Information Sciences 182 (2012) 40–55

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Circle detection using electro-magnetism optimization

Erik Cuevas [a], Diego Oliva [a], Daniel Zaldivar [a,*], Marco Pérez-Cisneros [a], Humberto Sossa [b]

[a] *Departamento de Ciencias Computacionales, Universidad de Guadalajara, CUCEI, Av. Revolución 1500, Guadalajara, Jal, Mexico*
[b] *Centro de Investigación en Computación, IPN, Av. Juan de Dios Bátiz s/n, Mexico, DF, Mexico*

ARTICLE INFO

Article history:
Available online 9 January 2011

Keywords:
Circle detection
Electro-magnetism optimization
Nature-inspired algorithms
Collective intelligence
Shape detection
Intelligent image processing

ABSTRACT

Nature-inspired computing has yielded remarkable applications of collective intelligence which considers simple elements for solving complex tasks by common interaction. On the other hand, automatic circle detection in digital images has been considered an important and complex task for the computer vision community that has devoted a tremendous amount of research, seeking for an optimal circle detector. This paper presents an algorithm for the automatic detection of circular shapes embedded into cluttered and noisy images without considering conventional Hough transform techniques. The approach is based on a nature-inspired technique known as the Electro-magnetism Optimization (EMO). It follows the electro-magnetism principle regarding a collective attraction–repulsion mechanism which manages particles towards an optimal solution. Each particle represents a solution by holding a charge which is related to the objective function to be optimized. The algorithm uses the encoding of three non-collinear points embedded into an edge-only image as candidate circles. Guided by the values of the objective function, the set of encoded candidate circles (charged particles) are evolved using an EMO algorithm so that they can fit into actual circular shapes over the edge-only map of the image. Experimental evidence from several tests on synthetic and natural images which provide a varying range of complexity validates the efficiency of our approach regarding accuracy, speed and robustness.

1. Introduction

Nature-inspired computing [23] studies the application of biology concepts to solve demanding problems by assuming that natural world experiences may hold some answers to real-life technical challenges [16]. Engineering is being challenged everyday by more complex, large, ill-structured and distributed systems yielding a renovated interest on the subject. However, nature is providing simple structures and organizations which are capable of dealing with most complex systems and tasks. Many nature-inspired approaches consider phenomena that exhibit some type of collective intelligence [14,26,39,24], a concept that has lately been developed and applied to problems holding a complex behavioral pattern [13,41]. The approach follows the idea that a system is composed of decentralized individuals that may effectively interact with other elements according to their localized knowledge. Therefore, the overall image of the system emerges from aggregating individual interactions. Special kinds of artificial collective-individuals are the elements created by analogy with bees [20], ants [7] or charged particles [30]. Nature-inspired algorithms with collective intelligence characteristics have recently

* Corresponding author. Tel.: +52 33 1378 5900x7715.
E-mail addresses: erik.cuevas@cucei.udg.mx (E. Cuevas), diego.oliva@cucei.udg.mx (D. Oliva), daniel.zaldivar@cucei.udg.mx (D. Zaldivar), marco.perez@cucei.udg.mx (M. Pérez-Cisneros), hsossa@cic.ipn.mx (H. Sossa).

0020-0255/$ - see front matter © 2011 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2010.12.024

gained considerable interest from the computer vision community as they effectively add interesting features from other subjects such as optimization, pattern recognition, shape detection and machine learning.

The problem of detecting circular shapes holds paramount importance for image analysis, in particular for industrial applications such as automatic inspection of manufactured products and components, aided vectorization of drawings, target detection, etc. [5,38]. Two sorts of techniques are commonly applied to solve the object location challenge: first hand deterministic techniques including the application of Hough transform based methods [50], geometric hashing and template or model matching techniques [1,32]. On the other hand, stochastic techniques including random sample consensus techniques [12], simulated annealing [8] and Genetic Algorithms (GA) [37], have been also used.

Template and model matching techniques are the first approaches to be applied to shape detection yielding a considerable amount of publications [22,27,48]. Shape coding techniques and combination of shape properties have been commonly used to represent objects. Their main drawback is related to the contour extraction step from a real image which hardly deals with pose invariance, except for very simple objects.

Circle detection in digital images is commonly performed by the Circular Hough Transform [29]. A typical Hough-based approach employs edge information obtained by means of an edge detector to infer locations and radius values. Peak detection is then performed by averaging, filtering and histo-gramming the transformed space. However, such an approach requires a large storage space given the required 3-D cells to cover all parameters $(x,y,\ r)$ and a high computational complexity which yields a low processing speed. The accuracy of the extracted parameters for the detected circle is poor, particularly under the presence of noise [3]. For a digital image holding a significant width and height and a densely populated edge pixel map, the required processing time for Circular Hough Transform makes it prohibitive to be deployed in real time applications. In order to overcome this problem, some other researchers have proposed new approaches based on the Hough transform, for instance the probabilistic Hough transform [40], the randomized Hough transform (RHT) [46] and the fuzzy Hough transform [15]. Alternative transformations have also been presented in literature as the one proposed by Becker et al. [6]. Although those new approaches demonstrated faster processing speeds in comparison to the original Hough Transform, they are still highly sensitive to noise.

Stochastic search methods such as Genetic Algorithms (GA) are other alternatives to shape recognition in computer vision. In particular, GA has recently been used for important shape detection tasks. In [37], Roth and Levine have proposed the use of GA for primitive extraction of images, while in [25], Lutton et al. have added a further improvement of the aforementioned method. Also in [47], Yao et al. have proposed a multi-population GA to detect ellipses. In [51], GA has been used for template matching when the pattern has been the subject of an unknown affine transformation. Ayala-Ramirez et al. presented a GA based circle detector [4] which is capable of detecting multiple circles on real images albeit failing on detecting imperfect circular shapes. Another example is presented in [36] which discusses how soft computing techniques can be used for shape classification. In the case of ellipsoidal detection, in [35] Rosin proposes an ellipse fitting algorithm that uses five points, while in [54] Zhang and Rosin extends the later algorithm to fit data into super-ellipses. Most of such approaches perform circle detection under an acceptable computation time despite noisy conditions. However, they still fail at the time of facing complex conditions such as occlusion or superposition.

In this paper, the circle detection task is considered as an optimization problem. Although several traditional optimization algorithms have been proposed, they require substantial gradient information targeting the solution within a neighborhood of a given initial approximation. If the problem has more than one local solution, the convergence to the global solution may depend on the provided initial approximation [2].

On the other hand, nature-inspired methods with collective behavior have been successfully applied for solving constrained global optimization problems [13,41,53]. The electro-magnetism optimization (EMO) algorithm has been recently proposed. It is a flexible and effective nature-based approach for single objective optimization problems. EMO originates from the electro-magnetism theory of physics by assuming potential solutions as electrically charged particles which spread around the solution space. The charge of each particle depends on its objective function value. This algorithm employs a collective attraction–repulsion mechanism to move the particles towards optimality. Such relationship among particles corresponds to the interaction among individuals produced by reproduction, crossover and mutation in Genetic Algorithms (GA) [10]. Likewise, EMO moves particles around the search space following similar paths from other methods such as the Particle Swarm Optimization (PSO) [11,49] and the Ant Colony Optimization (ACO) [42].

Although EMO algorithm shares some characteristics to other nature-inspired approaches, recent works (see [33,34,43,45]) have exhibited EMO's great savings on computation time and memory allocation as to surpass other methods such as GA, PSO and ACO. EMO's low computational cost and its reassured convergence [18] have both been successfully applied to the solution of different sorts of engineering problems such as flow-shop scheduling [31], vehicle routing [52], array pattern optimization in circuits [19], neural network training [45] and control systems [21]. However, to the best of our knowledge, EMO has not yet been applied to any computer-vision related task.

This paper presents an EMO algorithm based circle detector method which assumes the overall detection process as an optimization problem. The algorithm uses the encoding of three non-collinear edge points lying on the edge-only image of the scene as candidate circles (x,y,r). Guided by the values of the objective function, the set of encoded candidate circles are evolved using an EMO algorithm so that they can fit into the actual circles on the edge map. The approach generates a subpixel circle detector which can effectively identify circles despite circular objects exhibiting significant occluded portions. Experimental results show performance evidence for detecting circles under different conditions.

This paper is organized as follows: Section 2 provides a brief outline of the EMO theory. Section 3 explains the application of the EMO algorithm for circle detection. Section 4 discusses the results after designing an EMO application for a number of images exhibiting different conditions. Finally, Section 5 offers some insights into the algorithm's performance and some conclusions.

2. Electro-magnetism Optimization Algorithm (EMO)

EMO algorithm is a simple and direct search algorithm which has been inspired by the electro-magnetism phenomenon. It is based on a given population and the optimization of global multi-modal functions. In comparison to GA, it does not use crossover or mutation operators to explore feasible regions; instead it does implement a collective attraction–repulsion mechanism yielding a reduced computational cost with respect to memory allocation and execution time. Moreover, no gradient information is required and it employs a decimal system which clearly contrasts to GA. Few particles are required to reach converge as has been already verified in [18].

EMO algorithm can effectively solve a special class of optimization problems with bounded variables in the form of:

$$\begin{aligned} &\min f(x), \\ &x \in [l,u], \end{aligned} \tag{1}$$

where $[l,u] = \{x \in \Re^n | l_d \leqslant x_d \leqslant u_d, d = 1,2\ldots,n\}$ and n being the dimension of the variable x, $[l,u] \subset \Re^n$, a nonempty subset and a real-value function $f : [l,u] \to \Re$. Hence, the following problem's features are known:

- n: Dimensional size of the problem.
- u_d: The highest bound of the kth dimension.
- l_d: The lowest bound of the kth dimension.
- $f(x)$: The function to be minimized.

EMO algorithm has four phases [17]: initialization, local search, computation of the total force vector and movement. A deeper discussion on each stage follows:

Initialization: a number of m particles is gathered as their highest (u) and lowest limit (l) are identified.
Local search: gathers local information for a given point $\mathbf{g}^p$, where $p \in (1,\ldots,m)$.
Calculation of the total force vector: charges and forces are calculated for every particle.
Movement: each particle is displaced accordingly, matching the corresponding force vector.

2.1. Initialization

First, the group **G** of m solutions is randomly produced at an initial state. Each n-dimensional solution is regarded as a charged particle holding a uniform distribution between the highest (u) and the lowest (l) limits. The optimum particle (solution) is thus defined by the objective function to be optimized. The procedure ends when all the m samples are evaluated, choosing the sample (particle) that has gathered the best function value.

```
$Counter \leftarrow 1$
$Length \leftarrow \delta(\max\{u_d - l_d\})$
for $p=1$ to $m$ do
    for $d=1$ to $n$ do
        $\lambda_1 \leftarrow U(0,1)$
        while $Counter < ITER$ do
            $\mathbf{t} \leftarrow \mathbf{g}^p$
            $\lambda_2 \leftarrow U(0,1)$
            if $\lambda_1 > 0.5$ then
                $\mathbf{t}_d \leftarrow \mathbf{t}_d + \lambda_2(Length)$
            else
                $\mathbf{t}_d \leftarrow \mathbf{t}_d - \lambda_2(Length)$
            end if
            if $f(\mathbf{t}) < f(\mathbf{g}^p)$ then
                $\mathbf{g}^p \leftarrow \mathbf{t}$
                counter ← ITER - 1
            end if
            $Counter \leftarrow Counter + 1$
        end while
    end for
end for
$\mathbf{g}^{best} \leftarrow \arg\min\{f(\mathbf{g}^p), \forall p\}$
```

Fig. 1. Pseudo-code list for the local search algorithm.

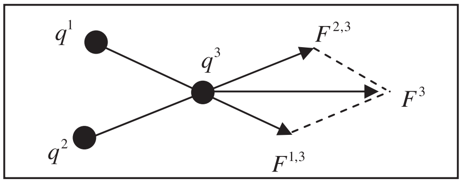

Fig. 2. The superposition principle.

2.2. Local search

Theoretically, the local search should be able to find a better solution when applicable. However it may be unnecessary for some problems [19] as a broad classification of the algorithms can be formulated as follows: EMO with no local search, EMO with local search applied to all particles and EMO with local search applied only to the current best particle.

Considering a determined number of iterations, known as *ITER*, and a feasible neighborhood search δ, the procedure iterates as follows: point $\mathbf{g}^p$ is assigned to a temporary point $\mathbf{t}$ to store the initial information. Next, for a given coordinate d, a random number is selected (λ_1) and combined with δ as a step length, which in turns moves the point $\mathbf{t}$ along the direction d, with a randomly determined sign (λ_2). If point $\mathbf{t}$ observes a better performance within the iteration number ITER, point $\mathbf{g}^p$ is replaced by $\mathbf{t}$ and the neighborhood search for point $\mathbf{g}^p$ finishes, otherwise $\mathbf{g}^p$ is held. Finally the *current best* point is updated. The pseudo-code is shown in Fig. 1.

In general, the local search for all particles can reduce the risk of falling onto a local solution but is relatively time consuming. Keeping the local search focused only on the current best particle is a rather convenient procedure in order to maintain the computational efficiency and precision. The step length for local search is therefore an important factor that depends on the limits for each dimension and determines the performance of the overall local search.

2.3. Total force vector computation

The total force vector computation is based on the *superposition principle* (Fig. 2) from the electro-magnetism theory which states: "the force exerted on a point via other points is inversely proportional to the distance between the points and directly proportional to the product of their charges" [28]. The particle moves following the resultant Coulomb's force which has been produced among particles as a charge-like value. In the EMO implementation, the charge for each particle is determined by its fitness value as follows:

$$q^p = \exp\left(-n\frac{f(\mathbf{g}^p) - f(\mathbf{g}^{best})}{\sum_{h=1}^{m}(f(\mathbf{g}^h) - f(\mathbf{g}^{best}))}\right), \quad \forall p, \tag{2}$$

where n denotes the dimension of $\mathbf{g}^p$ and m represents the population size. A higher dimensional problem usually requires a larger population. In Eq. (2), the particle showing the best fitness function value $\mathbf{g}^{best}$ is called the "*best particle*", getting the highest charge and attracting other particles holding high fitness values. The repulsion effect is applied to all other particles exhibiting lower fitness values. Both effects, attraction–repulsion are accordingly applied depending on the actual proximity between a given particle and the best-graded element.

The overall resultant force between all particles determines the actual effect of the optimization process. The final force vector for each particle is evaluated under the Coulomb's law and the superposition principle as follows:

$$\mathbf{F}^p = \sum_{h \neq p}^{m} \left\{ \begin{array}{ll} (\mathbf{g}^h - \mathbf{g}^p)\frac{q^p q^h}{\|\mathbf{g}^h - \mathbf{g}^p\|^2} & \text{if } f(\mathbf{g}^h) < f(\mathbf{g}^p) \\ (\mathbf{g}^p - \mathbf{g}^h)\frac{q^p q^h}{\|\mathbf{g}^h - \mathbf{g}^p\|^2} & \text{if } f(\mathbf{g}^h) \geqslant f(\mathbf{g}^p) \end{array} \right\}, \quad \forall p \tag{3}$$

where $f(\mathbf{g}^h) < f(\mathbf{g}^p)$ represents the attraction effect and $f(\mathbf{g}^h) \geqslant f(\mathbf{g}^p)$ represents repulsion force (see Fig. 3). The resultant force of each particle is proportional to the product between charges and is inversely proportional to the distance between particles. In order to keep feasibility, the vector in expression Eq. (3) should be normalized as follows:

$$\hat{\mathbf{F}}^p = \frac{\mathbf{F}^p}{\|\mathbf{F}^p\|}, \quad \forall p. \tag{4}$$

2.4. Movement

The change of the d-coordinate for each particle p is computed with respect to the resultant force as follows:

$$g_d^p = \left\{ \begin{array}{ll} g_d^p + \lambda \cdot \hat{F}_d^p \cdot (u_d - g_d^p) & \text{if } \hat{F}_d^p > 0 \\ g_d^p + \lambda \cdot \hat{F}_d^p \cdot (g_d^p - l_d) & \text{if } \hat{F}_d^p \leqslant 0 \end{array} \right\}, \quad \forall p \neq best, \ \forall d. \tag{5}$$

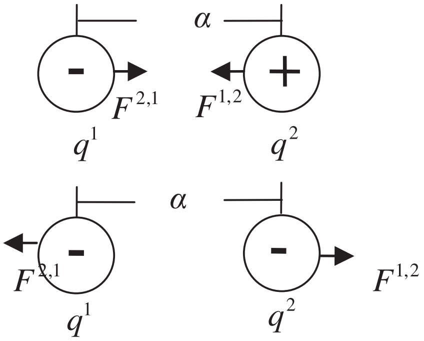

Fig. 3. Coulomb law: α represents the distance between charged particles, q^1, q^2 are the charges, and F is the exerted force as has been generated by the charge interaction.

In Eq. (5), λ is a random step length that is uniformly distributed between zero and one. u_d and l_d represent the upper and lower boundary for the d-coordinate, respectively. $\hat{F}_d^p$ represents the d element of $\hat{\mathbf{F}}^p$. The particle moves towards the highest boundary by a random step length if the resultant force is positive. Otherwise, it moves toward the lowest boundary. The best particle does not move at all, because it holds the absolute attraction pulling or repelling all others in the population.

The process is halted when a maximum iteration number is reached or when the value $f(\mathbf{g}^{best})$ is near zero or the required optimal value.

3. Circle detection using EMO

Circles are represented by the parameters of the well-known second degree equation shown by Eq. (6), considering only three points [3] in the edge-only space of the image. A pre-processing stage requires marking the object's contour by applying a single-pixel edge detection method. For our purpose, such task is accomplished by the classical Canny's algorithm, storing the locations for each edge point. All points are the only potential candidates to define circles by considering triplets. They are stored within a vector array $\mathbf{E} = \{\mathbf{e}^1, \mathbf{e}^2, \ldots, \mathbf{e}^{Np}\}$ with Np being the total number of edge pixels in the image. In turn, the algorithm stores the (x_v, y_v) coordinates of each edge pixel yielding the edge vector $\mathbf{e}^v(e_1^v = x_v, e_2^v = y_v)$.

In order to construct each candidate circle (or charged particles within the EMO framework), the indexes v_1, v_2 and v_3 of three non-collinear edge points must be combined, assuming that the circle's circumference includes the points $\mathbf{e}_{v_1}; \mathbf{e}_{v_2}; \mathbf{e}_{v_3}$. A number of candidate solutions are generated randomly for the initial pool of particles. The solutions will thus evolve through the recursive application of the EMO algorithm as the evolution affects all particles until a minimum is reached and the best particle is considered as a solution. The objective function improves at each generation step by discriminating non-plausible circles, locating others and avoiding useless visits to other image points. The discussion below explains the required steps to formulate the circle detection task from EMO's perspective.

3.1. Particle representation

Each particle (circle candidate) $\mathbf{C}$ is defined by the combination of three edge points. Under such representation, edge points are stored by assigning an index which refers to their relative position within the edge array $\mathbf{E}$. In turn, the procedure will encode each particle as the circle passing through three points: $\mathbf{e}_i$, $\mathbf{e}_j$ and $\mathbf{e}_k$ ($\mathbf{C}^p = \{\mathbf{e}_i, \mathbf{e}_j, \mathbf{e}_k\}$). Each circle $\mathbf{C}$ is thus represented by three parameters: x_0, y_0 and r, with (x_0, y_0) representing the circle's centre coordinates and r its radius. The circle equation can thus be computed as follows:

$$(x - x_0)^2 + (y - y_0)^2 = r^2, \tag{6}$$

considering

$$\mathbf{A} = \begin{bmatrix} x_j^2 + y_j^2 - (x_j^2 + y_i^2) & 2\cdot(y_j - y_i) \\ x_k^2 + y_k^2 - (x_i^2 + y_i^2) & 2\cdot(y_k - y_i) \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} 2\cdot(x_j - x_i) & x_j^2 + y_j^2 - (x_i^2 + y_i^2) \\ 2\cdot(x_k - x_i) & x_k^2 + y_k^2 - (x_i^2 + y_i^2) \end{bmatrix}, \tag{7}$$

$$x_0 = \frac{\det(\mathbf{A})}{4((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i))}, \quad y_0 = \frac{\det(\mathbf{B})}{4((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i))}, \tag{8}$$

and

$$r = \sqrt{(x_0 - x_b)^2 + (y_0 - y_b)^2}, \tag{9}$$

where det(.) stands for the determinant and $b \in \{i,j,k\}$. Fig. 4 illustrates the parameters defined by Eqs. (7)–(9).

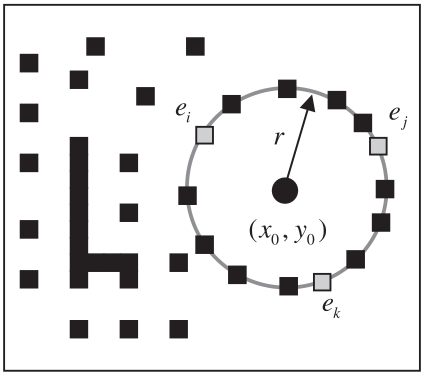

Fig. 4. Circle candidate (charged-particle) following the combination of points p_i, p_j and p_k.

Therefore, it is possible to represent a set of shape parameters for each circle $[x_0, y_0, r]$ as a transformation T for all edge vector indexes i, j and k which yields

$$[x_0, y_0, r] = T(i, j, k), \tag{10}$$

with T being the transformation calculated after the previous computations for x_0, y_0, and r.

EMO algorithm explores the space by assuming each circle as a particle, searching effective circular shape parameters and eliminating unfeasible solutions.

3.2. Objective function

A circumference may be calculated as a virtual shape as a way to measure the matching factor between the candidate circle **C** and the actual circle within the image, i.e. it must be validated if such circular shape actually exists within the edge-only image. The test for such points is verified using $\mathbf{S} = \{\mathbf{s}^1, \mathbf{s}^2, \ldots, \mathbf{s}^{Ns}\}$, where Ns represents the number of testing points recalling that the vector $\mathbf{s}^w$ contains the pixel coordinates $s_1^w = x_w$ and $s_2^w = y_w$.

Test **S** is generated by means of Midpoint Circle Algorithm (MCA) [9] which calculates the vector **S** representing a circle from parameters (x_0, y_0) and r. For details refer to [44]. Although the algorithm is considered to be the fastest providing a sub-pixel precision, it is important to assure it does not consider points lying outside the image plane. The matching function, also known as the objective function $J(\mathbf{C})$, represents the error resulting from pixels **S** of a given circle candidate and **C** including only pixels that really exist in the edge image, yielding:

$$J(\mathbf{C}) = 1 - \frac{\sum_{v=1}^{Ns} V(\mathbf{s}^v)}{Ns}, \tag{11}$$

where $V(\mathbf{s}^v)$ is a function that verifies the pixel existence in $\mathbf{s}^v$, such as:

$$V(\mathbf{s}^v) = \begin{cases} 1, & \text{if the pixel in position } (x_v, y_v) \text{ exists,} \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Eq. (11) accumulates the number of identified edge points (points in **S**) that are actually present in the edge image. **Ns** holds for the number of pixels lying on the circle's perimeter that correspond to **C**, currently under testing.

The algorithm seeks to minimize $J(\mathbf{C})$, since a smaller value implies a better response (minimum error) of the "circularity" operator. The optimization process can thus be stopped either because the maximum number of epochs is reached or because the best individual is found.

3.3. EMO implementation

The implementation of the proposed algorithm can be summarized into the following steps:

Step 1 : Canny's filter is applied to obtain image edges, storing them into the $\mathbf{E} = \{\mathbf{e}^1, \mathbf{e}^2, \ldots, \mathbf{e}^{Np}\}$ vector. The *iteration* count is set to 0.

Step 2 : m initial particles are generated (each one holding e_i, e_j and e_k elements, where e_i, e_j and $e_k \in \mathbf{E}$). Particles belonging to a very small or to a quite long radius are eliminated (collinear points are discarded). The objective function $J(\mathbf{C}^p)$ is evaluated to determine the best particle $\mathbf{C}^{best}$(where $\mathbf{C}^{best} \leftarrow \arg\min\{J(\mathbf{C}^p), \forall p\}$).

Step 3 : For a given coordinate $d \in (i, j, k)$, the particle $\mathbf{C}^p$ is assigned into a temporary point **t** to store the initial information. Next, a random number is selected and combined with δ yielding the step length. Therefore, point **t** is moved along the direction ***d*** with the sign being randomly determined. If $J(\mathbf{t})$ is minimized, the particle $\mathbf{C}^p$ is replaced by **t** and the neighborhood searching for a particle p finishes, otherwise $\mathbf{C}^p$ is held. Finally the *current best* particle $\mathbf{C}^{best}$ is updated.

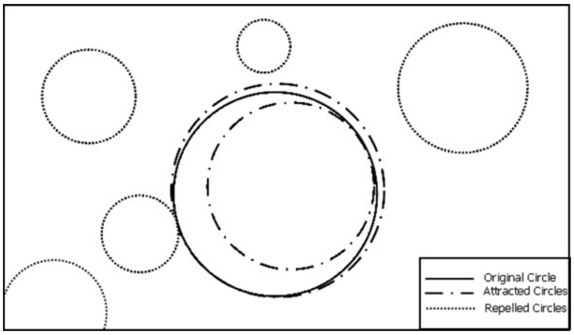

Fig. 5. A circle-based analogy for Coulomb's law.

Step 4 : The charge between particles is calculated using Eq. (2), and its vector force is calculated using Eq. (3). The particle featuring a better objective function holds a bigger charge and therefore a bigger attraction force.

Step 5 : The new particle's position is calculated by Eq. (5). $\mathbf{C}^{best}$ is not moved because it has the biggest force and it attracts others particles to itself.

Step 6 : The *Iteration* index is increased. If *Iteration* = *MAXITER* or if $J(\mathbf{C})$ value is as small as the pre-defined threshold value, then the algorithm is stopped and the flow jumps to step 7. Otherwise, it jumps to step 3.

Step 7 : The best $\mathbf{C}^{best}$ particle is selected from the last iteration.

Step 8 : From the original edge map, the algorithm marks points corresponding to $\mathbf{C}^{best}$. In case of multi-circle detection, it jumps to step 2.

Step 9 : Finally the best particle C^{best}_{Nc} from each circle is used to draw (over the original image) the detected circles, considering Nc as the number of successfully found circles.

Fig. 5 shows an analogy to the Coulomb's law. The original circle to be detected is represented by a solid line while the discontinuous line represents some circles holding the most attractive force, i.e. they have the lowest error value. Other repelled circles are represented by dotted lines as they hold a larger error.

4. Experimental results

In order to evaluate the performance of the EMO-based circle detector, several tests have been documented as follows: (4.1) circle detection, (4.2) shape discrimination, (4.3) multiple circle detection, (4.4) circular approximation, (4.5) approximation from occluded or imperfect circles or arc detection and (4.6) accuracy and computational time.

Each test is provided with a pool of particles of size $m = 10$, a maximum iteration for the local search $ITER = 2$, a step length for the local search $\delta = 3$. Parameters λ_1 and λ_2 are random values uniformly distributed while the maximum iteration value $MAXITER = 20$. For the particles movement (Eq. (5)), the step length λ is also set as a uniformly distributed random number. Finally, the search space (edge-only pixels) implies boundaries $u = 1$, $l = Np$ for each variable: $\mathbf{e}_i$, $\mathbf{e}_j$ and $\mathbf{e}_k$.

4.1. Circle localization

4.1.1. Synthetic images

The experimental setup includes a number of synthetic images of 256 × 256 pixels. Each image has been generated by drawing only a randomly located imperfect circle (ellipse). Some of these images are contaminated by adding noise to

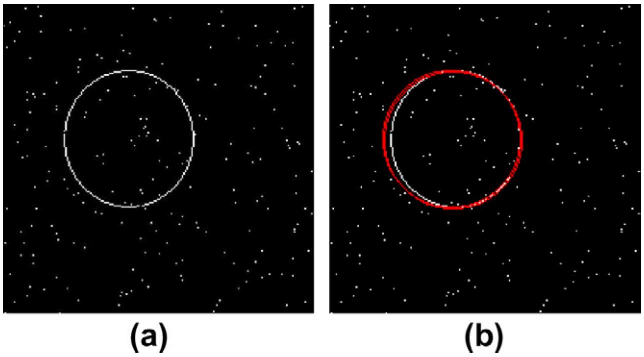

Fig. 6. Circle detection over synthetic images: (a) an original circle image with noise and (b) its corresponding detected circle.

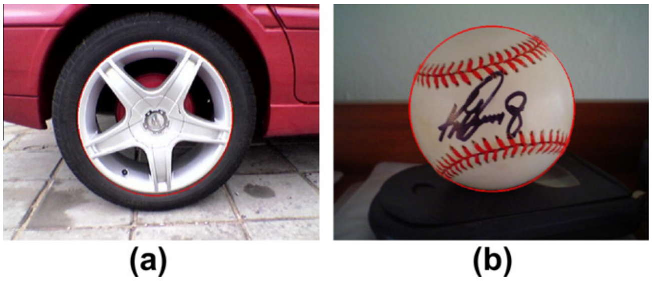

Fig. 7. Circle detection applied to two natural images, (a) the detected circle is shown near the wheel's ring and (b) the detected circle is shown close to the ball's periphery.

increase complexity in the detection process. The parameters to be estimated are the centre of the circle position (x,y) and its radius (r). The algorithm is set to 20 iterations for each test image. In all cases, the algorithm has been able to detect the circle's parameters despite noise presence. The detection is robust to translation and scale, keeping a reasonable low elapsed time (typically under 1 ms). Fig. 6 shows the circle detection over two different synthetic images.

4.1.2. Real-life images

In this experiment we test the circle detection algorithm upon real-life images. Twenty-five images of 640 × 480 pixels are used in the test. All images have been captured using an 8-bit color digital camera and each natural scene includes a circle shape among other objects. All images are pre-processed using an edge detection algorithm before applying the EMO's circle detector. Fig. 7 shows two cases from 25 tested images.

Real-life images rarely contain perfect circles. Therefore the detection algorithm approximates the circle that adapts better to the imperfect circle, i.e. the circle corresponding to the smallest error in the objective function $J(\mathbf{C})$. All results have been statistically analyzed to ease comparison. The detection time for the image shown in Fig. 7(a) is 13.540807 s while it is 27.020633 s for the image in Fig. 7(b). The detection algorithm has been executed over 20 times on the same image (Fig. 7(b)), yielding the same parameter set: $x_0 = 214$, $y_0 = 322$, and $r = 948$. The proposed EMO algorithm was able to converge to the minimum solution as is referred by the objective function $J(\mathbf{C})$.

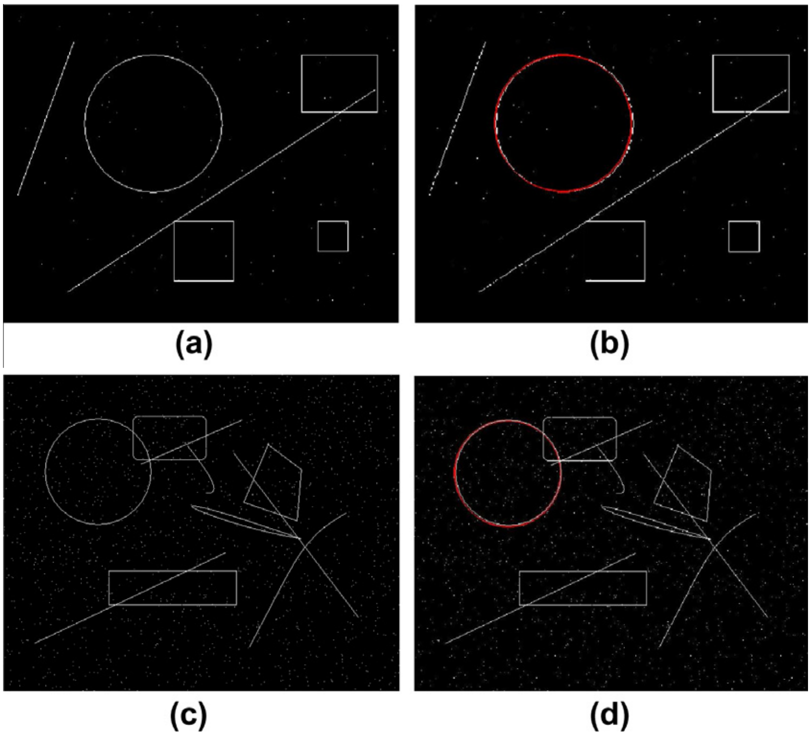

Fig. 8. Circle detection on synthetic images: illustrations (a) and (c) show the original images while (b) and (d) show their corresponding detected circles.

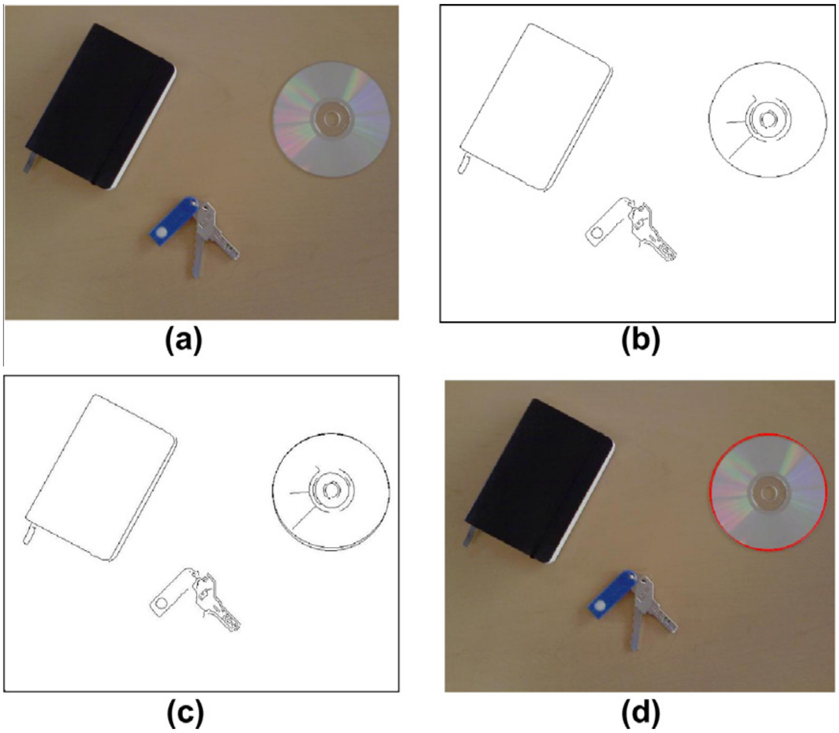

Fig. 9. Different shapes embedded into real-life images, (a) the test image, (b) the corresponding edge map, (c) the detected circle and (d) the detected circle overlaid on the original image.

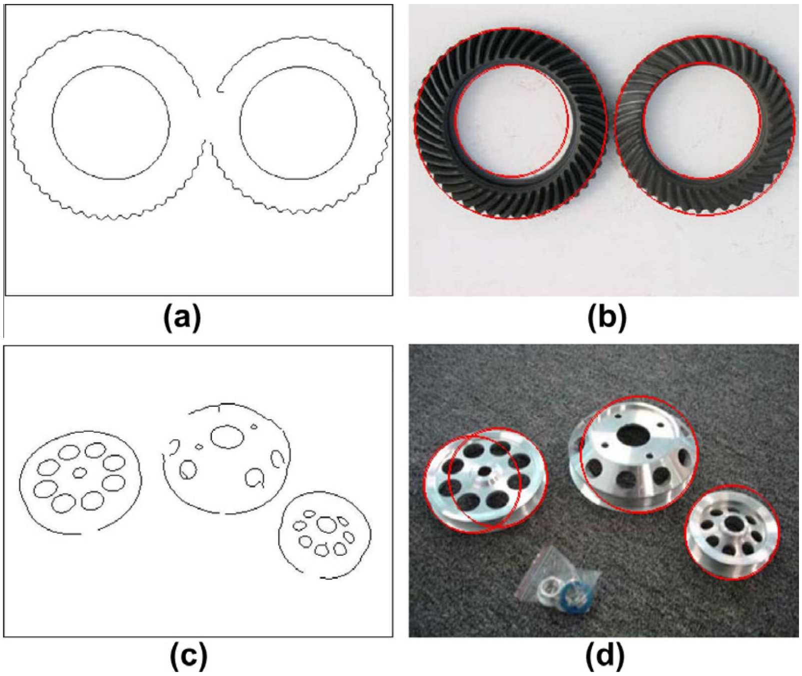

Fig. 10. Multiple circle detection on real images: pictures in (a) and (c) show edge images as they are obtained from applying Canny's algorithm. Images (b) and (d) present original scenes with the overlaid of detected circles being quite evident.

4.2. Shape discrimination

In this section we focus on the ability for detecting circles despite the presence of any other shape on the image. Five synthetic images of 540 × 300 pixels are considered for this experiment. Noise has been added to all images and a maximum of 20 iterations was used for the detection. Two examples of circle detection on synthetic images are shown by Fig. 8.

The same experiment was repeated for some real-life images (see Fig. 9) confirming that the circle detection is completely feasible on natural images despite other shapes being present on the scene.

4.3. Multiple circle detection

The presented approach was also capable of detecting several circles over real-life images. In the first experiment, a maximum number for circular shapes to be detected had been set. The algorithm works on the original edge-only image until the first circle is detected. The first circle represents the circle with the minimum objective function value $J(\mathbf{C})$. Such shape is then masked (eliminated) and the EMO circle detector continues operating over the modified image. The procedure was repeated until the maximum number of detected circular shapes was reached. As a final step, all detected circles were validated by analyzing their circumference continuity as proposed in [4]. Such procedure becomes necessary in case more circular shapes are required in the future.

Fig. 10(a) shows the edge image after applying the Canny's operator and Fig. 10(b) presents the actual image including several detected circles which have been overlaid for presentation. Fig. 10(c) and (d) shows similar cases.

EMO algorithm takes the optimized image from the previous step as input. It is important to consider that the new input does not contain any complete circle, because any other has already been detected and masked at the previous step. Therefore, the algorithm focuses on detecting other potential circles until the maximum of 20 iterations is reached.

4.4. Circular approximation

Since circle detection has been approached as an optimization problem, it is possible to approximate a given shape as the concatenation of a number of circles. By using the multiple-circle feature of the EMO algorithm presented in the previous section, the EMO algorithm may continually find circular shapes that may be gathered to resemble a non-circular shape according to the values from the objective function $J(\mathbf{C})$.

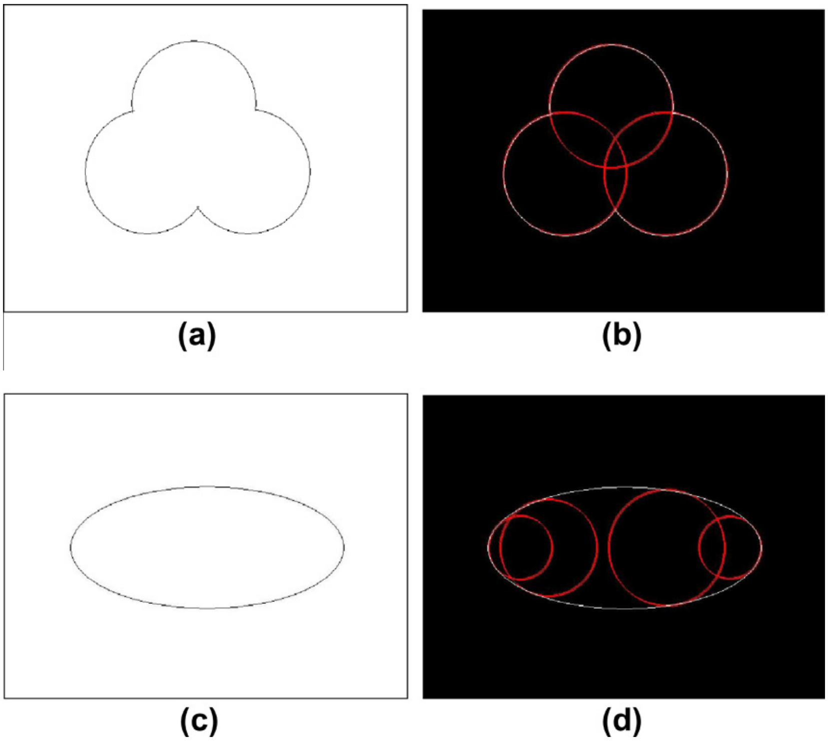

Fig. 11. Circular approximation: (a) the original image, (b) its circular approximation considering 3 circles, (c) another testing image, and (d) its circular approximation considering 4 circles.

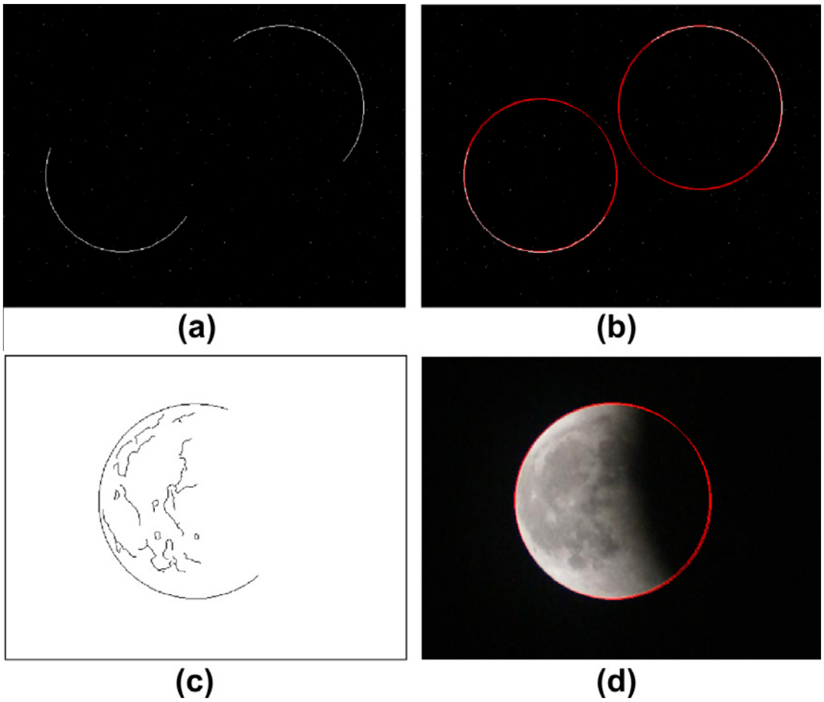

Fig. 12. Approximation of circles from occluded shapes, imperfect circles and arc detection: (a) the original image with 2 arcs, (b) the circle approximation for the first image, (c) occluded natural image of the moon and (d) circle approximation for the moon example.

Fig. 11 shows some examples of circular approximation. Fig. 11(a) presents a circular-like shape which may host a circular superposition. In that case, Fig. 11(b) features the approximation with 3 circles after applying the EMO multi-circle detector. Fig. 11(c) presents an ellipse that has been approximated by the concatenation of four circles as shown by Fig. 11(d).

4.5. Circle localization from occluded or imperfect circles and arc detection

Circle detection may also be useful to approximate circular shapes from arc segments, occluded circular segments or imperfect circles, all representing common challenges for typical computer vision problems. Again, EMO algorithm was able to find the circle shape that approaches an arc segment according to the values in the objective function $J(\mathbf{C})$. Fig. 12 shows some examples of such functionality.

4.6. Accuracy and computational time

This section discusses the algorithm's ability to find the best circle considering synthetic images under noise presence. The experiment gets ten contaminated images of 256 × 256 pixels which contain only one circle centre at $x = 128$, $y = 128$, with $r = 64$ pixels. Two kinds of noise distributions are considered: the Salt & Pepper and the Gaussian noise.

Table 1
Data analysis for images containing added Salt & Pepper noise.

Image properties				Results					
				Error sum (Es)				Computational time (s)	
Size	Circle center	Radius	Salt & Pepper level	Total	Mean	Standard deviation	Mode	Mean	Standard deviation
256 × 256	(128,128)	64	0.01	0	0	0	0	12.5505	0.5426
256 × 256	(128,128)	64	0.02	0	0	0	0	13.0491	0.7455
256 × 256	(128,128)	64	0.03	0	0	0	0	13.3401	0.7365
256 × 256	(128,128)	64	0.04	0	0	0	0	14.3228	0.655
256 × 256	(128,128)	64	0.05	2	0.0571	0.2355	0	14.2141	0.7982
256 × 256	(128,128)	64	0.06	0	0	0	0	14.8669	0.9604
256 × 256	(128,128)	64	0.07	3	0.0857	0.5071	0	15.3725	0.5147
256 × 256	(128,128)	64	0.08	7	0.2	0.6774	0	14.5373	0.5147
256 × 256	(128,128)	64	0.09	6	0.1714	0.5137	0	14.5747	0.8907
256 × 256	(128,128)	64	0.10	28	0.8	2.1666	0	14.7288	0.8116

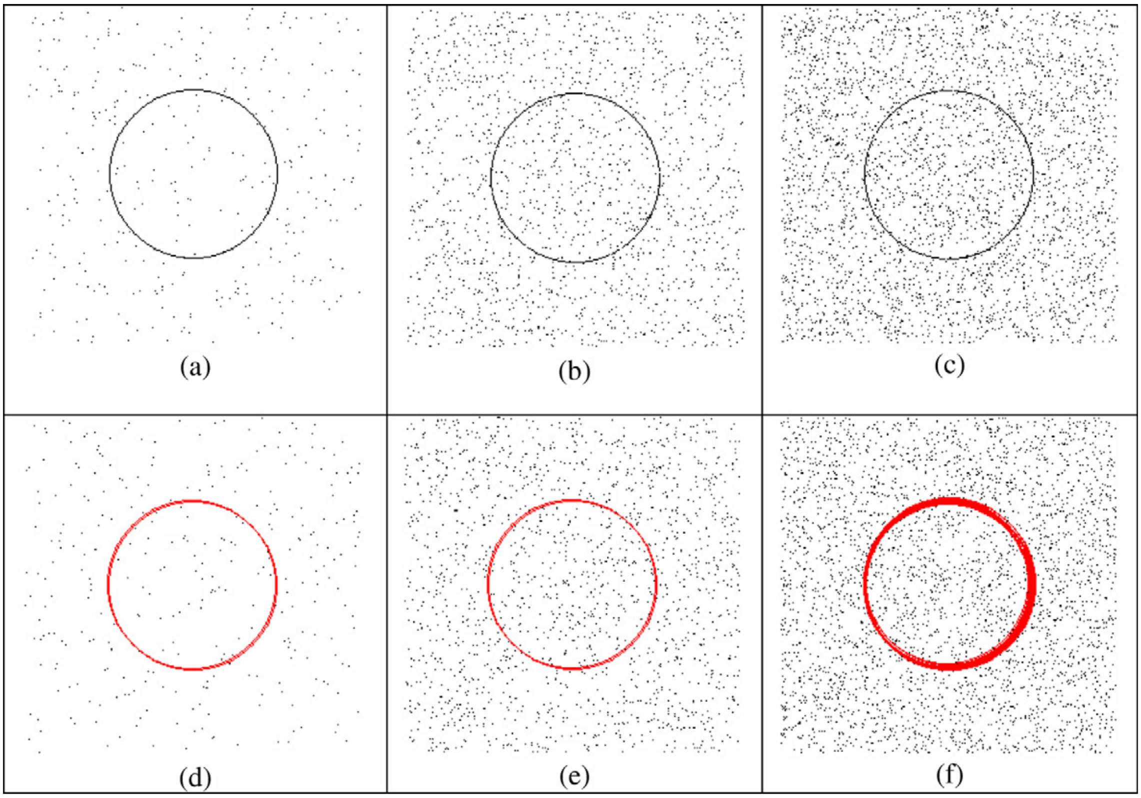

Fig. 13. Test images with added Salt & Pepper noise: (a) image holding 0.01 of added noise, (b) image including 0.05 of added noise, (c) image containing 0.1 of added noise, and (d), (e) and (f) show the final 25 detected circles for each test image.

Table 2
Data analysis following the Gaussian noise application.

Image properties					Results					
					Error sum (Es)				Computational time (s)	
			Gaussian noise							
Size	Circle center	Radius	Mean	Standard deviation	Total	Mean	Standard deviation	Mode	Mean	Standard deviation
256 × 256	(128, 128)	64	0	0.01	0	0	0	0	11.7008	0.6566
256 × 256	(128, 128)	64	0	0.02	0	0	0	0	12.8182	0.9257
256 × 256	(128, 128)	64	0	0.03	0	0	0	0	13.8073	0.6438
256 × 256	(128, 128)	64	0	0.04	6	0.1714	0.5681	0	13.7866	0.8309
256 × 256	(128, 128)	64	0	0.05	10	0.2857	0.825	0	14.0476	1.4691
256 × 256	(128, 128)	64	0	0.06	26	0.7429	0.95	0	14.0622	0.5463
256 × 256	(128, 128)	64	0	0.07	32	0.9143	1.961	0	14.6079	0.4346
256 × 256	(128, 128)	64	0	0.08	158	4.5143	11.688	0	14.1732	0.8198
256 × 256	(128, 128)	64	0	0.09	106	3.0286	6.5462	0	13.9074	0.8165
256 × 256	(128, 128)	64	0	0.1	175	5	8.7447	0	15.1276	0.8606

EMO algorithm iterates 35 times per image and the particle showing the best fitness value is regarded as the best circle matching to the actual one in the image. The process was repeated over 25 times per image to test consistency. The accuracy evaluation was achieved by the Error sum (Es) which measures the difference between the ground truth circle (actual circle) and the detected one. The error sum is defined as follows:

$$Es = |x_d - x_{true}| + |y_d - x_{true}| + |r_d - r_{true}|, \tag{13}$$

where x_{true}, y_{true}, r_{true} are the coordinates of the centre and the radius value of the ground truth circle in the image. Moreover x_d, y_d, r_d correspond to the values of centre and radius respectively after being detected by the method.

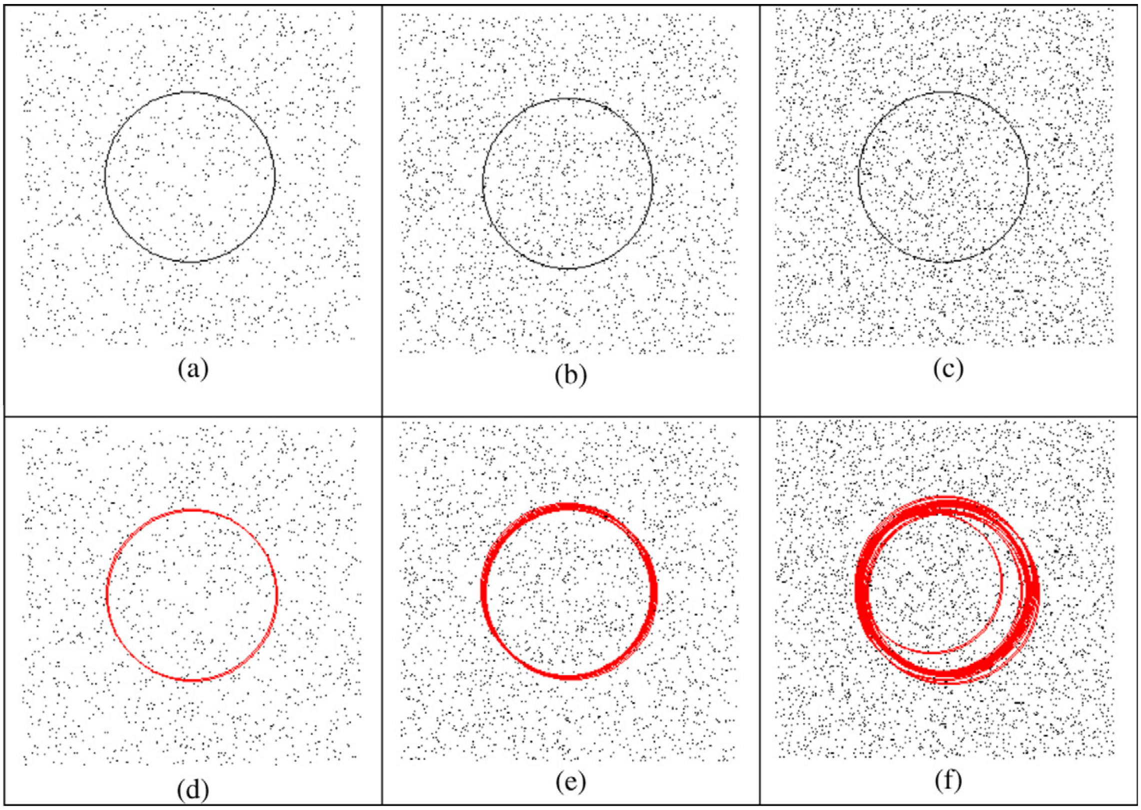

Fig. 14. Test images contaminated with Gaussian noise: (a) image holding 0.01 added noise, (b) image including 0.05 added noise, (c) image containing 0.1 added noise while images, (d), (e) and (f) are showing the final 25 detected circles as overlaid for each test image.

Table 3
Experimental results after applying the EMO circle detector to real-life images.

Image properties		Results					
		Matching error (%)				Computational time (s)	
Image name	Size	Total	Mean	Standard deviation	Mode	Mean	Standard deviation
Cue ball	430 × 473	28.1865	0.805	0.0061	0.8035	26.6641	1.4018
Street lamp	474 × 442	19.0638	0.545	0.0408	0.5326	23.4763	1.7857
Wheel	640 × 480	22.265	0.636	0.0062	0.6336	31.356	1.4351

The first experiment considers contaminated images doped with Salt & Pepper noise. The EMO parameters are: maximum iterations *MAXITER* = 35; for the local search, $\delta = 4$ and *ITER* = 4. The added noise is produced by MatLab©, considering noise levels between 1% and 10%. The resulting values for ***Es*** and the computing elapsed time are reported in Table 1. Fig. 13 shows three different images as examples, including 25 detected circles that are overlaid on each original image. It is evident from Fig. 13 that a higher added noise yields a higher dispersion on the detected shapes.

The second experiment explores the algorithm's application to images contaminated with Gaussian noise. By definition, Gaussian noise requires a threshold value to convert each value to binary pixels yielding an image exhibiting more noise than the Salt & Pepper contamination. Addition of Gaussian noise falls between 1% and 10%. The resulting values for ***Es*** and the computing elapsed time are reported in Table 2. Fig. 14 shows three different images as examples, including the overlay of the 25 detected circles. Again, it is evident that the dispersion on the detected circles increases proportionally to the added noise.

A similar test has been applied to different real images. It is important to mention that the Error sum (Es) is not used because the coordinates for the centre and the radio for each circle are unknown. Table 3 shows the results after applying the EMO circle detector considering the algorithm for natural image presented in Fig. 15. The final 25 detected circles are overlaid over the original images.

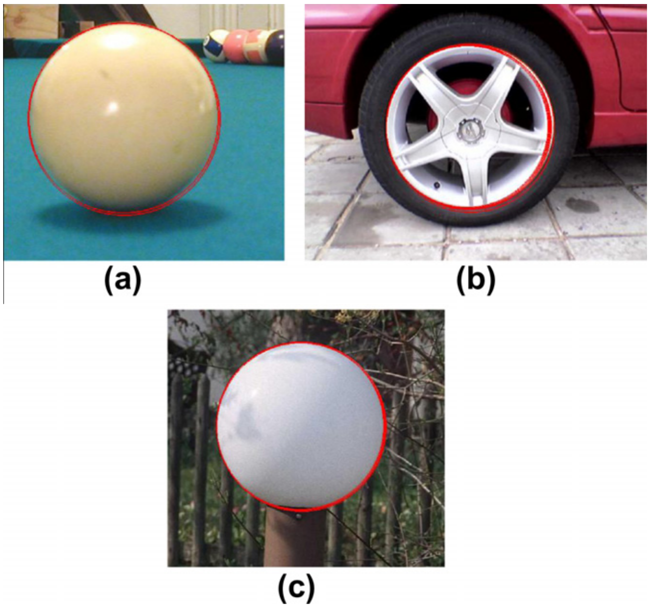

Fig. 15. Real-life images used in the experiments, (a) cue ball, (b) wheel, (c) street ball, including the overlaid detected circles.

5. Conclusions

This paper has presented an algorithm for the automatic detection of circular shapes among cluttered and noisy images with no consideration of the conventional Hough transform principles. The work describes a circle detection method which is based on a nature-inspired technique known as the Electro-magnetism Optimization (EMO). It is a heuristic method for solving complex optimization problems inspired by electro-magnetism principles. To the best of our knowledge, EMO algorithm has not yet been applied to any image processing task up to date. The algorithm uses the encoding of three non-collinear edge points as candidate circles (x,y,r) in the edge-only image of the scene. An objective function evaluates if such candidate circles (charged particles) are actually present. Guided by the values of this objective function, the set of encoded candidate circles are evolved using the EMO algorithm so that they can fit into the actual circles on the edge-only map of the image. From our results, the EMO approach has been capable of effectively detecting circular shapes embedded into complex images with little visual distortion despite the presence of noisy background pixels and still shows good accuracy and consistency.

An important feature is to consider the circle detection problem as an optimization approach. Such a view enables the EMO algorithm to find circle parameters according to $J(\mathbf{C})$ instead of voting all possible chances for occluded or imperfect circles as other methods do.

Although Hough transform methods for circle detection also uses three edge points to cast one vote for the potential circular shape in the parameter space, they would require huge amounts of memory and longer computational times to obtain a sub-pixel resolution. In the HT-based methods, the parameter space is quantized and the exact parameters for a valid circle are often not equal to the quantized values. This fact evidently yields a non-exact determination for a circle actually present in the image. However, the proposed EMO method does not employ the quantization of the parameter space. In our approach, the detected circles are directly obtained from Eqs. (6)–(9) effectively detecting the circle with sub-pixel accuracy.

Although the results offer evidence to demonstrate that the EMO method can yield good results on complicated and noisy images, the aim of our paper is not to devise a circle-detection algorithm that could beat all currently available circle detectors, but to show that electro-magnetism systems can be effectively considered as an attractive alternative for detecting circular shapes.

Acknowledgments

Humberto Sossa would like to thank SIP-IPN under Grant No. 20100468 for its economical support. The authors thank the European Union, the European Commission and CONACYT for the support. This paper has been prepared by an economical support of the European Commission under grant FONCICYT 93829. The content of this paper is an exclusive responsibility of the UDEG and the CIC-IPN and it cannot be considered a reflection of the European Union position. The proposed algorithm is part of the vision system used by a biped robot supported under the grant CONACYT CB 82877.

References

[1] F.A. Andalóa, P.A.V. Miranda, R.S. Torres, A.X. Falcão, Shape feature extraction and description based on tensor scale, Pattern Recognition 43 (2010) 26–36.
[2] N. Andrei, Acceleration of conjugate gradient algorithms for unconstrained optimization, Applied Mathematics and Computation 213 (2009) 361–369.
[3] T.J. Atherton, D.J. Kerbyson, Using phase to represent radius in the coherent circle Hough transform, in: Proceedings of the IEEE Colloquium on the Hough Transform, IEEE, London, 1993.
[4] V. Ayala-Ramirez, C.H. Garcia-Capulin, A. Perez-Garcia, R.E. Sanchez-Yanez, Circle detection on images using genetic algorithms, Pattern Recognition Letters 27 (2006) 652–657.
[5] X. Baia, X. Yangb, L. Jan-Latecki, Detection and recognition of contour parts based on shape similarity, Pattern Recognition 41 (2008) 2189–2199.
[6] J. Becker, S. Grousson, D. Coltuc, From Hough transforms to integral transforms, in: Proceedings of the International Geoscience and Remote Sensing Symposium, IGARSS-02, 2002, pp. 1444–1446.
[7] C. Blum, Ant colony optimization: introduction and recent trends, Physics of Life Reviews 2 (2005) 353–373.
[8] G. Bongiovanni, P. Crescenzi, Parallel simulated annealing for shape detection, Computer Vision and Image Understanding 61 (1995) 60–69.
[9] J.E. Bresenham, A linear algorithm for incremental digital display of circular arcs, Communications of the ACM 20 (1977) 100–106.
[10] U.K. Chak, Genetic and evolutionary computing, Information Sciences 178 (2008) 4419–4420.
[11] W. Dua, B. Li, Multi-strategy ensemble particle swarm optimization for dynamic optimization, Information Sciences 178 (2008) 3096–3109.
[12] M. Fischer, R. Bolles, Random sample consensus: a paradigm to model fitting with applications to image analysis and automated cartography, CACM 24 (1981) 381–395.
[13] M. Graña, Evolutionary algorithms, Information Sciences 133 (2001) 101–102.
[14] T. Gruber, Collective knowledge systems: where the social web meets the semantic web. Web semantics: science, Services and Agents on the World Wide Web 6 (2008) 4–13.
[15] J.H. Han, L.T. Koczy, T. Poston, Fuzzy Hough transform. In: Proceedings of the 2nd International Conference on Fuzzy Systems, 1993, pp. 803–808.
[16] M. Hongwei, Handbook of Research on Artificial Immune Systems and Natural Computing: Applying Complex Adaptive Technologies, IGI Global, 2009.
[17] B. İlker, S. Birbil, F. Shu-Cherng, An electromagnetism-like mechanism for global optimization, Journal of Global Optimization 25 (2003) 263–282.
[18] B. İlker, S. Birbil, F. Shu-Cherng, R.L. Sheu, On the convergence of a population-based global optimization algorithm, Journal of Global Optimization 30 (2004) 301–318.
[19] J. Jhen-Yan, L. Kun-Chou, Array pattern optimization using electromagnetism-like algorithm, AEU – International Journal of Electronics and Communications 63 (2009) 491–496.
[20] D. Karaboga, B. Akay, A comparative study of artificial bee colony algorithm, Applied Mathematics and Computation 214 (2009) 108–132.
[21] C.H. Lee, F.K. Chang, Fractional-order PID controller optimization via improved electromagnetism-like algorithm, Expert Systems with Applications 37 (2010) 8871–8878.
[22] Y.H. Lina, C.H. Chen, Template matching using the parametric template vector with translation, rotation and scale invariance, Pattern Recognition 41 (2008) 2413–2421.
[23] J. Liu, K. Tsui, Toward nature-inspired computing, Communications of the ACM 49 (2006) 59–64.
[24] V. Loia, Soft computing meets agents, Information Sciences 176 (2006) 1101–1102.
[25] E. Lutton, P. Martinez, A genetic algorithm for the detection 2-D geometric primitives on images, in: Proceedings of the 12th International Conference on Pattern Recognition, 1994, pp. 526–528.
[26] P. Lévy, From social computing to reflexive collective intelligence: the IEML research program, Information Sciences 180 (2010) 71–94.
[27] J.A. Martín, M. Santos, J. de Lope, Orthogonal variant moments features in image analysis, Information Sciences 180 (2010) 846–860.
[28] A. Moliton, Basic Electromagnetism and Materials, Springer-Verlag, 2007.
[29] H. Muammar, M. Nixon, Approaches to extending the Hough transform, in: Proceedings of the International Conference on Acoustics, Speech and Signal Processing ICASSP-89, 1989, pp. 1556–1559.
[30] Z. Naji-Azimi, P. Toth, L. Galli, An electromagnetism metaheuristic for the unicost set covering problem, European Journal of Operational Research 205 (2010) 290–300.
[31] B. Naderi, R. Tavakkoli-Moghaddam, M. Khalili, Electromagnetism-like mechanism and simulated annealing algorithms for flowshop scheduling problems minimizing the total weighted tardiness and makespan, Knowledge-Based Systems 23 (2010) 77–85.
[32] H. Qia, K. Lia, Y. Shena, W. Qu, An effective solution for trademark image retrieval by combining shape description and feature matching, Pattern Recognition 43 (2010) 2017–2027.
[33] A. Rocha, E. Fernandes, Hybridizing the electromagnetism-like algorithm with descent search for solving engineering design problems, International Journal of Computer Mathematics 86 (2009) 1932–1946.
[34] A. Rocha, E. Fernandes, Modified movement force vector in an electromagnetism-like mechanism for global optimization, Optimization Methods & Software 24 (2009) 253–270.
[35] P.L. Rosin, Further five point fit ellipse fitting, in: Proceedings of the 8th British Machine Vision Conference, Cochester, UK, 1997, pp. 290–299.
[36] P.L. Rosin, H.O. Nyongesa, Combining evolutionary, connectionist, and fuzzy classification algorithms for shape analysis, in: S. Cagnoni et al. (Eds.), Proceedings of the EvoIASP, Real-World Applications of Evolutionary Computing, 2000, pp. 87–96.
[37] G. Roth, M.D. Levine, Geometric primitive extraction using a genetic algorithm, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (1994) 901–905.
[38] K. Schindlera, D. Suterb, Object detection by global contour shape, Pattern Recognition 41 (2008) 3736–3748.
[39] M.C. Schut, On model design for simulation of collective intelligence, Information Sciences 180 (2010) 132–155.
[40] D. Shaked, O. Yaron, N. Kiryati, Deriving stopping rules for the probabilistic Hough transform by sequential analysis, Computer Vision Image Understanding 63 (1996) 512–526.
[41] D. Teodorović, Swarm intelligence systems for transportation engineering: principles and applications, Transportation Research Part C: Emerging Technologies 16 (2008) 651–667.
[42] J. Tiana, W. Yub, L. Ma, AntShrink: ant colony optimization for image shrinkage, Pattern Recognition Letters 31 (2010) 1751–1758.
[43] C.S. Tsou, C.H. Kao, Multi-objective inventory control using electromagnetism-like metaheuristic, International Journal of Production Research 46 (2008) 3859–3874.
[44] J.R. Van-Aken, An efficient ellipse drawing algorithm, CG& A 4 (1984) 24–35.
[45] P. Wu, Y. Wen-Hung, W. Nai-Chieh, An electromagnetism algorithm of neural network analysis an application to textile retail operation, Journal of the Chinese Institute of Industrial Engineers 21 (2004) 59–67.
[46] L. Xu, E. Oja, P. Kultanen, A new curve detection method: randomized Hough transform (RHT), Pattern Recognition Letters 11 (1990) 331–338.
[47] J. Yao, N. Kharma, P. Grogono, Fast robust GA-based ellipse detection, in: Proceedigns of the 17th International Conference on Pattern Recognition ICPR-04, Cambridge, UK, 2004, pp. 859–862.
[48] H. Yin, W. Huang, Adaptive nonlinear manifolds and their applications to pattern recognition, Information Sciences 180 (2010) 2649–2662.
[49] C. Ying-ping, J. Pei, Analysis of particle interaction in particle swarm optimization, Theoretical Computer Science 411 (2010) 2101–2115.
[50] H. Yuen, J. Princen, J. Illingworth, J. Kittler, Comparative study of Hough transform methods for circle finding, Image and Vision Computing 8 (1990) 71–77.
[51] S. Yuen, C. Ma, Genetic algorithm with competitive image labelling and least square, Pattern Recognition 33 (2000) 1949–1966.

[52] A. Yurtkuran, E. Emel, A new Hybrid electromagnetism-like algorithm for capacitated vehicle routing problems, Expert Systems with Applications 37 (2010) 3427–3433.

[53] Q. Zhang, M. Mahfouf, A nature-inspired multi-objective optimization strategy based on a new reduced space search ing algorithm for the design of alloy steels, Engineering Applications of Artificial Intelligence (2010), doi:10.1016/j.engappai.2010.01.017.

[54] X. Zhang, P.L. Rosin, Superellipse fitting to partial data, Pattern Recognition 36 (2003) 743–752.