Improvement of Fast Circle Detection Algorithm Using Certainty Factors

Ali Ajdari Rad
Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
alirad@aut.ac.ir

Karim Faez
Professor of Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran
kfaez@aut.ac.ir

Navid Qaragozlou
Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
eavidq@aut.ac.ir

Abstract

The Circle Hough Transform (CHT) has become a common method for circle detection in numerous image processing applications. Because of its drawbacks, various modifications to the basic CHT method have been suggested. In this paper, an algorithm is presented to find circles which are totally brighter or darker than their backgrounds. The method is size-invariant, and such circular shapes can be found very fast and accurately called Fast Circle Detection (FCD). Though Fast Circle Detection (FCD) method loses the generality of the CHT, there are many applications that can use this method after a simple preprocessing and gain a considerable improvement in performance against the CHT or its modified versions. In this paper, we present a method based on certainty factor concept to improve the FCD performance. The improved version of the FCD has been evaluated in some famous industrial and medical fields, and the results show a significant improvement.

Keywords

Circle Detection, Hough Transform, Certainty Factors.

INTRODUCTION

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. The Circle Hough Transform (CHT) [1] is the best-known algorithm and aims to find circular shapes with a given radius r within an image. Usually the edge map of the image is calculated then each edge point contributes a circle of radius r to an output accumulator space. For circles with unknown radiuses, the algorithm should be run for all possible radiuses to form a 3-dimensional parameter space, where two dimensions represent the position of the center, and the third one represents the radius. The output accumulator space has a peak where these contributed circles overlap at the center of the original circle.

In spite of its popularity owing to its simple theory of operation, the CHT has some disadvantages when it works on a discrete image. The large amount of storage and computing power required by the CHT are the major disadvantages of using it in real-time applications. Some modifications have been reported to increase the CHT performance so far. Tsuji and Matsumoto decomposed the parameter space by using the parallel property of circle [2]. Several methods use randomized selection of edge points and geometrical properties of circle instead of using the information of edge pixels and evidence histograms in the parameter space. Xu et al. [3] presented an approach that randomly has selected three pixels. The method selects three noncollinear edge pixels and votes for the circle parameters which are found by using the circle equation. Chen and Chung [4] improved Xu et al.'s method by using the randomized selection of four pixels. However, the randomized selection method has its own problems such as probability estimation, accuracy and speed that are dependent on the number of edge pixels. Yip et al. [5] proposed a method which has reduced the parameter dimension, but estimated the parameters of the circles based on local geometrical properties which often have suffered from poor consistency and location accuracy due to quantization error. To overcome these disadvantages, Ho and Chen [6] used the global geometrical symmetry of circle to reduce the dimension of the parameter space. The UpWrite method used the spot algorithm to produce the local models of the chosen edge pixels [7]. The chosen edge pixels are those in a circular neighborhood of the parameter radius r centered on the chosen edge pixels. Also there has been many works to make the CHT size-invariant [8]. Though these approaches reduced heavy computational burden, other problems have still remained.

With some prior knowledge about an image, we can simplify the CHT and reduce the algorithm’s difficulties according to especial features of the problem. Use of edge orientation information was first suggested by Kimme et al. [9], who noted that the edge direction on the boundary of a circle points towards or away from the circle’s center. This modification reduced computational requirements as only an arc needed to be plotted perpendicular to the edge orientation at a distance r from the edge point. In the limit, arcs may be reduced to a single point in the accumulator space. Minor and Sklansky [10] extended the use of edge orientation by plotting a line in the edge direction to detect circles over a range of sizes simultaneously. This has an additional advantage of using a two rather than a three-dimensional parameter space. The method can be further extended to the detection of circle-like shapes (compact convex objects, or blobs).

The FCD method [11] got the main idea of edge orientated methods, and presented an algorithm to find circular shapes that were totally brighter or darker than their background. According to the edge oriented methods, the edge direction on the boundary of a circle points towards or away from the circle's center. The condition of being totally darker or brighter than background forces all edge directions of a circle to be outward or inward, so the FCD can use the symmetry property of the gradient vectors to find vector pairs. These founded vector pairs are used to improve the circle detection algorithm. In this paper, we present a method based on certainty factor concept to improve the FCD performance. For each founded vector pair, some certainty factors are considered and final decision about location of founded circles is made according to these parameters. Using such certainty factors, the FCD is speed up with more accuracy and more robustness against noise.

The rest of the paper is organized as follows. Section 2 describes the FCD algorithm in details and section 3 presents improved method. Section 4 shows experimental results comparing with the CHT and original version of the FCD. Finally, in section 5, we describe the conclusion and our plans for future work.

FAST CIRCLE DETECTION USING GRADIENT VECTOR PAIRS

In this section, we present a fast circle detection algorithm based on gradient vector pairs. Suppose that we have a dark circle on a bright background[1], as shown in Figure 2.a. The following paragraphs describe the major steps of our algorithm.

The first step is calculating the gradient of the image. The gradient vectors of the circle we search for are in the form shown in Figure 2.b. These vectors' directions are outward the circle, because the circle is darker than its background. Due to the symmetry of circle, for each gradient vector there is another gradient vector in its opposite direction. We call these vectors vector pair. As shown in Figure 3.a, a specific vector V_1 is paired with a vector V_2 if the following two conditions are satisfied:

(i.) Angle α, defined as the absolute difference between directions V_1 and V_2, should be nearly 180 degrees.

(ii.) Angle β between the line connecting P_2 to P_1 (the bases of V_2 and V_1) and the vector V_1 should be nearly 0 degree[2] (This means $\overrightarrow{P_2P_1}$ that should be in the same direction as V1).

The second step of the algorithm is applied to find all vector pairs according to the above conditions in the gradient image. The second condition considerably removes noise by filtering useless vectors. As Figure 3.b shows, vectors V1 and V2 are not assumed as a vector pair due to condition (ii); however they satisfy condition (i).

To increase the speed of pair matching, vectors are sorted according to their directions. So, for each specific vector, vectors with opposite direction can be found fast and easily.

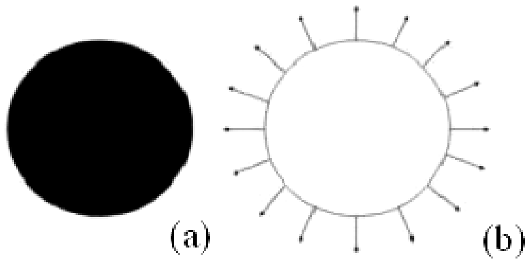

Figure 2. (a) A black circle in white background, (b) Gradient vectors of (a)

In the third step, a candidate circle is considered for each pair of vectors. Such a circle has its center at the midpoint of P_1 and P_2, and its radius is equal to half of the distance between P_1 and P_2. Figure 3.a shows such a candidate circle in dashed lines. In special cases, if the approximate radius of the desired circle is known, a third condition can be used to filter out those vector pairs whose distances are outside the range of the expected values. This can improve the performance of algorithm significantly.

In the fourth and final step, the desired circles are extracted from the candidate circles produced in the previous step. There are two ways to do this. One way is employing a 3-dimensional accumulator matrix to count the occurrence of quantized circles. Then, the desired circles can be found by searching for local maxima in such a space. This is just like the classic CHT approach.

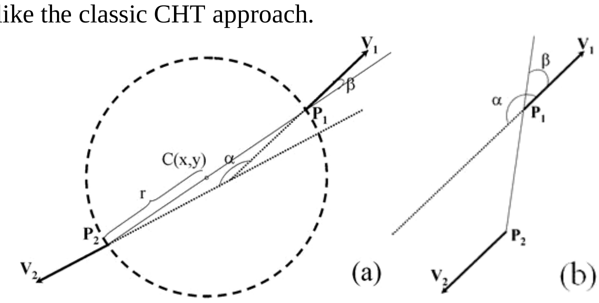

Figure 3. (a) Pair vectors (V_1 & V_2) and their candidate circle, (b) vectors rejected by condition (ii)

As the candidate circles are known, a second and easier approach can be used to find the desired circles. Candidate circles are saved as a set of triples (C_x, C_y, r). These triples are then clustered using Euclidian distance between them. The means of clusters then specify the desired circles. The method reduces the space complexity and optimizes the entropy of the saved data. Also, prior knowledge about the number of the circles in the image can be used to get better results. In our algorithm, we use the second approach.

[1] We can assume this without loss of generality because if the circle is brighter than its background, we can work on the negative image or simply reverse the direction of vectors.

[2] Or it should be in the opposite direction of V_2, because they are nearly parallel according to condition (i.)

Adjusting parameters of algorithm is critical and depends on image features and statistics. By increasing α and β, more pair vectors will be found. This may lead to better result (robustness against noise) or worse result (Finding more wrong pair vectors). So adjusting these parameters is an art in each application.

To improve the algorithm performance in real applications, some preprocessing on the input image may be necessary. As mentioned earlier, the algorithm searches for circles that are brighter or darker than their background. To obtain such image, some preprocessing can be done based on the problem features. Figure 4 shows an output result of the original FCD when 25% pepper-salt noise is applied to the image.

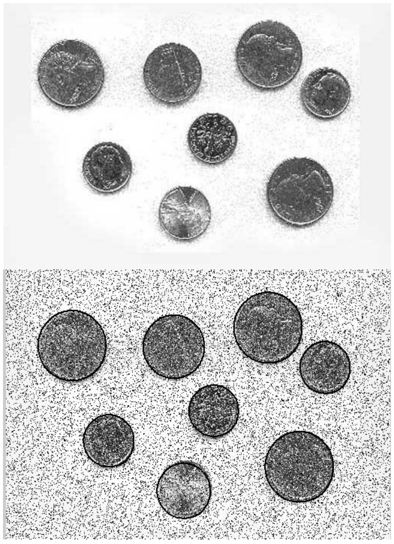

Figure 4. Output result of the FCD when 25% pepper-salt noise is applied to the image

THE FCD IMPROVEMENT USING CERTAINTY FACTORS

In this section, we describe how the performance of the FCD algorithm can be improved using certainty factors.

According to the original FCD, two vectors make a pair if they satisfy conditions i and ii of previous section. In real applications, these conditions are rarely satisfied. So a range of acceptable values should be used for α and β. If the deviations of α or β from their ideal value (180 and 0 degrees) become more than a threshold value, the vector pair is omitted and if both parameters place in range, the candidate circle is considered. In contrast with this binary decision, two certainty factors can be considered according to mentioned angles:

$$C_\alpha = \exp(-\frac{(\alpha - 180)^2}{\sigma_\alpha^2}) \text{ and } C_\beta = \exp(-\frac{\beta^2}{\sigma_\beta^2}).$$

These factors show the rate of satisfaction of i and ii conditions and can be used to control the behavior of the algorithm. Parameter σ can be used to adjust the effect of acceptable tolerance according to problem attributes.

Also, symmetry property of circle can be used to improve the performance of the FCD. By increasing the noise in the image the probability of matching of two random vectors as a vector pair increases. These wrong vector pairs increase error rate. Also, when the number of candidate circles is increased the clustering time is growth as well. Without loss of generality assume that the center of the candidate circle produced by v1 and v2 is placed at origin. According to figure 5, if a vector pair founded (suppose the bases of this vector pair are p1(x,y) and p2(-x,-y)), then 6 other points p3(y,x), p4(-y,x), p5(-x,y), p6(-y,-x), p7(y,-x) and p8(x,-y) should be placed on the same circle as well.

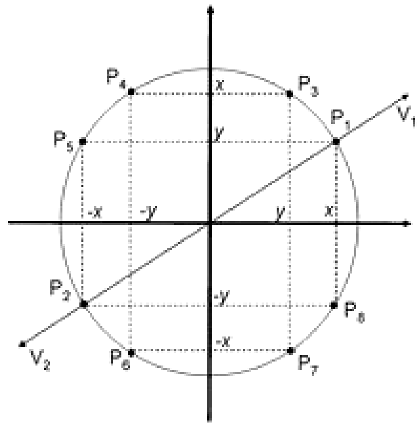

Figure 5. The pair vector and its related 6-points

To reduce the effect of noise, we can verify the existence of other 6 points in edge map of the image. If number of founded points is less than a specified threshold then the vector pair is omitted. In this way, random vectors that are not really placed on a circle are discarded. By omitting wrong candidate circles the number of candidate circles is reduced so the clustering step can be execute faster. If number of founded edge points is more than specified threshold then a candidate circle is considered just like the standard FCD. The number of founded points (n) also used to produce another certainty factor for the candidate circle denoted by C_8:

$$C_8 = \exp(-\frac{(n-8)^2}{\sigma_8^2})$$

In implementation, if either C_a, C_b, or C_8 become negative we omit the vector pair otherwise tree certainty factors used to make a final certainty factor according to the following formula:

$$CF = iC_\alpha + jC_\beta + kC_8$$

Parameters i, j, and k can be used to form different formulas according to problem features. For example for almost hidden circles, k parameter should considered near zero and for finding circles in noisy images, i parameter should be decreased. For the best adaptation to special problems, parameters, i, j, and k can be learned.

In clustering phase, to extract the desired circles, a weighted averaging is used based on certainty factors. Using weighted averaging causes more accuracy for the algorithm.

RESULTS

To test our proposed method, a database of about 100 different images is used. Each image contains one or more circles with different radiuses and some cases contain corrupted circles. The test images are collected from various real applications, with various subjects and also contain few hand made samples.

We structured our samples of database into four groups evaluate our algorithm more effectively than using random samples. The first group contained 100 original images. In the second group, binary images were produced from the original images by a simple gray level thresholding. Some pepper and salt noise and standard Gaussian noise were added to the original images to produce the third and the fourth categories. Two datasets with the same images (in the four groups) and with different image sizes (256 X 256 and 512 X 512) were produced and the evaluation was done on both of them to measure the effects of the increasing image size on the algorithm's performance. Also, the CHT (improved version) and the FCD algorithms were implemented to compare them with our proposed algorithm. The CHT is not a fast method, but it is a standard method. Almost all papers that presents a circle detection algorithm, implements the CHT and compare presented method with it. For the FCD versions both α and β parameters are set to 5 degree and gradient vectors are calculated using sobel operator and averaged in 5x5 windows and finally threshold by 30%. The All algorithms were implemented in Matlab 6.1, and the experiments were performed using a PC equipped with 1.8MHz Pentium IV processor and 512MB RAM. The average time of 10 trails that each algorithm used to produce the result for each image size and each image group is shown in Table 1.

Table 1. The results of execution time of the circle detection approaches (time is in second)

Image size				
256×256	Original	Binary	Pepper-Salt	Gaussian
CHT	205	172	319	262
FCD	0.30	0.26	0.32	0.33
FCD-CF	0.21	0.18	0.21	0.21
Image size				
512×512	Original	Binary	Pepper-Salt	Gaussian
CHT	1711	1567	2073	1932
FCD	1.58	1.44	1.66	1.60
FCD-CF	1.25	1.19	1.30	1.30

From Table 1, we find out that for 256x256 images, the proposed method is about one thousand times faster than the CHT and nearly 1.2 times faster than the original FCD. For 512x 512 size images, the proposed method still 1.2 times faster than the original FCD, and for the CHT it becomes more than 1600 times faster.

To compare accuracy of the FCD and proposed method, fifty different images were selected and the circle boundaries on them were determined manually. The error rate was calculated in three ways: Distance of the estimated centers from actual centers, difference between estimated radius and actual radius, and percentage of overlap of circles' surfaces.

Table 2 shows the final result of accuracy tests. Distance between centers and difference of the radii are normalized by the actual radius.

Table 2. The results of accuracy tests of the FCD

	Centers	Radii	Overlap
FCD	0.04	0.07	0.93
FCD with CF	0.02	0.04	0.96

Another evaluation is done to comparison the resistance of the FCD and proposed algorithm against noise. In this evaluation, the density of the pepper and salt noise was increased from 0 to 80 percent for each input image. Then error rates of these three approaches were calculated.

Results of this experiment show that proposed method resists against noise until 30% but the FCD error rate increases before 25% density. In this case the CHT just resists until 10%. After this threshold, the error rate increases exponentially for each approach. The CHT and FCD are totally failed after 30% and 52% noise but the improved method resists against noise until 70%. Figure 6 draws the error rates versus noise density for each approach.

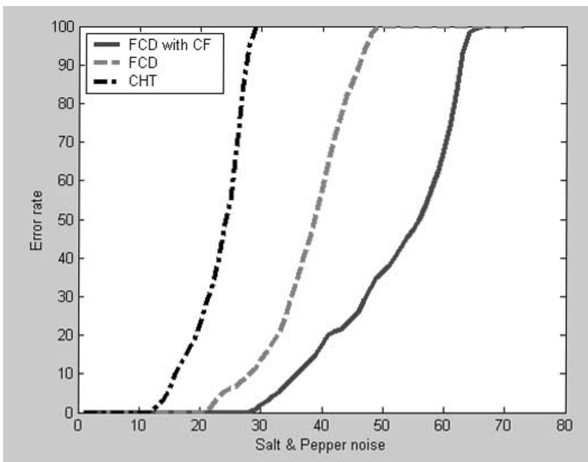

Figure 6. Resistance of the FCD and proposed algorithm against pepper-salt noise

CONCLUSION AND FURTHER RESEARCH

In this paper, we presented an improved version of a size invariant method to find circles that are totally brighter or darker than their backgrounds called Fast Circle Detection (FCD) method. The original idea is based on the symmetry of the gradient pair vectors on such circles. The improved method uses certainty factors for each vector pairs based on the parameters of the algorithm (α, β, and predicted edge points) to increase accuracy of the FCD algorithm.

The experimental results show that the proposed method is 1.2 times faster than the FCD and about a 1600 times faster then the CHT in case of 512x512 resolution images. The proposed method has better resistance against noise in comparison to the FCD and its accuracy is not affected until 30% pepper-salt noise is applied to the image (5% improvement in comparison to the original FCD). Also, the proposed method is more accurate than the original FCD. The improved method used for iris localization step in iris recognition applications, and the results show a significant improvement. Figure 7 shows an output proposed method for iris localization purpose [12].

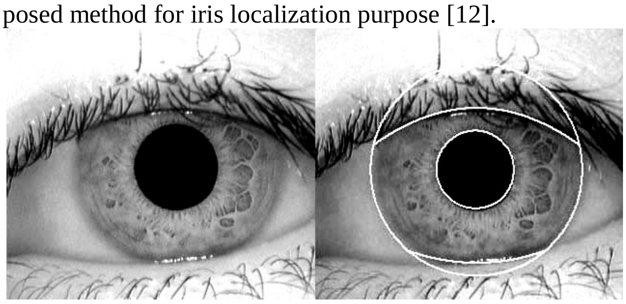

Figure 7. Output result of the FCD with certainty factors for iris localization

We plan to use the FCD and its improved version in other industrial and medical applications. Also, we plan to use the proposed algorithm to find known ellipses which have known ratio of diameters and directions. Figure 8 shows a simple result of the FCD to detect a known ellipse in MRI images of brain for fast brain boundary detection purpose.

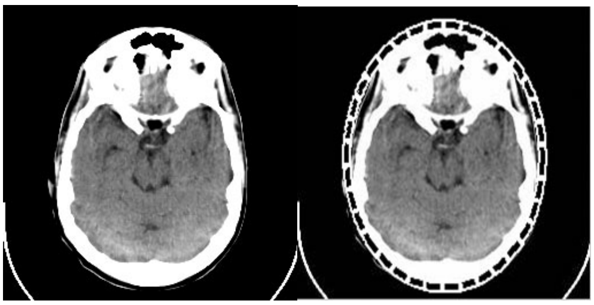

Figure 8. An example of ellipse detection for fast brain boundary detection in MRI images

REFERENCES

[1] Duda, R., and Hart, P., "Use of the Hough Transform to Detect Lines and Curves in Pictures", Communications of the ACM 15, pp: 11–15, 1975.

[2] Tsuji, S., and Matsumoto, F., "Detection of ellipses by a modified Hough transformation", IEEE Transactions on Computers, vol.C-27, no.8, pp: 777-781, 1978.

[3] Xu, L., Oja, E., and Kultanan, P., "A new curve detection method: randomized Hough transform (RHT)", Pattern Recognition Letter, vol.11, no.5, pp: 331-338, 1990.

[4] Chen, T., and Chung, K., "An efficient randomized algorithm", Computer Vision and Image Understanding, vol.63, no.83, pp: 172-191, 2001.

[5] Yip, R., Tam, P., and Leung, D., "Modification of Hough transform for circles and ellipse detection using a 2-dimensional array", Pattern Recognition, vol.25, no.9, pp: 1007-1022, 1992.

[6] Ho, C., and Chen, L., "A fast ellipse/circle detector using geometric symmetry", Pattern Recognition, vol.28, no.1, pp: 117-1995.

[7] McLaughlin, R., and Alder, M., "The Hough transform versus UpWrite," IEEE Trans, PAMI, vol.20, no.4, pp: 396-400, 1998.

[8] Atherton, T., and Kerbyson, D., "Size invariant circle detection", Image and Vision Computing, volume 17 pp: 795–803, 1999.

[9] Kimme, C., Ballard, D., and Sklansky, J., "Finding circles by an array of accumulators", Proc. ACM 18, pp: 120–122, 1975.

[10] Minor, L., and Sklansky, J., "Detection and segmentation of blobs in infrared images", IEEE Trans. SMC 11, pp: 194–201, 1981.

[11] Ajdari Rad, A., Faez, K., Qaragozlou, N., "Fast Circle Detection Using Gradient Pair Vectors", to be appear in 7th International Conference on Digital Image Computing: Techniques and Applications (DICTA'03), 2003.

[12] Ajdari Rad, A., Safabakhsh, R., Qaragozlou, N, "Fast Iris Localization Using Gradient Vector Pairs", submitted to First International Workshop on Biologically Inspired Approaches to Advanced Information Technology, Lausanne, Swizerland, 2004.