figure (13): Affect of no. of white pixels on the running time

The proposed method required just very little amount of space, it detects exactly the perfect number of circles (100% accuracy), There is a little computations (at first if it detects two points on the circle circumference then apply the other steps) that means little running time see figure (13), it needs neither previous knowledge of number of circles in the image, nor extra method to extract real circles, because the circles are directly detected. It not detects the other shapes and the noise as circles see table (1).

The proposed method needs to two nested loops as shown below:

```
loop1  horizontal scan   //to find p1.
    loop2 vertical scan    //to find p2.
       if exist circle       // if p3 or p4 exist.
          create template.      //calling function
          matching process   //calling function
          if matching
             delete circle.       //calling function
          end if
       end if
    end loop2
end loop1
```

where loop1 represents the number of inspection points (p1), loop1 needs n times to be completed, loop2 represents the number of points that lies on the same column of inspection point beneath it (p2), loop2 needs m times. The operations (create template, matching and remove the detected circle) are calling functions and they are needs constant time (c). Therefore the Big-O of the proposed method is O(n*m).

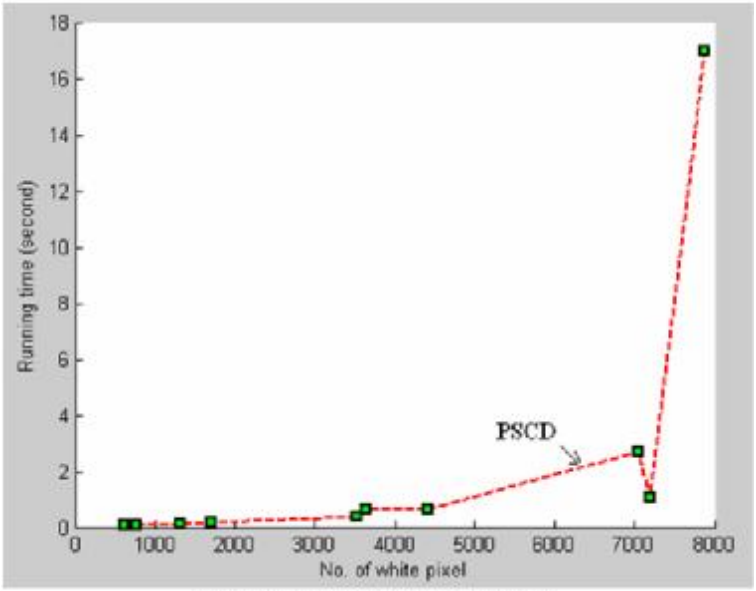


Figure (14): Running Time of PSCD

6- Conclusions:

1. The proposed algorithm can detect circles and parts of circle which are greater than half circle, but the parts less than half circle can't detected by this algorithm.
2. The proposed algorithm does not need the previous knowledge about the number of circles in the image.
3. It needs a very little amount of spaces.
4. The size of image (its dimensions) is not affected in the proposed algorithm.
5. Other shapes in the image like noise, triangles, rectangles,...etc. not affect on the proposed algorithm (not detected as circles), but they takes additional time, see figures (5) & (6) and figures (8) & (10) in appendix A and their running time in the table (1) and see figure (14).
6. The size of the circle is not affect on the proposed algorithm.
7. The running time of the proposed algorithm increased by increasing the number of circles.
8. The accuracy of the proposed algorithm was 100%, see table (1) and appendices A and B.
9. The proposed algorithm high speed, where the ratio of running time of the proposed algorithm to running time of HT for an image included 100 circles was 1:2180. And the average ratio for all tested images was 1:646.
10. From table (1), although figure (8) contains 82 circles but it takes running time more than figure (9) which contains 100 circles see figure (14). Therefore we conclude that existence of crossed or nested circles in the image will leads to increasing running time.

REFERENCES

[1] Anastasio, T. ,(2000) "Asymptotic analysis", 9 Feb.. www.asymptotic.analysis.mht.

[2] Chen, L.; Chen, H.; Pan, Y.; Chen,Y., (2004) "AFast Efficient Paralle; Hough Transform Algorithm on LARPBS"; The journal of Supercomputing, 29 , pp.185-195.

[3] Chen, T., and Chung, K., (2001) "An efficient randomized algorithm" Computer vision and image understanding, vol.63, no.83, pp. 172-191. From (11).

[4] Chiu, S.; Liaw, J., (2005) "An effective voting method for circle detection"; Pattern recognition letters, vol. 26, Issue 2, pp. 212-133, 15 January.

[5] Ding, L.; Goshtasby, A.; Satter, M.;"Volume image registration by template matching"; Image and vision computing, vol. 19, pp. 821-832, 2001.

[6] Fung, P.; Lee. W.; King, I.; "Randomized Generalized Hough Transform for 2-D grayscale object detection";(via Internet).

[7] Ji, V.; Xie, Y., (2003) "Randomized Hough transform with error propagation for line and circle detection"; Pattern anal applic, pp. 55-64.

[8] Le, D.; Thoma, G., (1993)"Document Skew Angle Detection Algorithm";Proc.1993 SPIE symposium on aerospace and remote sensing-visual information processing II, Orlando, FL vol. 1961, pp. 251-262, April 14-16.

[9] Mira, J.; Mayer, J., (2003) "Image feature extraction for application of biometric identification of iris-a morphological approach"; proceedings of the XVI Brazilian symposium on computer graphics and image processing, IEEE.

[10] Olson, Clark F., (1996) "Decomposition of the Hough transform: Curve detection with efficient error propagation"; Proc. of the European conference on computer vision, vol.1, pp.263-272.

[11] Rad, A.; Faez, K., Qaragozlou, N., (2003) "Fast circle detection using gradient pair vectors"; Proc. VIIth digital image computing: Techniques and applications, Sydney, 10-12 Dec.

[12] Tsuji, S.; Matsumoto, F., (1978) "Detection of ellipses by a modified Hough transformation", IEEE transactions on computers, vol.C-27, no.8, pp. 777-781. From (11).

[13] Xu, L., Oja, E.; Kultanan, P., (1990) "Anew curve detection method: randomized Hough transform (RHT)", Pattern recognition letter, vol.11, no.5, pp. 331-338. From (11).

[14] Yi Su, R.; "Seed image reconstruction using a template matching technique"; Biomedical image resource, Mayo clinic college of medicine, Rochester, MN, USA 55905,(via Internet).

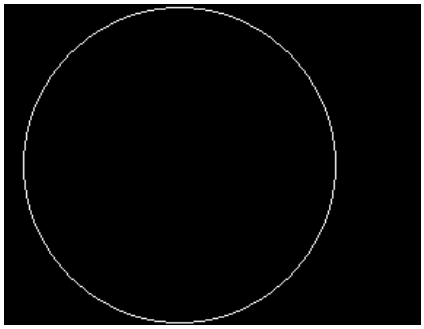
[15] Yu, Z.; Bajaj, C., (2004) "Detecting circular and rectangular particles based on geometric feature detection in electron micrographs", Journal of Structural Biology 145, pp. 168-180.

[16] http://scilab-imageprocessing.blogspot.com/2009/04/hough-transform-for-circle-detection_5899.html

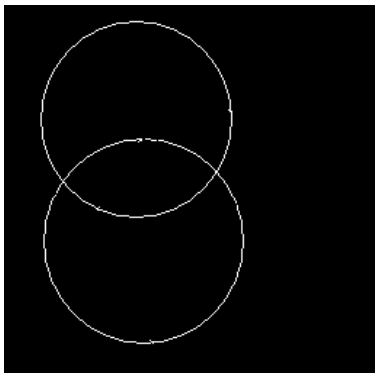
[17] http://www.answers.com/topic/computational-complexity-theory.

[18] http://www.progressive-coding.com.

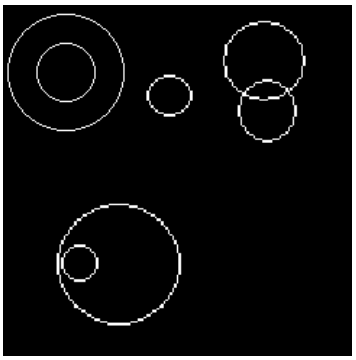
APPENDIX A



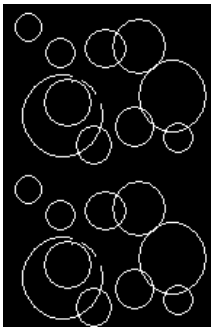
Figure(1)



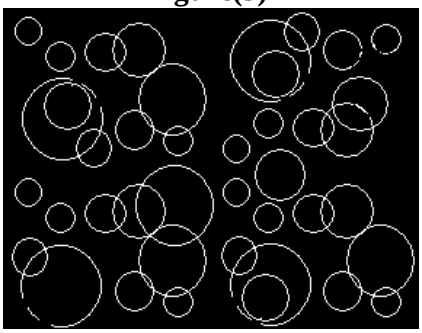
Figure(2)



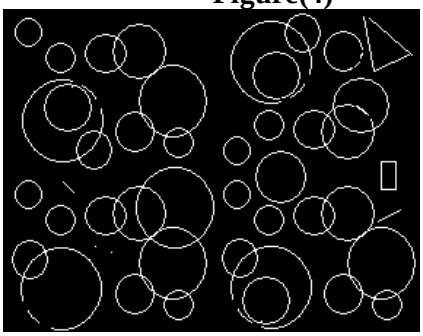
Figure(3)



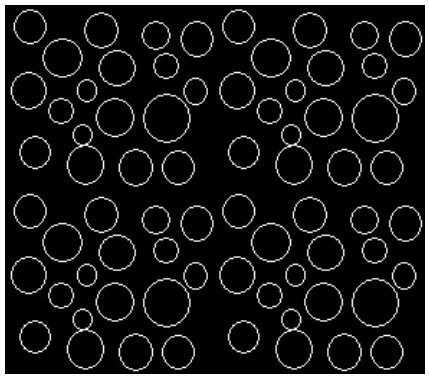
Figure(4)



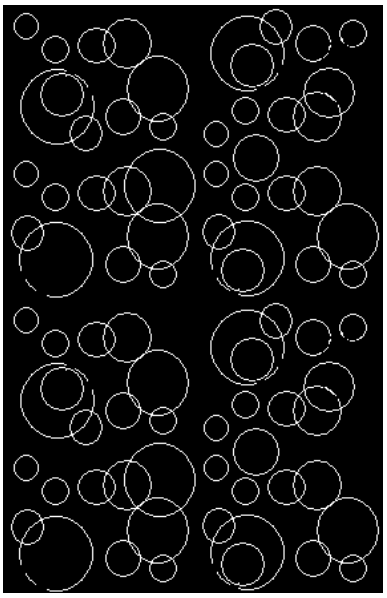
Figure(5)



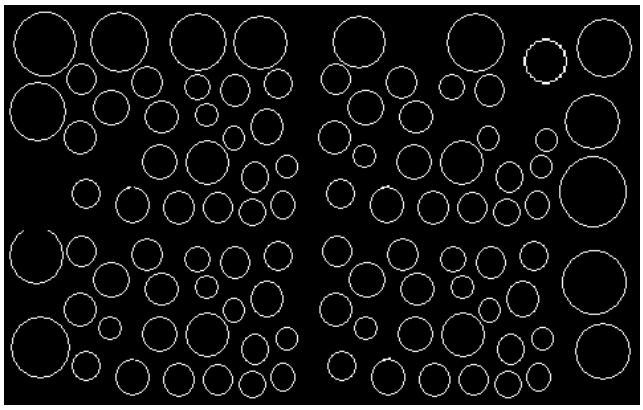
Figure(6)



Figure(7)



Figure(8)



Figure(9)

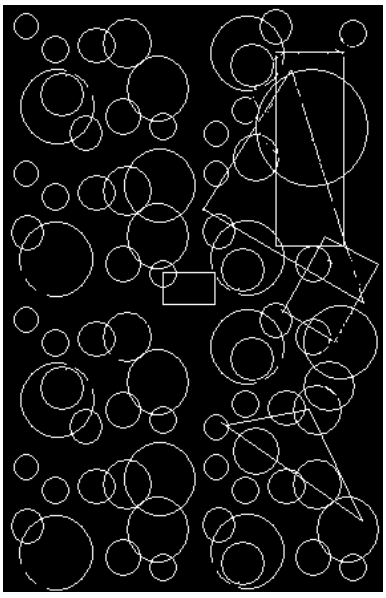


Figure (10)

APPENDIX B

Table of the results of figure (9) for PSCD method.												
xc	yc	radius	xc	yc	radius	xc	yc	radius	xc	yc	radius
81	26	21	63	147	10	101	250	12	29	357	22
190	25	20	193	147	10	231	251	12	214	347	9
259	28	22	120	154	16	57	251	11	154	354	11
30	32	24	250	154	16	187	252	11	284	355	11
101	58	12	84	154	9	143	255	11	65	367	11
231	58	12	214	154	9	273	256	11	195	368	11
57	59	11	154	162	11	29	269	20	101	366	8
187	59	11	284	162	11	78	274	14	231	367	8
143	63	11	65	175	11	208	275	14	157	380	10
273	63	11	195	175	11	115	273	9	287	381	10
29	88	22	101	174	8	245	274	9	131	382	10
78	82	14	231	174	8	152	290	13	261	383	10
208	82	14	29	194	20	282	291	13	223	392	12
245	81	9	157	188	10	59	301	12	190	401	11
152	98	13	287	188	10	189	302	12	43	409	16
282	98	13	131	190	10	119	310	13	152	403	9
59	109	12	261	190	10	249	311	13	282	404	9
189	109	12	93	199	12	85	312	13	123	406	8
119	118	13	223	199	12	215	313	13	253	407	8
249	118	13	60	208	11	154	325	12	103	410	8
85	120	13	190	208	11	284	326	12	142	445	25
215	120	13	152	211	9	63	339	10	210	446	24
154	133	12	282	211	9	120	346	16	89	445	21
284	133	12	123	214	8	193	340	10	262	453	21
29	147	21	253	214	8	250	347	16	33	454	21