Proc. VIIth Digital Image Computing: Techniques and Applications, Sun C., Talbot H., Ourselin S. and Adriaansen T. (Eds.), 10-12 Dec. 2003, Sydney

Fast Circle Detection Using Gradient Pair Vectors

Ali Ajdari Rad[1], Karim Faez[2], Navid Qaragozlou[1]
[1] Computer Engineering Department, Amirkabir University of Technology, Tehran, Iran
{alirad, navidq}@aut.ac.ir
[2] Professor of Electrical Engineering Department, Amirkabir University of Technology, Tehran, Iran
kfaez@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. This paper presents an algorithm 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. Though Fast Circle Detection (FCD) method loses the generality of the CHT, we show that 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. This method has been evaluated in some famous industrial and medical fields, and the results show a significant improvement of finding circular shape objects.

1 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 one of the best-known algorithms and aims to find circular shapes with a given radius r within an image. Usually edge map of the image is calculated then each edge point contributes a circle of radius r to an output accumulator space. For unknown circle 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. 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 circles [2]. Several methods utilize 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 selects 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 circles 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 the added 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).

In our research, we get main idea of edge orientated methods [8, 9], and design an algorithm to find circular shapes that are totally brighter or darker than their background. Later, we will show that there would be many applications that satisfy such conditions using simple preprocessing. 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 we can use the symmetry property of the gradient vectors to improve the circle detection and we present an algorithm called Fast Circle Detection (FCD).

The rest of the paper is organized as follows. Section 2 describes our algorithm in details and section 3 presents experimental results comparing with the CHT and edge oriented version of the CHT. Section 4 assigned to present conclusion and our plans for future works.

2 Circle Detection Improvement Using Pair of Gradient Vectors

In this section we present our algorithm. Suppose that we have a circle in a brighter background[1] (Fig. 1.a). The following paragraphs describe the major steps of our algorithm.

The first step is calculating the gradient of image. The gradient vectors of the circle we search for is in the form of Fig. 1.b and has some attributes. The vectors' directions are outward the circle (from center of circle to out of it) because the circle is darker than its background. Because of center symmetry of circle, for each vector there will be a pair of vectors in opposite directions. As can be seen in Fig. 2.a, for a specific vector V_1, its pair vector V_2 is the one that satisfies the following two conditions:

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

(ii.) The 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 that $\overrightarrow{P_2P_1}$ should be in the same direction as V_1).

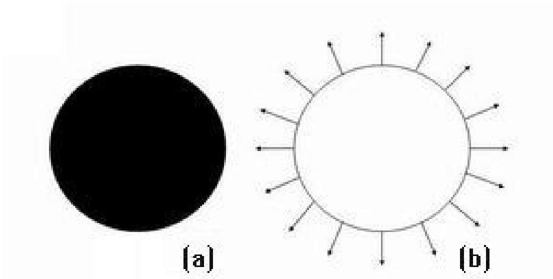

Fig. 1. (a) A black circle in white background, (b) Gradient vectors of (a)

The second step of algorithm is applied to find all pair vectors according to the above conditions in the gradient space of image. The second condition considerably removes noise by filtering useless vectors. As can be seen in fig 2.b, vectors V_1 and V_2 are not assumed as pair vectors due to condition (ii.); however they satisfy condition (i.) To increase the speed of pair matching phase, array of obtained vectors can be sorted according to direction of vectors so for each specific vector, the vectors that are in opposite directions can be found easily.

In the third step, a candidate circle is considered for each pair of vectors. Such a circle has its center, at midpoint of P_1 and P_2, and its radius is equal to half of the distance between P_1 and P_2. Fig. 2.a shows such a candidate circle by dashed and bold lines. In special cases, if the range of the desired circle's radius is known, a third

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

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

condition can be used to filter those pair vectors that their distance is outside the range of the expected radius. Our experiments show that if there is a little knowledge about the range of radius, the performance of algorithm will increase significantly.

In the fourth and final step, the desired circles are extracted from the candidate circles produced in 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 desired circles could be found by searching for local maximums in such a space. This is just like the classic CHT approach. As the list of candidate circles is known, the second and easier approach can be used to find exact circles. We can save candidate circles as a set of triples (C_x, C_y, r) and then cluster these triples according to Euclidean distance of them. In this way, desired circles can be found by using seeds of clusters and additional processing like mean vector calculation or cluster variance reduction. This approach shows a better performance comparing to the classic approach. This method also reduces the space complexity and in our experiments entropy of saved data was near optimum. Also clustering properties can be used to get better results like prior knowledge about the number of the circles in the image. So in our algorithm we use the second approach.

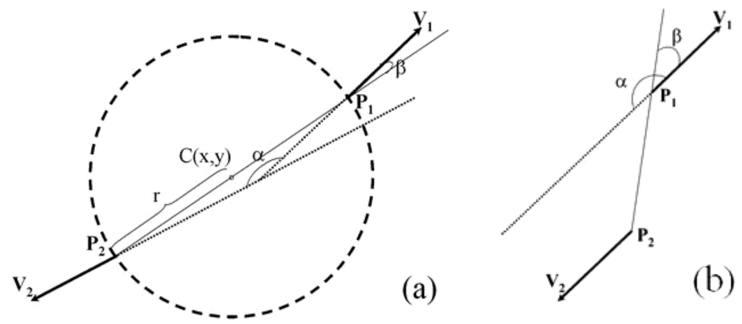

Fig. 2. (a) Pair vectors (V_1 & V_2) and their candidate circle (b) vectors rejected by 2nd condition.

Along with the four major steps, to employ the algorithm in real applications, some preprocessing should be done to make the input image suitable for the algorithm or to improve the algorithm's performance. 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. For example in the field of detecting pupil in eye images a Gaussian filter may be used to smooth the image, then an appropriate gray level thresholding should be done to make a binary image. In that binary image the pupil appears as a solid black circle and will have the desired conditions to be fed to our algorithm.

Other preprocessing methods that can be employed in various applications to improve the performance are: smoothing, edge map calculation, color classification, and cropping the useless part of an image.

3 The Experimental Results

To test our proposed method, a database of about 100 different images was used. Each image contains one or more circles of different radius and some cases contain corrupted circles. The test images are collected from various real applications and with various subjects and also contain few hand made samples. Fig. 3 shows some of such images and the result of applying the FCD algorithm on them. In some cases we need a simple preprocessing phase to make the image suitable for the input of the FCD algorithm.

To better evaluate our algorithm, we structure our sample database into four groups. The first group contains 50 original images. In the second group, binary images are produced from the original images by a simple gray level thresholding. Finally some salt & pepper noise and standard Gaussian noise are 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 (256×256 and 512×512) are produced and the evaluation is done on both of them to measure the effects of the increasing image size on the algorithm's performance.

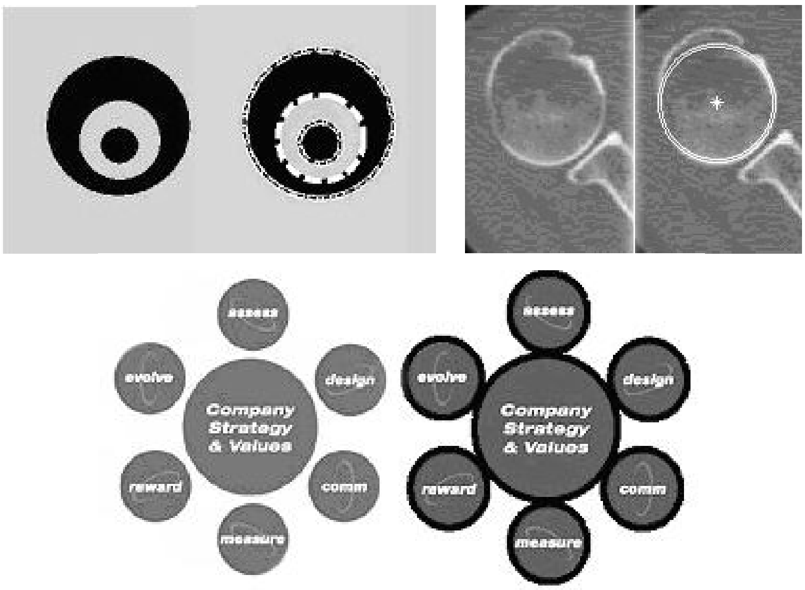

Fig. 3. Some outputs of the FCD algorithm

We also implement the CHT and the EOCHT algorithms to compare them with our proposed FCD algorithm. All algorithms are implemented in Matlab 6.1 and we attempt to reduce the side effects of the environment to obtain more reliable results. The average time that each algorithm used to produce the result is shown in Table 1 for each image size and each image group. The experiment was repeated for 10 trials and the average detection times are shown in Table 1.

The experiments were performed using a PC equipped with 1.8MHz Pentium IV processor and 512MB RAM. From Table 1, we find out that in 256x256 images, the FCD is about eighty times faster than the edge orientated approach and more than 700 times faster than the CHT. In 512×512 size images, these rates become more than eighty for the EOCHT and more than thousand times faster for the CHT. 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%.

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

Image size 256×256	Original	Binary	Salt&Pepper	Gaussian
CHT	205	172	319	262
EOCHT	23	19.7	29	27
FCD	0.30	0.26	0.32	0.33
Image size 512×512	Original	Binary	Salt&Pepper	Gaussian
CHT	1711	1567	2073	1932
EOCHT	139	113	157	148
FCD	1.58	1.44	1.63	1.60

Another evaluation is done to measure the resistance of our algorithm against noise. In this evaluation, the density of the salt & pepper noise was increased from 0 to 80 percent and the error rate of the FCD algorithm was calculated. Results of this experiment show that the noise with less than 25% density has no effect on the algorithm's accuracy and the algorithm resists against noise until this degree. In contrast the CHT loses robustness against noise from 10%. After this threshold, the error rate increases exponentially for both algorithm but sharper for the CHT, and it failed totally when the noise density exceeds 65% for the FCD and 38% for the CHT. Fig. 4 shows the result of applying the FCD algorithm on a noisy image. The FCD has detected all of the eight coins correctly when 25% salt & pepper noise was applied to image. Fig. 5 draws the error rate versus noise density.

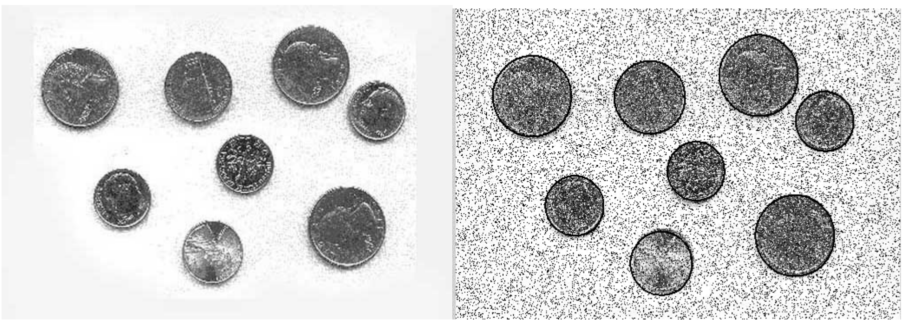

Fig. 4. Output result of the FCD for 25% salt & pepper noise

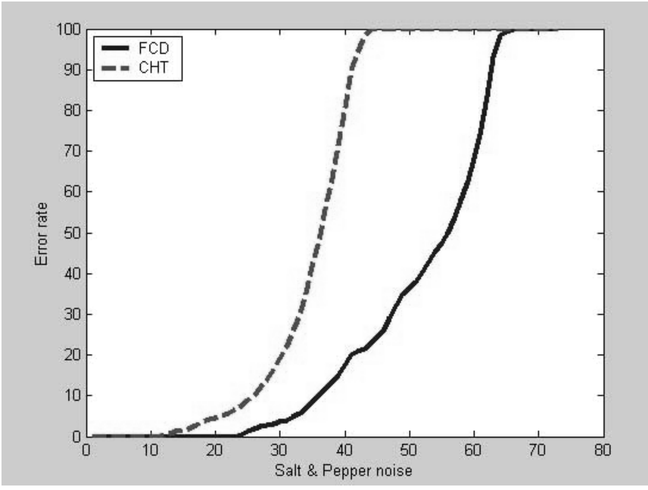

Fig. 5. Resistance of the FCD and the CHT against salt & pepper noise

4 Conclusions and Further Research

In this paper, we presented a size invariant method to find circles that are totally brighter or darker than their backgrounds called Fast Circle Detection (FCD) method. The main idea is based on the symmetry of the gradient pair vectors on such circles. There are various applications that (with some simple preprocessing) the FCD could be applied on. Medical images such as X-Ray image, brain MRI images, and glenohumeral joint CT images, and industrial applications such as eye and pupil images, and microscopy images are some examples of such applications. Fig. 6 shows the result of using of the FCD method in iris localization application.

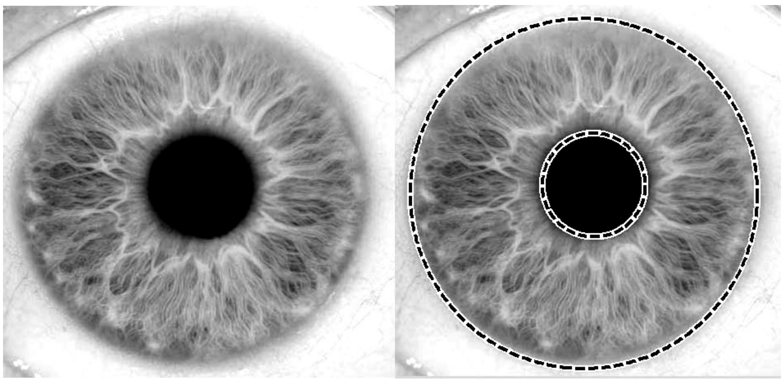

Fig. 4. Iris localization using the FCD method

The experimental results show that the FCD is more than thousand times faster than the CHT and about eighty times faster than the edge oriented CHT in case of 512x512 resolution images. The FCD has good resistance against noise and its accuracy is not affected until 25% salt & pepper noise is applied to the image.

For increasing the accuracy of the FCD, one of our further works is to use certainty factor for each candidate circle based on the parameters of algorithm (α, β, and difference between calculated radius and specified radius). Also 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.

We also plan to use the FCD algorithm to find known ellipses which have known ratio of diameters and directions. Fig. 6 shows a simple result of the FCD to detect a known ellipse in MRI images of brain for fast brain boundary detection.

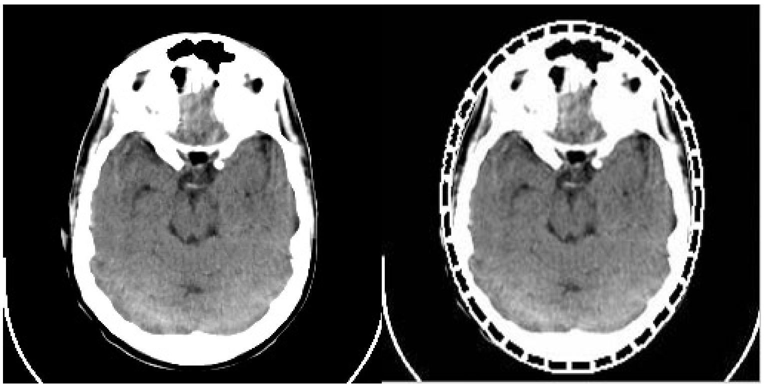

Fig. 5. An example of ellipse detection for fast brain boundary detection in MRI images

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.