Soft Comput (2010) 14:1151–1164
DOI 10.1007/s00500-009-0508-z

ORIGINAL PAPER

Automatic circle detection on digital images with an adaptive bacterial foraging algorithm

Sambarta Dasgupta · Swagatam Das · Arijit Biswas · Ajith Abraham

Published online: 15 October 2009

Abstract This article presents an algorithm for the automatic detection of circular shapes from complicated and noisy images without using the conventional Hough transform methods. The proposed algorithm is based on a recently developed swarm intelligence technique, known as the bacterial foraging optimization (BFO). A new objective function has been derived to measure the resemblance of a candidate circle with an actual circle on the edge map of a given image based on the difference of their center locations and radii lengths. Guided by the values of this objective function (smaller means better), a set of encoded candidate circles are evolved using the BFO algorithm so that they can fit to the actual circles on the edge map of the image. The proposed method is able to detect single or multiple circles from a digital image through one shot of optimization. Simulation results over several synthetic as well as natural images with varying range of complexity validate the efficacy of the proposed technique in terms of its final accuracy, speed, and robustness.

Keywords Object recognition · Swarm intelligence · Bacterial foraging · Computer vision · Hand drawn shape location

1 Introduction

Past few decades have seen a massive growth in the field of biologically inspired metaheuristics for search and optimization. Computational cost having been reduced almost dramatically; researchers from all corners are showing more interest in following the underlying principles of nature to solve nearly intractable optimization problems. Following this tradition, practitioners in the field of computer vision are also applying such metaheuristics to difficult problems arising in the field. For example, Wachowiak et al. (2004) applied the particle swarm optimization (PSO) algorithm (Kennedy et al. 2001; Abraham et al. 2006; Liu et al. 2007a, b), which draws inspiration from the intelligent group behavior of social creatures, to the problem of biomedical image registration. Le Hégarat-Mascle et al. (2007) proposed a non-stationary Markov model-based image regularization algorithm, which uses another swarm intelligence algorithm known as ant colony optimization (ACO) (Dorigo and Gambardella 1997). De-Sian and Chien-Chang (2008) applied ACO for improving the edge detection from digital images. Very recently, Santamaría et al. (2009) used memetic algorithms (MAs) (Ong and Keane 2004) that are based on synergy of evolutionary or any population-based approach with separate individual learning or local improvement procedures, in 3D Reconstruction of Forensic Objects.

Automatic identification and extraction of geometric shapes from images are very useful in a number of tasks because such shapes are often present in human-made

S. Dasgupta · S. Das · A. Biswas
Department of Electronics and Telecommunication Engineering,
Jadavpur University, Kolkata, India
e-mail: sambartadg@gmail.com

S. Das
e-mail: swagatamdas19@yahoo.co.in

A. Biswas
e-mail: arijitbiswas87@gmail.com

A. Abraham (✉)
Machine Intelligence Research Labs (MIR Labs),
Scientific Network for Innovation and Research Excellence,
P.O. Box 2259, Auburn, WA 98071-2259, USA
e-mail: ajith.abraham@ieee.org
URL: http://www.mirlabs.org

environments. They are also widely used as a part of man-designed symbols. In particular, circle and ellipse detection problems have been widely studied in the shape recognition community by using the deterministic techniques like Hough transform (HT) (Illingworth and Kittler 1988; Leavers 1993; Duda and Hart 1972), which often necessitates high computation and storage requirements particularly for high-dimensional applications. In order to overcome these limitations, researchers have proposed new approaches to HT, e.g., the probabilistic HT (Shaked et al. 1996), the randomized HT (Xu et al. 1990), and the fuzzy HT (Han et al. 1993). Lam and Yuen (1996) came up with a hybrid approach based on hypothesis filtering and HT to detect circles. New integral transforms have also been proposed in this context, e.g., see Becker et al. (2002). As an alternative to the HT-based techniques, the shape recognition problem in computer vision has been handled with stochastic search methods that include random sample consensus (Fischer and Bolles 1981), simulated annealing (SA) (Bongiovanni and Crescenzi 1995), and genetic algorithm (GA) (Goldberg 1989). Stochastic algorithms like GA exhibit adaptive and robust search behavior can produce near optimal solutions very fast due to possessing a large amount of implicit parallelism. However, they cannot promise to achieve the solution every time. Recently there has been a growing interest of researchers in using GA for important shape detection tasks, e.g., Roth and Levine (1994) proposed use of GA for primitive extraction of images. Lutton et al. carried out a further improvement of the aforementioned method recently in 2000 (Lutton and Martinez 1994). Yao et al. (2004) came up with a multi-population GA to detect ellipses. Ayala-Ramirez et al. (2006) presented a GA-based circle detector. Their approach is capable of detecting multiple circles on real images but fails frequently to detect small and imperfect circles.

Passino (2002) proposed a powerful algorithm for real parameter optimization, now known as the bacterial foraging optimization algorithm (BFOA) (Liu and Passino 2002). BFOA tries to mimic the individual and grouped foraging behavior of *Escherichia coli*, a bacteria living in our intestines. Until date the algorithm has found successful applications in several real world problems like optimal controller design (Liu and Passino 2002), harmonic estimation (Mishra 2005), transmission loss reduction (Tripathy et al. 2006), and power delay filter design (Mishra and Bhende 2007; Das et al. 2009a, b; Kim et al. 2007; Biswas et al. 2007). This work presents an attempt to detect circles (recognizing them by their shapes) from digital images using an adaptive variant of the classical BFOA. The work is not an extension of the HT techniques described earlier and presents an altogether new approach to this classical problem of computer vision. We use the Canny edge detector (Canny 1986) to generate the edge-map from a gray-scale image. The adaptive BFOA is then applied to search the entire edge-map for circular shape. Each bacterium here models a trial circle and an objective function has been derived over the domain of such trial circles in order to measure to what degree a trial circle matches to an actual circle on the edge map of the image. The better a test circle approximates the actual edge-circle, the lesser becomes the value of this function. Minimization of the objective function with BFOA ultimately leads to the fast and robust extraction of circular shapes from the given image. Comparison with one state-of-the-art GA-based method and a randomized Hough transform (RHT) (Xu et al. 1990) algorithm on multiple images indicate the superiority of the proposed method.

The rest of the paper is organized as follows. In Sect. 2, we provide a brief review of classical BFOA. In Sect. 3, we derive the adaptation scheme of the chemotactic step in BFOA leading to its faster convergence with a mathematical justification. Section 4 describes bacteria representation and also lays out the mathematical basis of the objective function. Results of computer simulation over several images have been presented in Sect. 5, and finally the paper is concluded in Sect. 6 with a discussion of future research directions.

2 The classical BFOA: an overview

Now suppose that we want to find the minimum of $J(\theta)$ where $\theta \in \Re^p$ (i.e., θ is a p-dimensional vector of real numbers), and we do not have measurements or an analytical description of the gradient $\nabla J(\theta)$. BFOA mimics the four principal mechanisms observed in a real bacterial system: chemotaxis, swarming, reproduction, and elimination-dispersal to solve this non-gradient optimization problem. Below we introduce the formal notations used in BFOA literature and then provide the complete pseudo-code of the BFO algorithm. A more detailed description of the steps of BFOA is out of the scope of this paper and can be found in Passino (2002), Liu and Passino (2002) and Dasgupta et al. (2008).

Let us define a chemotactic step to be a tumble followed by a tumble or a tumble followed by a run. Let j be the index for the chemotactic step. Let k be the index for the reproduction step. Let l be the index of the elimination-dispersal event. Also let

p	Dimension of the search space,
S	Total number of bacteria in the population,
N_c	The number of chemotactic steps,
N_s	The swimming length
N_{re}	The number of reproduction steps,

N_{ed} The number of elimination-dispersal events,
P_{ed} Elimination-dispersal probability,
$C\ (i)$ The size of the step taken in the random direction specified by the tumble.

Let $P(j,k,l) = \{\theta^i(j,k,l) | i = 1,2,\ldots,S\}$ represent the position of each member in the population of the S bacteria at the jth chemotactic step, kth reproduction step, and lth elimination-dispersal event. Here, let $J(i,j,k,l)$ denote the cost at the location of the ith bacterium $\theta^i(j,k,l) \in \Re^p$ (sometimes, we drop the indices and refer to the ith bacterium position as θ^i). Note that we will interchangeably refer to J as being a "cost" (using terminology from optimization theory) and as being a nutrient surface (in reference to the biological connections). For actual bacterial populations, S can be very large (e.g., $S = 109$), but $p = 3$. In our computer simulations, we will use much smaller population sizes and will keep the population size fixed. BFOA, however, allows $p > 3$ so that we can apply the method to higher dimensional optimization problems. Below, we briefly describe the four prime steps in BFOA. We also provide a pseudo-code of the complete algorithm.

1. *Chemotaxis* This process simulates the movement of an *E. coli* cell through swimming and tumbling via flagella. Suppose $\theta^i(j,k,l)$ represents ith bacterium at jth chemotactic, kth reproductive and lth elimination-dispersal step. $C(i)$ is a scalar and indicates the size of the step taken in the random direction specified by the tumble (run length unit). Then in computational chemotaxis, the movement of the bacterium may be represented by

$$\theta^i(j+1,k,l) = \theta^i(j,k,l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}, \tag{1}$$

   where Δ indicates a unit length vector in the random direction.

2. *Swarming* An interesting group behavior has been observed for several motile species of bacteria including *E. coli* and *S. typhimurium*, where stable spatio-temporal patterns (swarms) are formed in semisolid nutrient medium. A group of *E. coli* cells arrange themselves in a traveling ring by moving up the nutrient gradient when placed amidst a semisolid matrix with a single nutrient chemo-effecter. The cells when stimulated by a high level of *succinate*, release an attractant *aspartate*, which helps them to aggregate into groups and thus move as concentric patterns of swarms with high bacterial density. The cell-to-cell signaling in *E. coli* swarm may be represented by the following function.

$$\begin{aligned} J_{\text{cc}}(\theta, P(j,k,l)) &= \sum_{i=1}^{S} J_{\text{cc}}(\theta, \theta^i(j,k,l)) \\ &= \sum_{i=1}^{S}\left[-d_{\text{attractant}} \exp\left(-w_{\text{attractant}} \sum_{m=1}^{p} (\theta_m - \theta_m^i)^2\right)\right] \\ &\quad + \sum_{i=1}^{S}\left[h_{\text{repellant}} \exp\left(-w_{\text{repellant}} \sum_{m=1}^{p} (\theta_m - \theta_m^i)^2\right)\right], \end{aligned} \tag{2}$$

   where $J_{\text{cc}}(\theta, P(j,k,l))$ is the objective function value to be added to the actual objective function (to be minimized) to present a time varying objective function. The coefficients $d_{\text{aatractant}}, w_{\text{attractant}}, h_{\text{repellant}},$ and $w_{\text{repellant}}$ control the strength of the cell-to-cell signaling. More specifically $d_{\text{aatractant}}$ is the depth of the attractant released by the cell, $w_{\text{attractant}}$ is a measure of the width of the attractant signal (a quantification of the diffusion rate of the chemical), $h_{\text{repellant}} = d_{\text{attractant}}$ is the height of the repellant effect (a bacterium cell also repels a nearby cell in the sense that it consumes nearby nutrients and it is not physically possible to have two cells at the same location), and $w_{\text{repellant}}$ is a measure of the width of the repellant. For a detailed discussion on the function J_{cc} please see, Passino (2002) and Liu and Passino (2002).

3. *Reproduction* The least healthy bacteria eventually die while each of the healthier bacteria (those yielding lower value of the objective function) asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant.
4. *Elimination and dispersal* To simulate this phenomenon in BFOA some bacteria are liquidated at random with a very small probability while the new replacements are randomly initialized over the search space.

3 A simple adaptation scheme for chemotactic step of the BFOA

3.1 Mathematical foundation

Let us consider a single bacterium cell that undergoes chemotactic steps according to (1) over a one-dimensional objective function. The bacterium lives in continuous time and at the tth instant its position is given by $\theta(t)$. Below we list a few assumptions that are considered for the sake of gaining mathematical insight.

1. The objective function $J(\theta)$ is continuous and differentiable at all points in the search space.
2. The chemotactic step size C is not very large.

3. The analysis applies to the regions of the fitness landscape where gradients of the function are small, i.e., near to the optima.

Let, at time t position of an individual bacterium be θ and value of objective function (to be minimized) be $J(\theta)$. Also assume that, after a very small time interval Δt, its position changes by an amount $\Delta\theta$. Then the new value of the objective function becomes $J(\theta+\Delta\theta)$. According to the chemotactic step used in BFOA, the bacterium changes its position only if the modified objective function value is less than the previous one, i.e., $J(\theta) > J(\theta+\Delta\theta)$, i.e., $J(\theta)-J(\theta+\Delta\theta)$ is positive, i.e., $\frac{J(\theta)-J(\theta+\Delta\theta)}{\Delta t}$ is positive. (As time is unidirectional, Δt is a positive infinitesimally small quantity. So, sign of any number is retained after it is divided by Δt.)

The decision-making (i.e., whether to take a step or not) activity of the bacterium can be modeled by a unit step function $u(x)$ as,

$$u\left(\frac{J(\theta)-J(\theta+\Delta\theta)}{\Delta t}\right) = 1, \quad \text{if } \frac{J(\theta)-J(\theta+\Delta\theta)}{\Delta t} > 0,$$
$$= 0, \quad \text{otherwise.}$$

Thus,

$$\frac{\Delta\theta}{\Delta t} = u\left(\frac{J(\theta)-J(\theta+\Delta\theta)}{\Delta t}\right)C\Delta, \tag{3}$$

where C indicates the chemotactic step height and Δ indicates the direction of tumble (here it can assume only two values 1 or -1 with uniform probabilities). Note that modeling the decision-making activity of a bacterium with unit step function reinforce the local search behavior since a step is taken only if it keeps on minimizing (i.e., improving the fitness of a bacterium) following the directives of original BFOA. In that case, the advancement is done with an intensity C in Δ direction. Here, we have assumed in unit time bacterium takes a chemotactic step. $\frac{\Delta\theta}{\Delta t}$ is the velocity.

$$\Rightarrow \quad \frac{\Delta\theta}{\Delta t} = u\left[-\frac{J(\theta+\Delta\theta)-J(\theta)}{\Delta t}\right]C\Delta.$$

Now velocity of the bacterium is given by,

$$V_b = \underset{\Delta t\to 0}{Lt}\frac{\Delta\theta}{\Delta t}$$
$$= u\left[-\left(\underset{\Delta\theta\to 0}{Lt}\frac{J(\theta+\Delta\theta)-J(\theta)}{\Delta\theta}\right)\underset{\Delta t\to 0}{Lt}\frac{\Delta\theta}{\Delta t}\right]C\Delta, \tag{4}$$

$$\Rightarrow \quad V_b = \theta' = u\left[-\frac{dJ(\theta)}{d\theta}\frac{d\theta}{dt}\right]C\Delta,$$

$$\Rightarrow \quad V_b = u(-GV_b)C\Delta, \tag{5}$$

where $G = J'(\theta)$ = gradient of the objective function at θ. Since the unit step function $u(x)$ has a jump discontinuity at $x = 0$, to simplify the analysis further, we replace $u(x)$ with the continuous logistic function $\phi(x)$ (Anwal 1998), where

$$\phi(x) = \frac{1}{1+e^{-kx}}. \tag{6}$$

We note that, $u(x) = Lt_{k\to\infty}\,\phi(x) = Lt_{k\to\infty}\,\frac{1}{1+e^{-kx}}$

Figure 1 illustrates how the logistic function may be used to approximate the unit step function used for decision-making in chemotaxis. For analysis purpose, k cannot be infinity. We restrict ourselves to moderately large values of k (say $k = 10$) for which $\phi(x)$ fairly approximates $u(x)$. The following subsection describes the error limit introduced by this assumption. Thus, for moderately high values of k $\phi(x)$ fairly approximates $u(x)$. Hence from (5),

$$V_b = \frac{C\Delta}{1+e^{kGV_b}}. \tag{7}$$

According to assumptions (1) and (3), if C and G are very small and $k \sim 10$, then we may have $|kGV_b| \ll 1$. In that case, we neglect higher order terms in the expansion of e^{kgv_b} and have $e^{kgv_b} \approx 1 + kGV_b$. Substituting it in (7), we obtain,

$$V_b = \frac{C\Delta}{2+kGV_b},$$

$$\Rightarrow \quad V_b = \frac{C\Delta}{2}\frac{1}{1+\frac{kGV_b}{2}},$$

$$\Rightarrow \quad V_b = \frac{C\Delta}{2}\left(1-\frac{kGV_b}{2}\right) \quad \left[\because \left|\frac{kGV_b}{2}\right| \ll 1 \text{ Neglecting higher order terms and } \left(1+\frac{kGV_b}{2}\right)^{-1} \approx \left(1-\frac{kGV_b}{2}\right)\right].$$

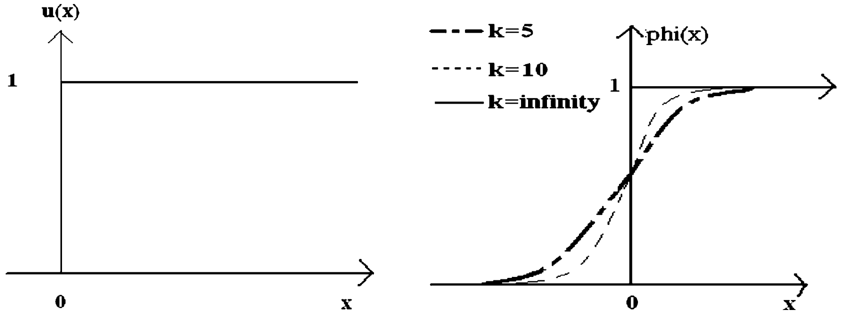


Fig. 1 The unit step and the logistic functions

After some manipulation we have,

$$V_{\mathrm{b}} = \frac{2C\Delta}{4 + kGC\Delta} \tag{8}$$

$$\Rightarrow \quad V_{\mathrm{b}} = \frac{C\Delta}{2}\frac{1}{1+\frac{kCG\Delta}{4}},$$

$$\Rightarrow \quad V_{\mathrm{b}} = \frac{C\Delta}{2}\left(1 - \frac{kGC\Delta}{4}\right) \left[\because \left|\tfrac{kGC\Delta}{4}\right| \ll 1,\ \text{as } |\Delta| = 1 \text{ and neglecting the higher order terms}\right],$$

$$\Rightarrow \quad V_{\mathrm{b}} = -\frac{kC^2}{8}G + \frac{C\Delta}{2}, \tag{9}$$

$$\Rightarrow \quad \frac{\mathrm{d}\theta}{\mathrm{d}t} = -\alpha G + \beta, \tag{10}$$

where α is $\frac{-kC^2}{8}$ and β is $\frac{C\Delta}{2}$.

The classical gradient descent algorithm (CGDA) (Fletcher 1987) is given by the following dynamics in single dimension:

$$\frac{\mathrm{d}\theta}{\mathrm{d}t} = -\alpha G, \tag{11}$$

where α is the learning rate.

Similarity between Eqs. 9, 10, and 11 suggests that chemotaxis is a modified CGDA where α, which is a function of chemotactic step size, can be identified as the learning rate of chemotaxis.

3.2 Oscillation problem: need for adaptive chemotaxis

If magnitude of the gradient decreases consistently, near the optima or very close to the optima α of Eq. 15 becomes comparable to β. Then gradually β becomes dominant. When $|G| \to 0, \left|\frac{\mathrm{d}\theta}{\mathrm{d}t}\right| \approx |\beta| = \left|\frac{C\Delta}{2}\right| = \frac{C}{2}$ $[\because |\Delta| = 1]$.Let us assume the bacterium has reached close to the optimum. But as we have derived $\left|\frac{\mathrm{d}\theta}{\mathrm{d}t}\right| = \frac{C}{2}$, the bacterium does not stop taking chemotactic steps. It oscillates about the optima. This crisis can be remedied if step size C is made adaptive according to the following relation,

$$C = \frac{\psi|J(\theta)|}{|J(\theta)| + \lambda} = \frac{\psi}{1 + {}^{\lambda}\!/_{|J(\theta)|}}. \tag{12}$$

Here, λ and ψ (>1) are positive constants. Thus from (12), we see, if $J(\theta) \to 0$, then $C \to 0$. Therefore, there would be no oscillation if the bacterium reaches optima. The functional form of (12) causes C to vanish near the optima. Besides, it plays another important role described below. From (12), we have, when $J(\theta)$ is large $\frac{\lambda}{|J(\theta)|} \to 0$ and consequently $C \to \psi(>1)$. This has an important physical significance. If magnitude of cost function is large for an individual bacterium, it is in vicinity of the noxious substance. It will then try to move to a place with better nutrient concentration by taking large steps. On the other hand, the bacterium when in nutrient rich zone, i.e., with small magnitude of the objective function value, tries to retain its position. Naturally, its step size becomes small. Dasgupta et al. (2008) have illustrated the effectiveness of the above-mentioned adaptation scheme over several numerical benchmarks.

4 The ABFOA-based circle detection algorithm

4.1 The scheme for population initialization

An individual member of the population of sample circles is actually a trial solution. Each sample circle is represented as a bacterial position $\vec{\theta} = [x_0, y_0, r_0]^{\mathrm{T}}$, where first two components of the vector are x and y co-ordinates of the center of that circle and third term stands for the radius.

Let (x_k, y_k, r_k) be the kth test circle in the population, where $k = 1, \ldots, S$, S is the population size, i.e., it denotes total number of test circles. In initialization phase, random values within suitable range are assigned to each of the three entries of the vectors. Two statistical properties of sample points on a candidate circle are used in this regard, where one property is used to find a suitable value for co-ordinates of probable center of the test circle and, another is used to model its probable radius. These two properties are described below. Let us consider $2N$ number of equally spaced sample points on a circle of radius r_0 and center at (x_0, y_0). Figure 2 shows a similar case where $N = 4$. Now, ith sample point may be designated by $(x(i), y(i))$.

Here, $x(i) = x_0 + r_0 \cos\frac{\pi}{N}i$ and $y(i) = y_0 + r_0 \sin\frac{2\pi}{N}i$, $i = 1, \ldots, 2N$.

Let $(\overline{X}, \overline{Y})$ be defined as,

$$\overline{X} = \frac{1}{2N}\sum_{i=1}^{2N} x(i), \quad \overline{Y} = \frac{1}{2N}\sum_{i=1}^{2N} y(i). \tag{13}$$

Now, $\overline{X} = \frac{1}{2N}\sum_{i=1}^{2N} x(i) = \frac{1}{2N}\sum_{i=1}^{2N}\left(x_0 + r_0\cos\frac{\pi}{N}i\right) = x_0 + r_0\frac{1}{2N}\sum_{i=1}^{2N}\cos\frac{\pi}{N}i$

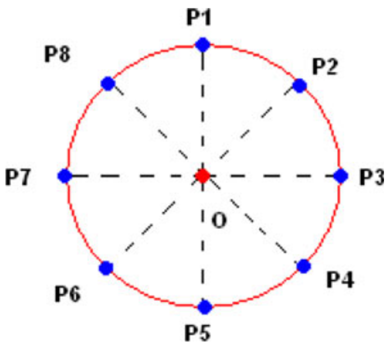


Fig. 2 An edge-circle with sample points depicted in blue

$$\therefore \overline{X} = x_0 + r_0 \frac{\sin\pi}{\sin\frac{\pi}{2N}} \cos\frac{\pi(2n+1)}{4n} \Rightarrow \overline{X} = x_0$$

$[\because \sin\pi = 0]$.

Similarly, we can show that $\overline{Y} = y_0$.

$$\therefore \overline{X} = x_0, \quad \overline{Y} = y_0. \tag{14}$$

Let us also define

$$\sigma = \sqrt{\frac{1}{2N}\left(\sum_{i=1}^{2N}(x(i)-\overline{X})^2 + (y(i)-\overline{Y})^2\right)}, \tag{15}$$

$$\Rightarrow \quad \sigma^2 = \frac{1}{2N}\left(\sum_{i=1}^{2N}(x(i)-x_0)^2 + (y(i)-y_0)^2\right),$$

$[\because \overline{X} = x_0, \overline{Y} = y_0]$,

$$\Rightarrow \quad \sigma^2 = \frac{1}{2N}\left(\sum_{i=1}^{2N}\left(r_0\cos\frac{\pi}{N}i\right)^2 + \left(r_0\sin\frac{\pi}{N}i\right)^2\right)$$

$$\Rightarrow \sigma^2 = r_0^2\frac{1}{2N}\left(\sum_{i=1}^{2N}\left(\cos\frac{\pi}{N}i\right)^2 + \left(\sin\frac{\pi}{N}i\right)^2\right),$$

$$\Rightarrow \quad \sigma^2 = r_0^2,$$

$$\Rightarrow \quad \sigma^2 = r_0^2. \tag{16}$$

To initialize kth test-circle we form a set of $2m_k$ edge-points drawn at random from the edge-map, where $2m_k = \mathrm{rand}(L, U)$ such that L, U specify upper and lower limit of $2m_k$. Let $(x(i), y(i))$ be the ith edge-point, $i = 1, \ldots, 2m_k$ and (x_k, y_k, r_k) denotes the kth test circle to be initialized. Now, we assign values to three entries of the vector taking guidelines of Eqs. 13 and 15.We assign

$$x_k = \frac{1}{2m_k}\sum_{i=1}^{2m_k} x(i), \quad y_k = \frac{1}{2m_k}\sum_{i=1}^{2m_k} y(i)$$

$$\text{and} \quad r_k = \sqrt{\frac{1}{2m_k}\left(\sum_{i=1}^{2m_k}(x(i)-x_k)^2 + (y(i)-y_k)^2\right)}.$$

This initialization scheme is adopted with the expectation that the selected edge points may lie on the actual circle or very close to it, so that we may reach a good approximation of the center and radius.

4.2 The objective function

We are assuming that in the edge-map edge pixels are simply represented by '1' and back-ground pixels by '0'. Figure 3 illustrates a similar situation. Let us consider a test circle be denoted by (x_0, y_0, r_0). And let A be the edge matrix. We denote the objective function be J. Let for the above-mentioned test circle, its value be J_0. $\therefore J_0 = J(A, x_0, y_0, r_0)$..

We define another function $P(x, y)$, which returns pixel value of a point (x, y).

$$\begin{aligned} P(x,y) &= 1, \quad \text{if } (x,y) \text{ is an edge - point} \\ &= 0, \quad \text{otherwise.} \end{aligned}$$

So the edge-matrix $A = (a_{ij}) = (P(i,j))$. As an example, for image in Fig. 3a, $P(2,3) = P(3,2) = P(3,4) = P(4,3) = 1$ (for the edge pixels marked in red) and for other non-edge pixels (white colored in Fig. 3a), e.g., $P(1,1) = P(2,2) = P(3,3) = P(3,1) = P(5,5) = 0$.

For sampling, we consider the family of circles with center at (x_0, y_0) with a radius varying in a range from $r_0 - \delta$ to $r_0 + \delta$. This forms the test-band, δ being a constant. After some trial and error, we found that taking $\delta = \frac{r_0}{6}$ is comfortable enough for this specific application. After this, N_S sample points are taken for each circle in test-band. Sample points are positioned on the circle maintaining equal distance between each other. Let ith sample point on a circle of test-band, having radius $r_0 + j\ (-\delta \le j \le \delta)$ be denoted by (x_i^j, y_i^j).

$$\therefore x_i^j = (r_0 + j)\cos\frac{2\pi}{N_S}i, \quad y_i^j = (r_0 + j)\sin\frac{2\pi}{N_S}i, \tag{17}$$

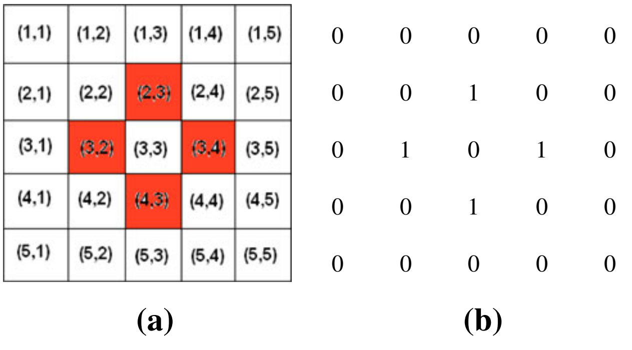
$i = 1, 2, \ldots N_S$ and

$j = -\delta, -\delta + 1, \ldots, -1, 0, 1, \ldots, \delta - 1, \delta.$

In this work, to deal with real world images containing incomplete and noisy circles, we define a distance function μ to measure the degree of belongingness of a sample point to the circumference of the central circle of the test band (i.e., circle with radius r_0). Value of this distance function is μ_i^j for sample point (x_i^j, y_i^j) and it is defined as:

$$\mu_i^j = P(x_i^j, y_i^j)\exp\left(-\frac{1}{2\sigma^2}(r + j - r_0)^2\right). \tag{18}$$

If (x_i^j, y_i^j) is not an edge-point, we can infer $\mu_i^j = 0$, i.e., a zero belongingness to the circumference of the central circle $[\because P(x_i^j, y_i^j) = 0]$. Let us consider now the case that (x_i^j, y_i^j) is an edge-point. Then $\mu_i^j = \exp\left(-\frac{1}{2\sigma^2}(r + j - r_0)^2\right)$ and note



(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)

(a)

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

(b)

Fig. 3 **a** An 5 × 5 image matrix and **b** its corresponding edge-matrix

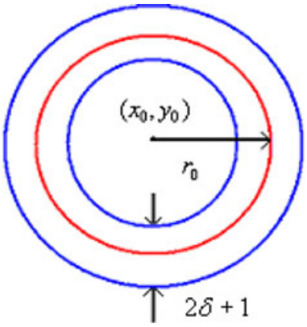


Fig. 4 Test-band for a test circle. *Red circle* shows the central circle. Two *blue circles* show the boundaries of test-band

that $\mu_i^j = 1$ if $j = 0$ and $r = r_0$, i.e., the distance function is unity or maximum when sampled edge-point lies on central circle. It decreases when $|j|$ increases and, it follows the distribution shown in Fig. 4. If σ decreases the function becomes more sharp cut-off or narrow-band. Now, normalized objective function corresponding to (x_0, y_0, r_0) for the edge matrix A is defined as following,

$$J(A, x_0, y_0, r_0) = 1 - \frac{1}{(2\delta + 1)N_S} \sum_{i=1}^{N_S} \sum_{j=-\delta}^{\delta} \mu_i^j$$

$$\Rightarrow \quad J(A, x_0, y_0, r_0) = 1 - \frac{1}{(2\delta + 1)N_S} \sum_{i=1}^{N_S} \sum_{j=-\delta}^{\delta} P(x_i^j, y_i^j)$$

$$\exp\left(-\frac{1}{2\sigma^2}(r + j - r_0)^2\right). \tag{19}$$

Note that this function is a heuristic that weighs each edge-point according to the distance of the candidate circle. Here, we would also like to point out that most of the existing GA-based circle detection methods (e.g., Ayala-Ramirez et al. 2006) encode one circle in one chromosome with the coordinates of three points on that circle. They usually evaluate fitness of the encoded circle by taking several sample points on the test circle and checking their status (whether edge pixel, etc.). Now two circles can have maximum two points of intersection. Due to digital approximation of the circle, the number may be greater than two by small amount but still not large enough to draw a comprehensive inference to guide the search process. To circumvent this limitation in this work, we take a band of circles symmetrically distributed around actual test circle. Sample points are taken on the periphery of each circle and then we test whether they are edge-point or not. Therefore, the probability of detecting the circle increases even when the edge pixels are not completely connected. To differentiate between points on actual test circle and that of lying on circle on test band, a distance function is assigned to each sample point. The function is basically a Gaussian deviate and it reaches the maxima when point is on the central circle of the band and decreases on both sides.

4.3 Multiple circle detection

The proposed algorithm with slight modification can detect multiple circular objects on real image. At first, an upper limit of the number of circular shapes is specified. The method is run on the original edge-map until convergence is achieved for first circle detection. This shape is then masked on the edge-map and the ABFOA circle-detector is then again run on the edge-map. The procedure is repeated until the maximum number of circles is attained. In this approach, there is a possibility of false detection as the actual number of circles present may be lesser than the maximum number specified. To overcome this, once the process is terminated, all the detected circles are validated by analyzing their connectivity of circumference segments using the method of Kelly and Levine (1997).

5 Computer simulation results

5.1 Test images

The test bed includes 15 synthetic (handcrafted) gray-scale images and 15 natural test images each of size 256 × 256 pixels. The natural images include a circular-shaped object among various other configurations. All the images were preprocessed using a standard edge-detector (Canny or Sobel edge-detector in image-processing toolbox, MATLAB 7.0). We have reported detailed results for six images in order to save space.

5.2 Parametric setup

Table 1 presents the parameters of ABFOA algorithm used here. Once set (after a considerable number of trials and error) values of the parameters were kept unaltered for all the test images without allowing any type of hand tuning.

Detailed meaning of all these parameters has been provided in Sect. 2. In order to make an objective evaluation of the proposed method, circular shapes in three of the natural images were detected manually and the circles detected by our algorithm were then compared with the

Table 1 Parameter setup for ABFOA

S	N_c	N_s	N_{ed}	P_{ed}	$d_{attract}$	$w_{attract}$	$w_{repellant}$	$h_{repellant}$	λ	ψ
50	100	4	1	0.25	0.1	0.2	10	0.1	400	6

hand-label ones (considered as the ground truth images) using a new heuristic score.

For the GA algorithm described in Ayala-Ramirez et al. (2006), we take the population size = 70, crossover probability = 0.55, mutation probability = 0.10, number of elite individuals = 2. We also took as selection method, the roulette wheel selection and as a crossover method, the 1-point crossover. The parametric setup appears as the best set, configured by Ayala-Ramirez et al. (2006) after lot of hand tuning experiments. The same fitness function as the one described by Ayala-Ramirez et al. (2006) has been used for the GA implementation.

5.3 Error score and success rate

To test accuracy of the algorithm obtained result is compared with ground truth. First, the circle in the edge-map is detected manually and it is referred as the ground-truth image. Next three peripheral points are taken on the manually detected circle. We know that through three non-collinear points only one circle can pass. So from previous three points, we get integer-approximated (as we are dealing with digital circle) center and radius of the ground-truth circle (GTC). Let center of the GTC be $(x_{\mathrm{GTC}}, y_{\mathrm{GTC}})$ and its radius be r_{GTC}. Let the chosen points on the GTC be $(x_1, y_1), (x_2, y_2)$, and (x_3, y_3).Then

$$x_{\mathrm{GTC}} = \frac{\begin{vmatrix} x_2^2 + y_2^2 - (x_1^2 + y_1^2) & 2(y_2 - y_1) \\ x_3^2 + y_3^2 - (x_1^2 + y_1^2) & 2(y_3 - y_1) \end{vmatrix}}{4(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)}$$

$$\text{and} \quad y_{\mathrm{GTC}} = \frac{\begin{vmatrix} 2(x_2 - x_1) & x_2^2 + y_2^2 - (x_1^2 + y_1^2) \\ 2(x_3 - x_1) & x_3^2 + y_3^2 - (x_1^2 + y_1^2) \end{vmatrix}}{4(x_2 - x_1)(y_3 - y_1) - (x_3 - x_1)(y_2 - y_1)}.$$

If center and radius of the detected circle found by algorithm is denoted by $(x_{\mathrm{D}}, y_{\mathrm{D}})$ and r_{D}, then we can define an error score in the following way,

$$\mathrm{ES} = \eta(|x_{\mathrm{GTC}} - x_{\mathrm{D}}| + |(y_{\mathrm{GTC}} - y_{\mathrm{D}}|) + \mu|r_{\mathrm{GTC}} - r_{\mathrm{D}}| \qquad (20)$$

First term represents shift of center of the detected circle from that of GTC and second term signifies difference of their radii. η and μ are two weights associated with each term in (20). They may be chosen according to the accuracy required. We have taken $\eta = 0.05$ and $\mu = 0.1$. This particular choice of parameters ensures that radii difference gets priority to the difference of the centers of the manually detected circle and the machine-detected circle. Here we assume that if *ES* is found to be less than 1, then the algorithm gets a success, otherwise, we say that it has failed to detect the edge-circle. Note that for $\eta = 0.05$ and $\mu = 0.1$, ES < 1 means the maximum difference of radius tolerated is 10 while the maximum mismatch in the location of the center can be 20 (in number of pixels). From this consideration, success rate (SR) is defined as percentage of reaching success in certain number of trials.

5.4 Simulation strategy

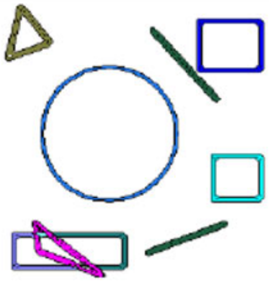
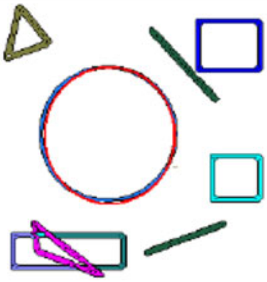
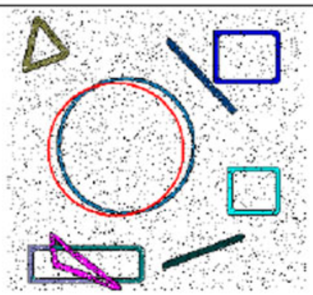
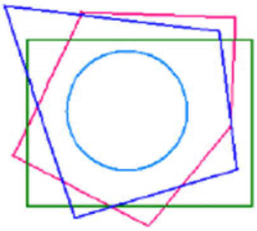
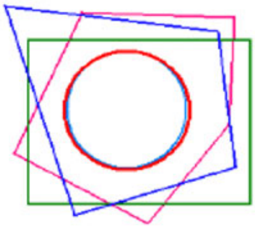
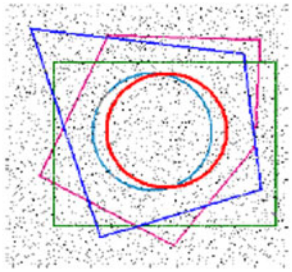
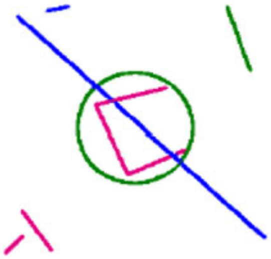
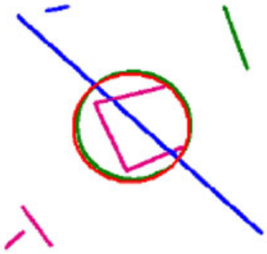
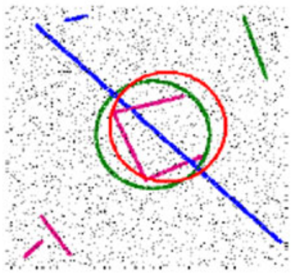
The algorithm has been developed from scratch on a Pentium IV 2.4 GHz PC using C language under windows XP environment. Fifty independent runs of the proposed algorithm were carried out with different seeds of random number generator over each test image.

5.5 Presentation of results

In Table 2, we report the final heuristic score (between ground truth and the actual image with circle detection), as well as the time taken for each image. In Table 2, we have provided three synthetic images and their counterparts processed with the proposed algorithm. Table 3 presents the same experimental results on natural images. In order to test the robustness of our algorithm, we have added salt and pepper noise to the synthetic images before the algorithm was applied. It also illustrates the performance of the algorithm in presence of such noisy and corrupted pixels. In all cases, we have reported the results for median-run solution (when the runs were ranked according to their final fitness values). In order to test the effectiveness of the adaptive chemotaxis scheme, we compared the performance of ABFOA with that of the classical BFOA over the same test-bed. In all the test cases, ABFOA yielded more accurate final results consuming lesser amount of computational time. Figure 5 indicates the superior performance of ABFOA in terms of convergence speed over natural image 1 and synthetic image III (the noisy version). We omit rest of the graphical results of the comparison as they show a similar trend.

Table 4 provides a visual illustration of the performances of classical BFOA, ABFOA and the GA algorithm described in (Ayala-Ramirez et al. 2006) on two hard problem instances (one natural and one synthetic image). The results are for the median of 50 independent runs of each algorithm. It is interesting to observe that the deviation between the detected circle and actual circle is least for ABFOA. Table 5 shows the average time taken, success rate (in %), and average error score (calculated following Eq. 20) for the three competitor algorithms over six test images considered in Tables 2 and 3. The best entries are marked in bold in this Table. A close inspection of Table 5 reveals that the ABFOA based method was able to achieve highest success rate and minimum error taking least computational time in majority of the cases.

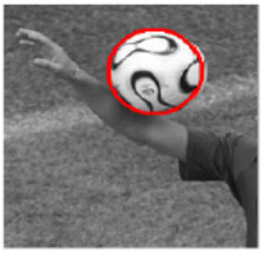
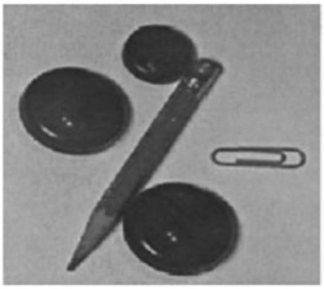
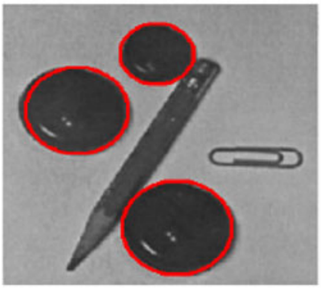
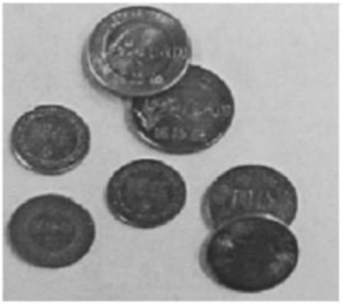
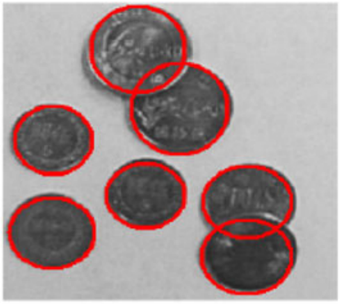
Table 2 Synthetic images and detected circles

Image Index	Original image	Image with detected circle	Image with 'salt and pepper' noise
I	 (a)	 (b)	 (c)
II	 (a)	 (b)	 (c)
III	 (a)	 (b)	 (c)

A non-parametric statistical significance test called Wilcoxon's rank sum test for independent samples (Wilcoxon 1945; García et al. 2008) has been conducted at the 5% significance level on the error score (ES) data of Table 5. Table 6 shows reports the *P*-values produced by Wilcoxon's rank sum test for comparison of the error scores of two groups (one group corresponding to ABFOA and the other corresponding to a competitor algorithm) at a time. As a null hypothesis, it is assumed that there is no significant difference between the mean values of two groups. Whereas, the alternative hypothesis is that there is significant difference in the mean values of the two groups. All the *P* values reported in the Table are less than 0.05 (5% significance level). This is strong evidence against the null hypothesis, indicating that the better mean values of the performance metrics produced by ABFO is statistically significant and has not occurred by chance.

Table 7 demonstrates relative performance of ABFOA in comparison to the RHT algorithm described in Ong and Keane (2004). Images taken for the test are complicated and also comprises of different geometrical figures. RHT finds out the circle located in the figures satisfactorily but ABFOA is found to determine the position of circles with higher accuracy. Table 8 reports the corresponding average time taken, success rate (in %), and average error score (calculated following Eq. 20) for ABFOA and RHT algorithms over three test images considered in Table 7.

Table 3 Natural images and detected circles

Image Index	Original image	Image with detected circle
I	 (a)	 (b)
II	 (a)	 (b)
III	 (a)	 (b)

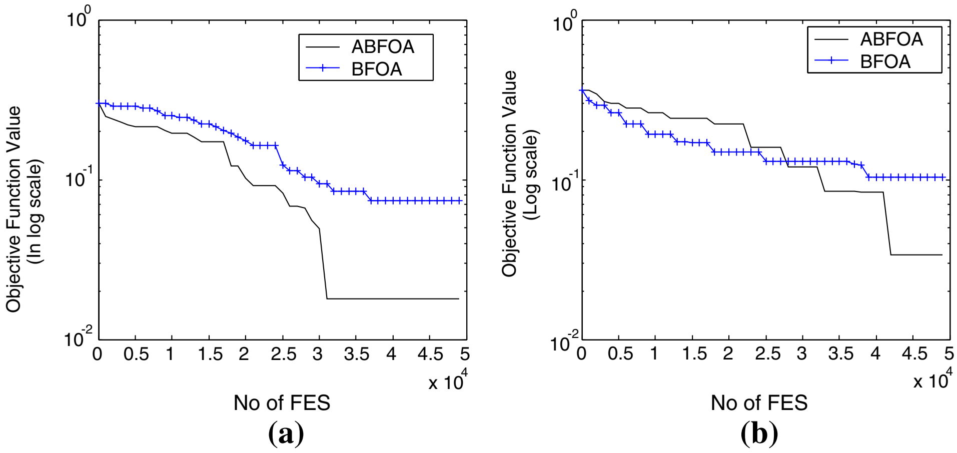


Fig. 5 Convergence characteristics: **a** for natural image I, **b** for synthetic image III (noisy version)

Table 4 Performances of ABFOA, Classical BFOA, and GA based method (Ayala-Ramirez et al. 2006) on a natural image (row 1) and a synthetic image (row 2) corrupted with salt and pepper noise

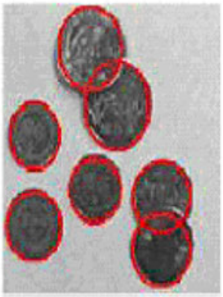
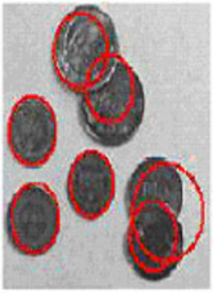
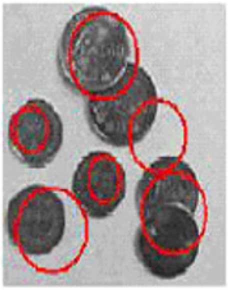
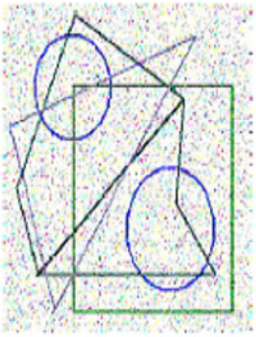
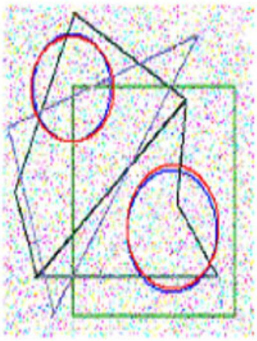
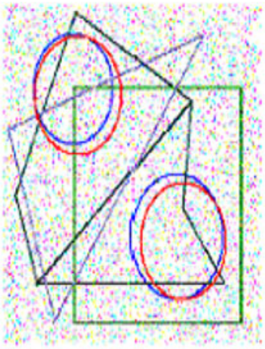
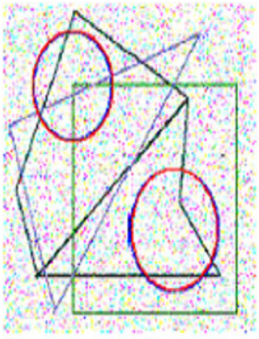
Original Image	Circle detected by ABFOA	Circle detected by BFOA	Circle detected by GA
			
			

Table 5 Average time taken and success rate of ABFOA, classical BFOA and the GA based method (Lam and Yuen 1996) over the six test images

Index	Average time ± Standard deviation (s)			Success rate (SR) (%)			Average error score (ES) ± Standard deviation		
	BFOA	ABFOA	GA based method (Ayala-Ramirez et al. 2006)	BFOA	ABFOA	GA based method (Ayala-Ramirez et al. 2006)	BFOA	ABFOA	GA based method (Ayala-Ramirez et al. 2006)
Synthetic image									
I	1.39 ± (0.24)	**0.94 ± (0.18)**	1.21 ± (0.16)	93	**100**	**100**	0.68 ± 0.093	**0.35 ± 0.057**	0.56 ± 0.073
II	1.86 ± (0.58)	**1.04 ± (0.26)**	1.13 ± (0.55)	**100**	**100**	93	0.54 ± 0.036	**0.22 ± 0.043**	0.48 ± 0.079
III	1.65 ± (0.21)	1.34 ± (0.54)	**1.30 ± (0.03)**	95	**97**	92	0.72 ± 0.032	**0.31 ± 0.074**	0.68 ± 0.042
Natural images									
I	1.76 ± (0.34)	**1.46 ± (0.86)**	2.19 ± (0.34)	96	**100**	**100**	0.71 ± 0.038	**0.46 ± 0.083**	0.57 ± 0.072
II	2.34 ± (0.56)	**1.67 ± (0.39)**	1.91 ± (0.30)	94	**98**	94	0.81 ± 0.096	**0.58 ± 0.075**	0.69 ± 0.042
III	2.65 ± (0.58)	**1.98 ± (0.59)**	2.44 ± (0.68)	94	**95**	92	1.49 ± 0.143	**0.34 ± 0.059**	0.93 ± 0.136

6 Discussion and conclusion

This paper has presented a novel application of the BFOA to the task of automatic circle detection from gray images. To the best of our knowledge, BFOA has not been applied to any such computer-vision related task till date. An adaptive version of the basic BFOA has been used in order to improve its convergence behavior. In addition, a new fitness function was derived specifically for the circle detection task. As can be perceived from the result provided in Tables 3 and 5, ABFOA detects most of the circles in an image with little visual distortion even in the presence of noisy background pixels. Unlike most of the conventional papers on circle detection and instead of relying on visual perception only we have formulated a score function [presented in (20)], which can objectively evaluate the mismatch between a manually detected circle and a machine-detected circle. From Table 4, we observe that for natural images with more complexity the heuristic score is relatively high and the average time elapsed is

Table 6 *P* values produced by Wilcoxon's rank sum test taking comparing ABFOA with other algorithms on the Error Score (ES) data of Table 5

Image	*P* value	
	Classical BFOA	GA-based method (Ayala-Ramirez et al. 2006)
Synthetic I	1.8165e−004	1.8267e−004
Synthetic II	1.8172e−004	1.8264e−004
Synthetic III	1.7036e−004	1.9252e−004
Natural I	1.8231e−004	1.8115e−004
Natural II	1.8138e−004	1.8235e−004
Natural III	1.8368e−004	1.8094e−004

Table 7 Relative performance of ABFOA and HT

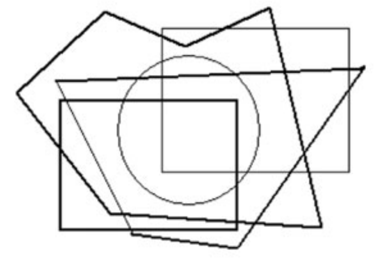
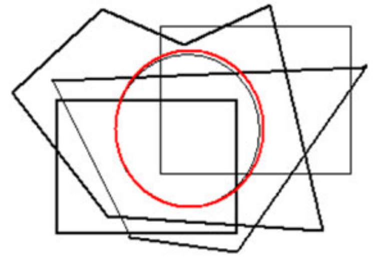
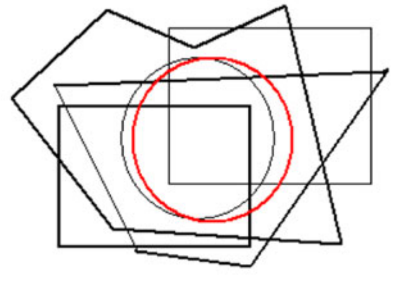
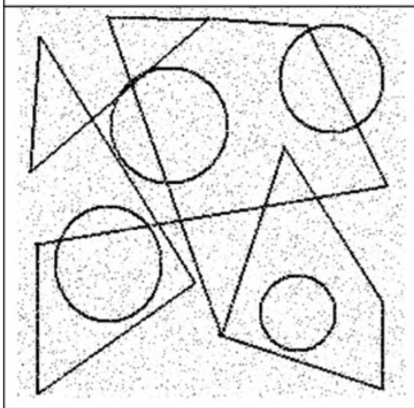
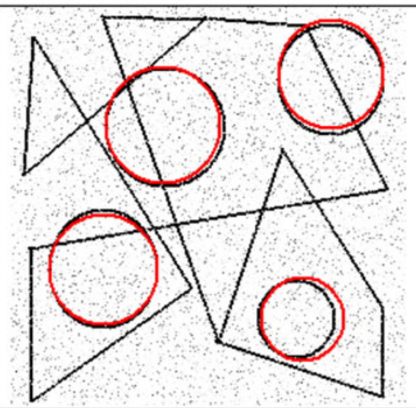
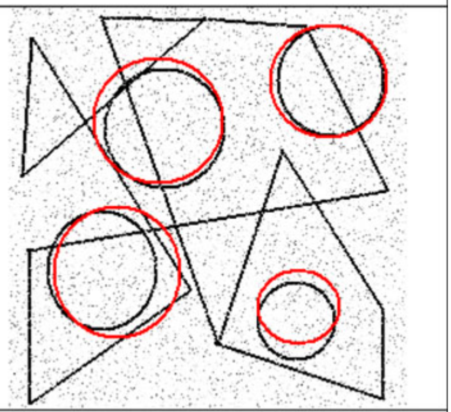
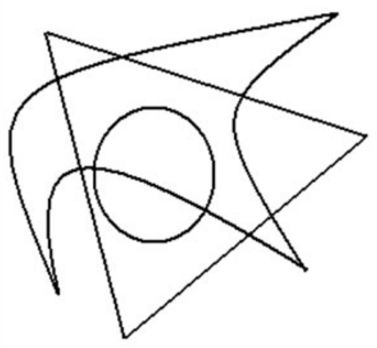
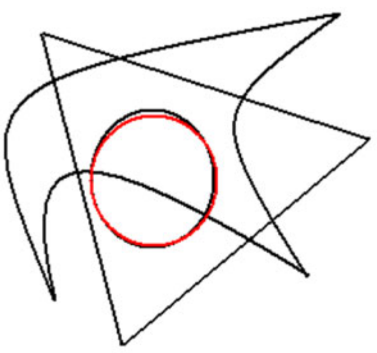
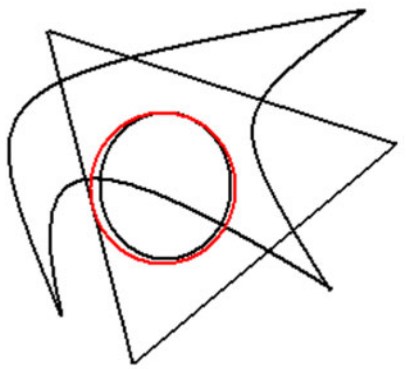
Original Image	Circle Detected Image using ABFOA	Circle Detected Image using RHT [12]
		
		
		

Table 8 Average time taken and success rate of ABFOA and HT over the 3 test images

Index	Average time ± Standard deviation (s)		Success rate (SR) (%)		Average Error Score (ES) ± Standard deviation	
	ABFOA	HT based method (Xu et al. 1990)	ABFOA	HT based method (Xu et al. 1990)	ABFOA	HT based method (Xu et al. 1990)
I	**1.24 ± (0.36)**	1.81 ± (0.28)	**94**	91	**0.68 ± 0.088**	0.74 ± 0.096
II	**2.82 ± (0.45)**	3.01 ± (0.55)	**90**	86	**0.92 ± 0.087**	1.12 ± 0.132
III	2.65 ± (0.46)	**1.60 ± (0..53)**	**94**	90	**0.56 ± 0.069**	0.47 ± 0.078

higher as well. Through the use of non-parametric Wilcoxon's rank sum test, we demonstrated that the ABFOA-based method could outperform both GA (as described in Ayala-Ramirez et al. 2006) and classical BFOA in a statistically significant fashion. Since both classical BFOA and ABFOA use the same initial population, bacteria representation scheme and fitness function, it is evident that the superior performance of ABFOA is attributed to its adaptation scheme of the chemotactic step-size *C*.

Although Table 7 indicated that ABFOA can yield better results on complicated and noisy images as compared to the RHT, note that the objective of our paper is not to devise a circle-detection algorithm that could beat all the HT based methods discovered earlier, but to show that a new swarm intelligence algorithm called BFOA with the suggested modifications can serve as an attractive alternative of the GA that has been used earlier to extract circular shapes from objects successfully. This way, the work undertaken here, encourages further research on the application of BFOA to the tasks related to shape extraction in computer vision. Future research will focus on hybridizing ABFOA with different HT based techniques for automatic shape extraction. In addition, application of the ABFOA based techniques for automatic shape recognition by mobile robots may also be studied. In future, we shall also attempt to compare the performance of our ABFOA based algorithm with other evolutionary computation techniques on the circle detection problem quantitatively.

References

Abraham A, Guo H, Liu H (2006) Swarm intelligence: foundations, perspectives and applications, swarm intelligent systems. In: Nedjah N, Mourelle L (eds) Studies in computational intelligence. Springer, Germany, pp 3–25

Anwal RP (1998) Generalized functions: theory and technique, 2nd edn. Birkhãuser, Boston

Ayala-Ramirez V, Garcia-Capulin CH, Perez-Garcia A, Sanchez-Yanez RE (2006) Circle detection on images using genetic algorithms. Pattern Recogn Lett 27:652–657

Becker J, Grousson S, Coltuc D (2002) From Hough transforms to integral transforms. In: Proceedings of international geoscience and remote sensing symposium, IGARSS_02 3:1444–1446

Biswas A, Dasgupta S, Das S, Abraham A (2007) A synergy of differential evolution and bacterial foraging algorithm for global optimization. Neural Netw World 17(6):607–626

Bongiovanni G, Crescenzi P (1995) Parallel simulated annealing for shape detection. Comput Vision Image Understanding 61(1):60–69

Canny J (1986) A computational approach to edge detection. IEEE Trans Pattern Anal Mach Intell 8:679–714

Das S, Biswas A, Dasgupta S, Abraham A (2009a) Bacterial foraging optimization algorithm: theoretical foundations, analysis, and applications, foundations of computational intelligence. Global optimization, studies in computational intelligence, vol 3. Springer, Germany, pp 23–55. ISBN: 978-3-642-01084-2

Das S, Dasgupta S, Biswas A, Abraham A, Konar A (2009b) On stability of the chemotactic dynamics in bacterial foraging optimization algorithm. IEEE Trans Syst Man Cybern Part A 39(3):670–679

Dasgupta S, Das S, Abraham A, Biswas A (2009) Adaptive computational chemotaxis in bacterial foraging optimization: an analysis. IEEE Trans Evol Comput 13(4):919–941

De-Sian L, Chien-Chang C (2008) Edge detection improvement by ant colony optimization. Pattern Recogn Lett 29(4):416–425

Dorigo M, Gambardella LM (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1(1):53–66

Duda RO, Hart PE (1972) Use of the Hough transformation to detect lines and curves in pictures. Comm Assoc Comput Mach 15:11–15

Fischer M, Bolles R (1981) Random sample consensus: a paradigm to model fitting with applications to image analysis and automated cartography. CACM 24(6):381–395

Fletcher R (1987) Practical methods of optimization, 2nd edn. Wiley, Chichester

García S, Molina D, Lozano M, Herrera F (2008) A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special session on real parameter optimization. J Heurist. doi:10.1007/s10732-008-9080-4

Goldberg DE (1989) Genetic algorithms in search. Optimization and machine learning. Kluwer, Boston

Han JH, Koczy LT, Poston T (1993) Fuzzy Hough transform. In: Proceedings 2nd international conference on fuzzy systems 2:803–808

Illingworth J, Kittler J (1988) Survey: a survey of the Hough transform. Comput Vision Graphics Image Process 44:87–116

Kelly M, Levine M (1997) Advances in image understanding. Finding and describing objects in complex images. IEEE Computer Society Press, pp 209–225

Kennedy J, Eberhart R, Shi Y (2001) Swarm intelligence. Morgan Kaufmann Academic Press, Menlo Park

Kim DH, Abraham A, Cho JH (2007) Hybrid genetic algorithm and bacterial foraging approach for global optimization. Inf Sci 177(18):3918–3937

Lam W, Yuen S (1996) Efficient techniques for circle detection using hypothesis filtering and Hough transform. IEEE Proc Visual Image Signal Process 143(5):292–300

Le Hégarat-Mascle S, Hégarat-Mascle L, Kallel A, Descombes X (2007) Ant colony optimization for image regularization based on a nonstationary markov modeling. IEEE Trans Image Process 16(3):865–878

Leavers VF (1993) Survey: which Hough transforms. CVGIP Image Understanding 58:250–264

Liu Y, Passino KM (2002) Biomimicry of social foraging bacteria for distributed optimization: models, principles, and emergent behaviors. J Optim Theory Appl 115(3):603–628

Liu H, Abraham A, Clerc M (2007a) Chaotic dynamic characteristics in swarm intelligence. Appl Soft Comput J 7(3):1019–1026

Liu H, Abraham A, Zhang W (2007b) A fuzzy adaptive turbulent particle swarm optimization. Int J Innov Comput Appl 1(1):39–47

Lutton E, Martinez P (1994) A genetic algorithm for the detection 2-D geometric primitives on images. In: Proceedings of the 12th international conference on pattern recognition (ICPR_94), vol 1, Jerusalem, Israel, pp 526–528

Mishra S (2005) A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation. IEEE Trans Evol Comput 9(1):61–73

Mishra S, Bhende CN (2007) Bacterial Foraging Technique-Based Optimized Active Power Filter for Load Compensation. IEEE Trans Power Deliv 22(1):457–465

Ong YS, Keane AJ (2004) Meta-Lamarckian learning in memetic algorithms. IEEE Trans Evol Comput 8:99–110

Passino KM (2002) Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems Magazine, pp 52–67

Roth G, Levine MD (1994) Geometric primitive extraction using a genetic algorithm. IEEE Trans Pattern Anal Mach Intell 16(9):901–905

Santamaría J, Cordón O, Damas S, García-Torres JM, Quirin A (2009) Performance evaluation of memetic approaches in 3D reconstruction of forensic objects. Soft Comput 13(8–9):883–904

Shaked D, Yaron O, Kiryati N (1996) Deriving stopping rules for the probabilistic Hough transform by sequential analysis. Computer Vision Image Understanding 63:512–526

Tripathy M, Mishra S, Lai LL, Zhang QP (2006) Transmission loss reduction based on FACTS and bacteria foraging algorithm. PPSN, pp 222–231

Wachowiak MP, Smolikova R, Yufeng Z, Zurada JM, Elmaghraby AS (2004) An approach to multimodal biomedical image registration utilizing particle swarm optimization. IEEE Trans Evol Comput 8(3):289–301

Wilcoxon F (1945) Individual comparisons by ranking methods. Biometrics 1:80–83

Xu L, Oja E, Kultanen P (1990) A new curve detection method: Randomized Hough Transform (RHT). Pattern Recogn Lett 11(5):331–338

Yao J, Kharma N, Grogono P (2004) Fast robust GA-based ellipse detection. In: Proceedings of 17th international conference on pattern recognition ICPR-04, vol 2, Cambridge, UK, pp 859–862