Automatic Circle Detection in Digital Images using Artificial Bee Colony Algorithm

Saeid. Fazli, and Saeid. Fathi Ghiri

Abstract**—** This article presents an algorithm for detecting circular shapes from real world complicated and noisy images without using the conventional Hough Transform (HT). The algorithm is based on one of the most recent swarm-intelligence techniques, named Artificial Bee Colony (ABC). Our algorithm is a novel application of the ABC. A new objective function has been derived for the edge map of a desired image. By the use of Artificial Bee Colony the objective function is minimized and this leads to automatic circle detection in digital images.

Keywords**—** Object recognition, Swarm intelligence, Bee Colony, Computer vision.

I. Introduction

AUTOMATIC identification and extraction of circles and ellipses from digital images is one of the most important problems in computer vision industrial applications. Because circular objects occur frequently in many natural and human-made scenes, it has received grate attention for the last few decades [1]. A number of methods have been applied in recent years. For example the Circle Hough Transform (CHT) and some modified versions have been known as robust techniques for circle detection [2], but due to their computational cost researchers have been seeking for some alternative new techniques.

Since heuristic algorithms have seen day light, they have been used for lots of applications because using heuristic algorithms computational cost has been reduced almost dramatically. Ramirez et al. [3] proposed a Genetic Algorithm (GA) based circle detector. Their approach is capable to detect multiply circles in real world images but it fails frequently to detect small and imperfect circles. Dasgupta et al.[4] presented a circle detection algorithm based on Adaptive Bacterial Foraging algorithm (ABFO).

In this work we use Artificial Bee Colony (ABC) [5] for identifying and extracting circles from digital images. We employ the Canny edge detector [6] to generate the edge-map from a gray-scale image. An ABC algorithm is then applied to search the edge-map for circle shape. Each individual bee here represents a trial circle and an objective function has been derived over domain of these trial circles. When the test circle represents a better approximation of the actual edge-circle, the value of the function becomes lesser. At last, minimizing the objective function using the ABC algorithm leads to fast and robust detection of circle in the given digital image.

Saeid. Fazli is with the Research Institute of Physiology and Biotechnology, University of Zanjan, Zanjan, Iran (e-mail: fazli@znu.ac.ir).

Saeid. Fathi Ghiri is with the Electrical Engineering Department, University of Zanjan, Zanjan, Iran (e-mail: Sa.fathi@znu.ac.ir).

II. The ABC:An Overview

The ABC is one of the most recently defined heuristic algorithms. The ABC has introduced by Dervis Karaboga in 2005 [7] and has inspired from the intelligent behavior of honey bees.

In the ABC the colony of bees is divided in three categories: employed bees, onlookers and scouts. A bee waiting on the dance area for gathering information from the others and making decision to choose food source is called onlooker bee. A bee is going to the food source which is visited by itself previously and it's location is saved in her memory, is called employed bee and a bee carrying out a random search is called scout [5,7].

In the ABC algorithm the number of food source are equal to the number of employed bees and it is half of the number of bees (NP). In other words the number of onlooker bees is equal to the number of employed bees. When an employed bee's food source is exhausted by the other bees it becomes a scout.

Each cycle of the ABC algorithm contain three main steps: sending the employed bees to the food source that they have visited and measuring the food sources quality. Choosing food sources by onlookers after employed bees shared their information and measuring their quality. Determining the scouts and sending them to possible food sources [5,7,8,9].

At the initial stage the food sources are randomly selected by the employed bees and their nectar amount (quality) are determined, then they share this information with the bees waiting in the hive. Then at the second stage every employed bee goes back not exactly to the food source which is visited by itself and is saved in its memory, but to a neighbourhood of the present one. This neighbourhood can be determined as below:

$$v_{ij} = x_{ij} + \varphi_{ij}\left(x_{ij} - x_{kj}\right), \qquad (1)$$

Where $k \in \{1, 2, . . . , BN\}$ (BN is the number of employed bees) and $j \in \{1, 2, . . . ,D\}$ (D is the number of dimensions of the objective function) are randomly chosen indexes. Although k is determined randomly, it has to be different from *i*. $\varphi i,j$ is a random number between [−1, 1] and It controls the production of a neighbour food source position around xi,j.

At the third stage the onlookers select food sources depending on their probability which is calculated as below,

$$p_i = \frac{fit_i}{\sum_{n=1}^{SN} fit_n}, \qquad (2)$$

Where fit_i is the fitness value of the solution (food source) and *SN* is the number of food sources (employed bees). As fitness of a food source increase the probability of choosing this food source by onlookers increase too.

In the ABC algorithm we define a limit that if for a certain number of cycles fitness value of a food source which is shown by an employed bee, does not improved the food source is abandoned and the employed bee becomes a scout bee. When a food source is abandoned by the bees, a new food source is randomly determined by a scout bee and replaced with the abandoned one.

In each cycle the best fitness value that has ever achieved and its associated food source location are memorized. This cycle is repeated until a predefined criterion of the objective function is achieved or the predefined maximum cycle number is matched.

III. THE ABC BASED CIRCLE DETECTION ALGORITHM

In this algorithm every individual food source in each step shows a trial circle. Each sample circle is represented as a food source position,

$$food_i = [x_i, y_i, r_i]^T,$$

where x_i *and* y_i are the coordinates of the center of the circle and r_i stands for radius. Let (x_i, y_i, r_i) be the *i*th test circle.

It is obvious that all the points on a complete circle in the edge-map have an intensity value equal to one while the intensity values belong to an interval between [0,1]. So if we have a test circle which it's pixels exactly match with the edge-map, the objective value must be minimum. We show the number of pixels that we need for representing the circle that *i*th food source indicates with N_c. In other words, N_c is the number of dots that we need to draw a circle with radius r_i on white paper while the dots are connected together and there is no dot on the other one.

We define a circle with radius r_i and then moving it to the coordinates (x_i, y_i).

Simply, the objective function can be defined as ,

$$Object\ Value_i = 1 - \frac{N_m}{N_c} \qquad (3)$$

Where N_m is the number of pixels that are exactly matched with the test circle that can be constructed by information which *i*th food source gives us. We see that as the number of matched pixels increase the object value decries, so the object value is equal to zero if all the pixels match with the test circle.

Figure (1) illustrates how pixels match with test circle and how we can calculate objective function. This example shows that the objective value is equal to 0.25 because the circle with radius one (Fig.1.a) is not a complete circle.

In the real world images, we frequently encounter with noisy and imperfect circles. In order to detecting such circles we need to define an interval of radii instead of a single radius. We can define this interval by adding and subtracting a portion of radius to itself. This interval can be $[r_i - {}^{r_i}/_{\alpha}\ , r_i + {}^{r_i}/_{\alpha}]$. In other words, instead of matching edge-map image with a circle, we match it with some more ones. After some try and error we found that α=12, is a good choice.

(1,1)	(1,2)	(1,3)	(1,4)	(1,5)
(2,1)	(2,2)	(2,3)	(2,4)	(2,5)
(3,1)	(3,2)	(3,3)	(3,4)	(3,5)
(4,1)	(4,2)	(4,3)	(4,4)	(4,5)
(5,1)	(5,2)	(5,3)	(5,4)	(5,5)

0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	0	0	0	0
0	0	0	0	0

Fig.1 **a)** a 5×5 image **b)** a's corresponding edge-matrix (the blue point are sample points of a test circle)

We define the fitness function such that at the optimal situation of objective function (minimization) it becomes maximum value. Equation (4) determines the fitness amount of each solution (food source).

$$fit_i = \frac{1}{1 + Object\ Value_i} \qquad (4)$$

IV. EXPREMENTAL RESULTS

We have tried to gather a set of actual images to test our approach. Our test has carried out by 12 artificial (man-made) color and gray-scale images and 12 natural images. Both of the categories include a circular-shape object of different sizes in each image. We have used standard canny edge-detector in image-processing Toolbox, MATLAB 7.0, as a preprocessing step. Our approach is very fast due to the small size of computation which is need to carried out in each cycle. We have reported the results of four images in order to save space.

Table 1 presents the ABC parameters that we have used in this work.

TABLE I

PARAMETER SETUP OF THE ABC ALGORITHM

NP	Limit	Maxcycle	D	BN
40	5	500	3	20

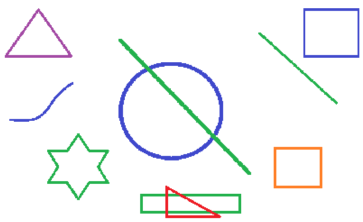
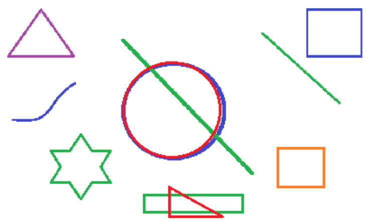
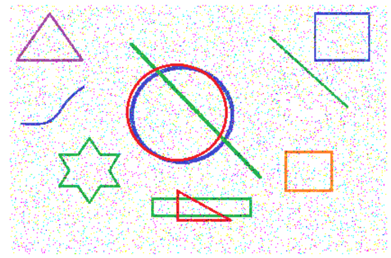
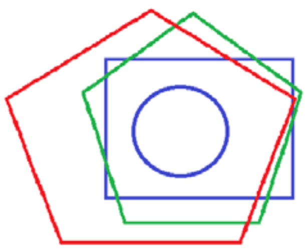
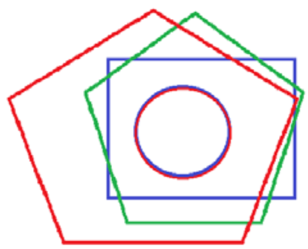
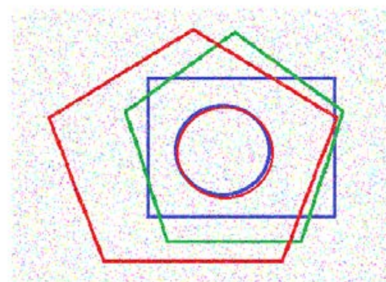
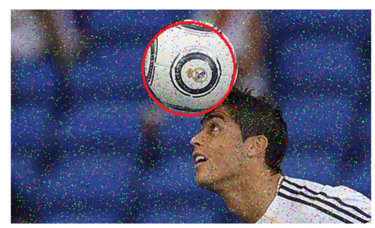
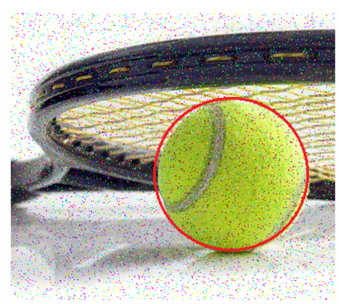
	ORIGINAL IMAGE	ORIGINAL IMAGE WITH THE CIRCLE DETECTED	IMAGE WITH SALT & PEPPER NOISE
I			
II			
III			
IV			

Fig. 2 Simulation results (Detected circles are shown in red.)

We have tested our approach in presence of noise and incomplete circles. The results show that this approach is robust enough to overcome such challenges. We have used salt and pepper noise because it really affects the edge-detection step. Fig.2 illustrates the results of applying our algorithm to four sample images.

The noisy images show our algorithm is robust enough to detect circles on the images which highly affected by a random noise.

V.CONCLUSION

In this paper, we have presented a novel application of the ABC algorithm which is automated circle detection in digital images. To the best of your knowledge it is the first time that the ABC is used for detecting circles in the images. As it is clear in the figure 2 this algorithm is successful in detecting most of circles in images with a small visual error.

REFERENCES

[1] Davies, E. R., “Machine Vision: Theory, Algorithms, Practicalities”, *Academic Press*, London, 1990.

[2] Illingworth J. , Kittler J. ,” A Survey Of The Hough Transform”. Computer vision ,graphics and image processing, 1988.

[3] Ayala-Ramirez, V., Garcia-Capulin, C. H., Perez-Garcia, A., and Sanchez-Yanez, R. E.:” Circle detection on images using genetic algorithms”, *Pattern Recognition Letters*, 27, 652–657, 2006.

[4]Dasgupta S. , Biswas A. , Abraham A. ,”Automatic Circle Detection on Digital Images With an Adaptive Bacterial Foraging Algorithm”, soft computing. Vol. 14, 1151-1164, 2010.

[5]Karaboga D., Basturk B.,” A powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm”, Journal of Global Optimization, Vol 39, Issue: 3, 459-171, 2007.

[6]Canny, J., “A Computational Approach to Edge Detection”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8, 679-714, 1986.

[7]Karaboga D., “An Idea Based On Honey Bee Swarm For Numerical Optimization”, Technical Report-Tr06, Erciyes University, Engineering Faculty, Computer Engineering Department 2005.

[8]Karaboga D., Basturk B., “On The Performance Of Artificial Bee Colony (ABC) Algorithm”, Applied Soft Computing,Volume 8, Issue 1, 687-697, 2008.

[9]Karaboga, D., Akay, B., “A Comparative Study of Artificial Bee Colony Algorithm”Applied Mathematics and Computation, 214, 108-132, 2009.