Applied Intelligence (2019) 49:2001–2016
https://doi.org/10.1007/s10489-018-1372-2

Automatic circle detection on images using the Teaching Learning Based Optimization algorithm and gradient analysis

A. Lopez-Martinez[1] · F. J. Cuevas[1]

Published online: 20 December 2018

Abstract
Circle extraction is usually a previous task used in different applications related to biometrics, robotics, medical image analysis among others. Solutions based on meta-heuristic approaches, such as evolutionary and swarm-based algorithms, have been adopted in order to overcome the main deficiencies of Hough Transform methods. In this paper, the task of circle detection is presented as an optimization problem, where each circle represents an optimum within the feasible search space. To this end, a circle detection method is proposed based on the Teaching Learning Based Optimization algorithm, which is a population-based technique that is inspired by the teaching and learning processes. Additionally, improvements to the evolutionary approach for circle detection are obtained by exploiting gradient information for the construction of the search space and the definition of the objective function. To validate the efficacy of the proposed circle detector, several tests using noisy and complex images as input were carried out, and the results compared with different approaches for circle detection.

Keywords Circle detection · Optimization · TLBO algorithm · Computer vision · Meta-heuristics · Pattern recognition

1 Introduction

The importance of automatic circle detection becomes evident considering the variety of computer vision applications for which circle extraction is a key element. For example, some biometric authentication systems rely largely on an accurate iris segmentation [1, 2]. In robotics, on the other hand, circular markers have been used for visual-based localization purposes [3, 4]. Similar applications seize on man-made objects such as traffic signs [5] for autonomous navigation tasks. Regarding industrial and manufacturing applications, circle extraction has been used for autonomous inspection [6, 7], and remote sensing processing used for oil circular depots detection [8, 9]. Given the relevance to this problem, different methods for circle detection have been investigated in order to offer a solution.

The methods based on the Hough transform (HT) have been extensively used due to their effectiveness. The classical Circular HT (CHT) performs an exhaustive exploration of the search space by mapping every point in the input image into a parametric space, replacing the difficult pattern detection in the image space into a local peak searching in the parameter space. This approach makes the CHT robust to noise, yet computationally inefficient. Consequently, some new approaches based on the CHT principles have been proposed to improve both storage and computational requirements. For instance, some methods modify the architecture of the traditional CHT in order to propose a multi-threaded implementation [10], or a FPGA implementation of the CHT [11]. Other approaches exploit the CHT principles in combination with gradient information and curvature analysis. For instance, the Curvature Aided Hough transform for Circle Detection (CACD) algorithm estimates the circle center and radius by adaptively estimating curvature radius [12]. The CACD approach estimates the curvature on the edge-only image to perform a one-to-one vote using accumulator arrays of different radius ranges. Similarly, the work in [13] uses first-order and second-order derivatives to compute a one-to-one dense CHT; however, the method does not performs edge extraction as a preprocessing step.

There are some other recent approaches that also exploit gradient information along with other geometrical properties of circles to perform circle detection. For instance, some work use the fact that the center of a circle lies along

✉ A. Lopez-Martinez
alan.lopez@cio.mx

1 Centro de Investigaciones en Optica, A.C., Loma del Bosque 115, Leon, Mexico

the normal direction of edgels, and that the intersection of these normal lines is the center of the circle [14, 15]. In the work of [16], on the other hand, an inverted gradient hash map is used to detect circles by using the fact that pairs of edgels on a circumference differ in orientation by 180 degrees. Another method implements an isosceles triangles sampling to estimate centers and radius [17]. The method performs a refinement process using chords and a linear compensation based on gradient information of edgels.

Other approaches rely on a sampling procedure to reduce the computational time for the circle detection task. An example is a recent method called GRCD-R (Gradient-based Randomized Circle Detection with Refinement) [18]. This method randomly selects four non-collinear edgels. Then, gradient orientation is used to test if the gradient vectors point to the center of a candidate circle. After that validation, three edgels are used to compute a circle, and the last one determines whether the candidate circle is a possible circle in function of its distance to the candidate circumference. Finally, a voting procedure is performed to decide whether possible circles are to be upgraded to detected circles. The work in [19] adds further constraints based on the curvature of the isophotes. Isophotes are curves connecting pixels in the image with equal intensity, whose properties make them particularly suitable for object detection.

Other methods use arc segments instead of edgels to find circles and ellipses in binary images [20, 21]. EDCircles [22], for instance, uses edges extracted by an edge segment extractor named EDPF (Edge Drawing Parameter Free). The edges are then converted into lines, and lines into circular arcs, which are joined together using heuristic algorithms to fit circles and ellipses. To eliminate false detections, EDCircles uses a contrario validation step based on the Helmholtz principle. Their idea is to compute the level line orientation field (which is orthogonal to the gradient orientation field) of a given image, and look for a contiguous set of pixels having similar level line orientation.

The methods described above are efficient, but sometimes they produce false and missing detections. Thence, evolutionary and swarm-based algorithms have been studied as a meta-heuristic manner to perform circle extraction. For example, circle detectors have been proposed using the Differential Evolution approach [23], the Electromagnetism-like optimization (EMO) algorithm [24], the Artificial Bee Colony (ABC) method [25], the Collective Animal Behavoir (CAB) approach [26], the Harmony Search (HS) algorithm [27], and the Clonal Selection Algorithm (CSA) [28]. Unfortunately, despite the fact that many meta-heuristics have been used for the circle detection task, to our knowledge, less effort has been done into exploring the use of gradient information within the search process. In particular, the pattern present in circular shapes within the gradient orientation field has not been used to better guide the search with the objective function.

If the objective function is improperly defined, it can lead to non-acceptable solutions whatever meta-heuristic is used. Most of the reported work evaluates the fitness of an individual by essentially counting the number of pixels in the target edge map that coincide with the perimeter of candidate circles. The fitness value is given by the ratio of candidate to target pixels. They also punish points not exactly on the perimeter of a candidate circle using a displacement factor. A drawback of this approach becomes evident while avoiding small false positive circles and working in noisy conditions. The reason is that this approach tends to favor smaller circles since fewer pixels are required to define a high ratio.

Another drawback of circle detectors based on meta-heuristics is that they need the tuning of several parameters. For these methods, a set of algorithm-specific parameters must be tuned appropriately to accomplish good accuracy and high detection rate when dealing with different image conditions, such as variations in illumination and blurring boundaries. For instance, the DE approach requires the tunning of the differential weight, and the crossover probability. The EMO algorithm, on its part, needs the tunning of the step length. ABC requires tuning of the number of bees (employed, scout, and onlookers), limit, and others. CAB, on its part, requires the probability of attraction, the probability of random movement, and the elite size. Similarly, HS requires tunning of the harmony memory and pitch adjusting rates, and the number of improvisations. In the case of CSA, the mutation rate is to be tuned, along with the clonal size, the length of the antibody and others.

To overcome the aforementioned issues, in this paper we propose, differently from previous meta-heuristic-based circle detectors, the utilization of gradient information to better customize the search space, and to improve the search guidance provided by the objective function. Additionally, in the interest of reducing the number of parameters to tune, we use another natural phenomena inspired optimizer, namely, the Teaching Learning Based Optimization (TLBO) algorithm [29]. The TLBO algorithm does not require the tuning of many parameters in comparison with other meta-heuristics, since only the number of iterations and the size of the population are needed, and no other algorithm-specific parameters need to be tuned.

The rest of the paper is organized as follows. Section 2 describes the TLBO algorithm as used in this work. Section 3 presents the use of the TLBO algorithm for circle detection. Section 4 shows the performance of the algorithm through some tests and comparisons. And finally, Section 5 states the conclusions and future work.

Algorithm 1 TLBO algorithm

```
1  Algorithm TLBO
2  Initialize the population of $N$ students $X$ randomly.
3  Compute the fitness $F(X)$ of each student in the
   population.
4  for each iteration do
5      Calculate the mean of each student in the
       population: $(X_{mean})$.
6      Find out the best solution: $(X_{teacher})$.
7      $r \leftarrow$ rand(0,1)
8      $TF \leftarrow$ round(1+rand(0,1))
9      for each $X_i$ in population do
10         Start teacher phase:
11         $X_{new} \leftarrow X_i + r \cdot (X_{teacher} - TF \cdot X_{mean})$
12         if $(F(X_{new}) > F(X_i))$ then
13             $X_i \leftarrow X_{new}$
14         Start learner phase:
15         Select randomly another student $X_j$ with $i \neq j$
16         if $(F(X_i) > F(X_j))$ then
17             $X_{new} \leftarrow X_i + r \cdot (X_i - X_j)$
18         else
19             $X_{new} \leftarrow X_i + r \cdot (X_j - X_i)$
20         if $(F(X_{new}) > F(X_i))$ then
21             $X_i \leftarrow X_{new}$
22 Find out the best solution: $(X_{teacher})$.
23 Select $X_{teacher}$ as the final solution.
```

2 Teaching Learning Based Optimization algorithm

The TLBO algorithm starts with an initialization procedure, where N random solutions (students) X_i $i \in \{1, \cdots, N\}$ are generated within the search space. In the TLBO context, the student X_i of the population is a real-valued vector with D elements, where D is the dimension of the problem and is used to represent the number of subjects that an individual enrolls for. The algorithm tries to improve the population by changing individuals during two consecutive phases: The teacher phase and the learner phase. The teacher phase aims to increase the knowledge level of the whole class and to help students individually to get better grades. The learner phase, on its part, attempts knowledge increase through the interaction between students. The algorithm is terminated after a certain number of iterations is completed.

Within the teacher phase, the best individual (best solution) is assigned as the teacher $X_{teacher}$. The algorithm attempts to improve other individuals by moving their position towards the position of the teacher by taking into account the current mean value of the individuals. The student position X_i is updated by:

$$X_{new} = X_i + r \cdot (X_{teacher} - T_F \cdot X_{mean}), \tag{1}$$

where r is a real random number between 0 and 1 and TF, called the teaching factor, can be either 1 or 2 and is decided randomly with equal probability. The former equation indicates how the improvement of student X_i may be influenced by the difference between the knowledge of the teacher and the qualities of all students. Thus, X_i is replaced by X_{new} if the latter gives better fitness value. The factors TF and r contribute to the exploration and exploitation capabilities of the TLBO algorithm. When $TF = 1$ and r approximates to 1, the individuals try to approximate the teacher, thus exploiting the search space near to the teacher. Conversely, when $TF = 2$, the individuals tend to explore far from the teacher according to the value of r.

During the second and final phase, the learner phase, a student learns with the help of other students. Hence, student X_i tries to improve its knowledge by learning from an arbitrary student X_j. In the case that X_j is better than X_i, X_i is moved towards X_j according to:

$$X_{new} = X_i + r \cdot (X_j - X_i). \tag{2}$$

Otherwise, it is moved away from X_j according to:

$$X_{new} = X_i + r \cdot (X_i - X_j). \tag{3}$$

The objective of this phase is to attain knowledge transfer from a more qualified student to a less qualified student. To this end, X_i is replaced by X_{new} if the latter gives better fitness value.

The TLBO algorithm is a successful meta-heuristic method for solving complex optimization problems, still keeping a simple structure and an easy implementation. Thence, several engineering and scientific applications using this meta-heuristic have been published [29–32]. The simplest form of the TLBO process is described in Algorithm 1. The proposed method adopts this process for circle extraction as explained in the following section.

3 Circle detection using the TLBO algoritm

In this section, we describe the proposed approach for circle extraction. Our method aims different objectives: To have high detection rate and good accuracy in images with cluttered textures, blurring boundaries, low contrast and occluded circles; to detect multiple circles; to work with real images; and to produce a few or no false detections.

In order to achieve these objectives, the TLBO algorithm is used, and gradient information is exploited. Since our detector is based on a meta-heuristic, we describe the following elements: The search space organization, the individual representation, and the objective-function definition. Further, we explain the use of gradient orientation for the particular case of circle detection.

3.1 Search space organization

Regarding the organization of the search space, in previous works [23–26, 28], edge extraction is first performed, to then organize the search space in a set where all edgel locations are indexed one after the other as the edgels are visited in the edge map from left to right and top to bottom. A different approach involves storing edgel locations randomly within the set [33]. These organizations, as stated in [34], do not ensure that nearby pixels in the edge map are neighbors in the search space. This is an important issue to our approach, since the TLBO algorithm locates better individuals in each iteration by shifting them towards the position of the best one. Thus, neighboring individuals within the search space are expected to belong to the same circle and have similar fitness value. Consequently, we perform a different procedure to enhance the search space for the TLBO algorithm. This improvement is achieved by reducing the size of the search space and by customizing its organization, so that its shape better suits the geometric meaning of the TLBO algorithm.

For the search space definition, we use gradient information extracted from image derivatives. First, image noise is reduced by using a 3 × 3 Gaussian filter. Then, edge extraction is performed in order to obtain one-pixel-wide edges. Image derivatives in horizontal (g_x) and vertical (g_y) directions are performed to extract gradient information, both magnitude and orientation. Gradient magnitude $M(x, y)$ is computed by:

$$M(x, y) = \sqrt{g_x(x, y)^2 + g_y(x, y)^2}, \tag{4}$$

and gradient orientation $O(x, y)$ is obtained by:

$$O(x, y) = atan2\left(g_y(x, y), g_x(x, y)\right). \tag{5}$$

Where $atan2(\cdot)$ is the function that returns the value of the arc tangent of g_y/g_x in radians; the function takes into account the sign of both arguments in order to determine the quadrant.

To perform the image derivatives g_x and g_y, we convolve the image with derivative kernels. To reduce the error in the gradient computation, the 3 × 3 Sobel operators are used since the accuracy of these operators can be up to 0.12 radians in noiseless conditions [35, 36].

We use gradient magnitude to achieve one of the objectives of the search space enhancement, which is to reduce the search space. This reduction is attained in two steps. Within the first step, all pixels with gradient magnitude lower than a threshold T_m are removed. The threshold T_m is computed as follows:

$$T_m = \tau \cdot M_{mean} \tag{6}$$

where M_{mean} is the mean magnitude value for the image as computed by Eq. 4, and $\tau \in [0, 1]$ is the factor that controls the distinctiveness of the magnitude values. When τ is near 1, more pixels are discarded. Conversely, when τ is near 0, less pixels are discarded. Finally, within the second step, the remaining pixels are connected into contours using the border-following algorithm that constructs the adjacency tree of the image [37]. Then, all the contours that are considered too small (i.e. below a threshold T_c pixels) to form a significant part of a circle are removed. To define the threshold T_c, we define a minimum radius r_{min} and compute T_c as follows:

$$T_c = ceiling(2\pi r_{min}). \tag{7}$$

In consequence of the previous steps, all edgels with small magnitude are removed as well as the short linked contours. Hence, the result is a smaller edge map organized into connected contours that are sufficiently large to be part of a circle. Also, gradient orientation of the remaining edgels is available. Figure 1 summarizes the operations of the preprocessing steps for the proposed approach.

The search space is finally organized in a set V_c with edgel locations stored together and ordered according to their respective contour c. Hence, the search space is encoded within the set $V_c = \{c_1, c_2, \cdots, c_n\}$ with n the number of linked contours, and $c_i = \{p_{i_1}(x_{i_1}, y_{i_1}), \cdots, p_{i_{m_i}}(x_{i_{m_i}}, y_{i_{m_i}})\}$ with m_i the number of edgels connected in the i-contour. One result of this organization is an increase of the probability that nearby edgels in the search space belong to the same circle within the edge map, since connected edgels are stored together.

3.2 Individual representation

Each student X of the population encodes a candidate circle C that passes through three different edgels. Thus, to define student X, three different elements are randomly chosen from V_c. Hence, $X = \{i, j, k\}$, with i, j and k being the index positions in the search space encoded in V_c for the edgel positions $p_i(x_i, y_i)$, $p_j(x_j, y_j)$ and $p_k(x_k, y_k)$ in the edge map, respectively.

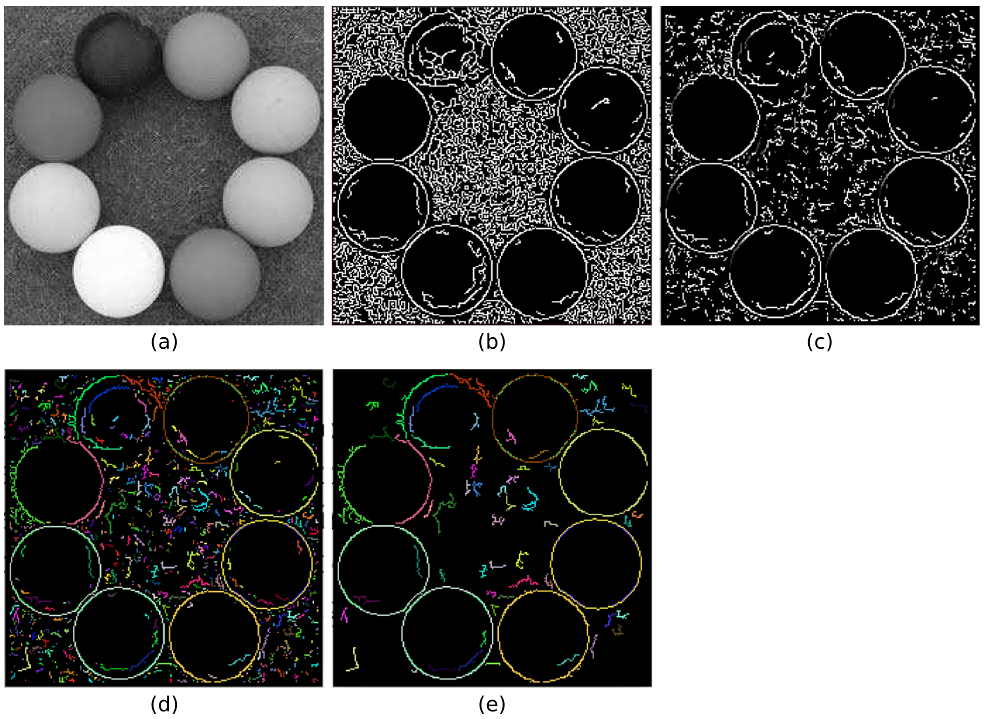

Fig. 1 The preprocessing step aims to reduce the search space and customize its organization. The gray-scale input image in (**a**) is processed to obtain the edge map as shown in (**b**). Then, edge pixels with small gradient magnitude are removed to obtain the image in (**c**). The remaining pixels are connected into contours as shown in (**d**). However, short contours are not considered to be part of the search space and are consequently removed. Thus, the input information to the proposed method is organized into linked contours large enough to be part of a circle as shown in (**e**)

From student X, the center (x_0, y_0) and the radius r of a candidate circle C are calculated by the following equations:

$$x_0 = \frac{\begin{vmatrix} x_j^2 + y_j^2 - (x_i^2 + y_i^2) & 2(y_j - y_i) \\ x_k^2 + y_k^2 - (x_i^2 + y_i^2) & 2(y_k - y_i) \end{vmatrix}}{4((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i))}, \tag{8}$$

$$y_0 = \frac{\begin{vmatrix} 2(x_j - x_i) & x_j^2 + y_j^2 - (x_i^2 + y_i^2) \\ 2(x_k - x_i) & x_k^2 + y_k^2 - (x_i^2 + y_i^2) \end{vmatrix}}{4((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i))}, \tag{9}$$

and

$$r = \sqrt{(x_0 - x_d)^2 + (y_0 - y_d)^2}, \tag{10}$$

where $d \in \{i, j, k\}$. Therefore, a candidate circle $C(x_0, y_0, r)$ can be represented by V_c indexes i, j, and k belonging to the student $X = \{i, j, k\}$, using (8) through (10).

3.3 Objective function

The last element to define is the objective function that guides the search. Usually, reported meta-heuristic approaches for circle detection only use edgel locations in the objective function [23–26, 28]. Conversely, our proposed method improve results by using additional data provided by the gradient orientation. However, the pattern of the gradient orientations within a circumference is not the same for a circle with less intensity than the background and a circle with more intensity than the background. This results in a gradient orientation inconsistency problem that needs to be tackled to reliably use gradient orientation in the definition of the objective function.

3.3.1 Orientation alignment

The gradient orientation inconsistency problem is shown in Fig. 2. This problem entails that for the same circle parameters (x_0, y_0, r), one pixel orientation, say $\pi/2$, may belong to a concave segment as in Fig. 2a; or to a convex segment as in Fig. 2b. In other words, the circle parameters encoded by student X is not enough information to assume whether the orientation vectors should point towards or against the center of the circle.

To solve the orientation inconsistency problem, the work in [18] uses a set of four 9×9 masks to verify whether an edgel belongs to a concave or convex segment. A different mask is used in different sections of the same circumference. Another used strategy involves the computation of second derivatives in order to calculate the contour curvature and decide whether a curve is concave or convex, consequently finding the correct orientation pattern [15]. These techniques are computational expensive or sensitive to noise. Hence, in order to reduce the computational burden, the proposed method partially verifies orientation alignment as now explained.

Instead of verifying whether a pixel belongs to a concave or convex segment, the proposed strategy validates the gradient orientation of a pixel within a circumference *regardless* it lies o n a concave or a convex segment. In regard to Fig. 2, it is clear that for a circle C with parameters

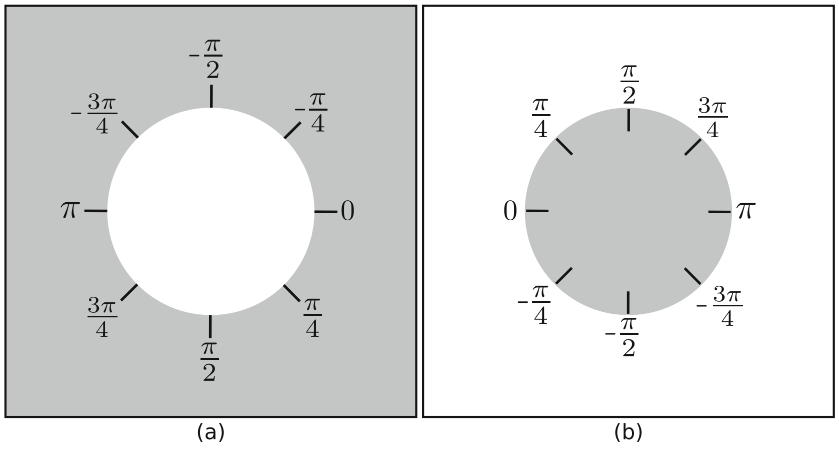

Fig. 2 Gradient orientation inconsistency problem. The orientation pattern obtained from image derivatives is different for a circle **a** brighter than the background, and **b** darker than the background

(x_0, y_0, r), the pixel $p_1(x_0 + r, y_0)$ within the perimeter of C has gradient orientation of either 0 as in Fig. 2a, or π as in Fig. 2b. Likewise, the pixel $p_2(x_0, y_0+r)$ has an orientation of $\pi/2$ as in Fig. 2a, or $-\pi/2$ as in Fig. 2b. For all pixels in the perimeter of C, the absolute difference of the two facing valid orientations is always π. We use this fact to verify orientation alignment.

Given the circle parameters (x_0, y_0, r) encoded by a student X, two valid patterns of gradient orientations can be computed. Taking the case of Fig. 2a for convenience, it is noted that for each pixel $p_i(x_i, y_i)$ within the circumference, the *ideal* gradient orientation $\overline{O}_i$ can be computed by:

$$\overline{O}_i = atan2\,(y_i - y_0, x_i - x_0)\,. \tag{11}$$

Thus, the proposed algorithm computes the degree of orientation alignment of the *actual* orientation O_i of the edgel $p_i(x_i, y_i)$ w.r.t the *ideal* orientation $\overline{O}_i$ using the following metric:

$$M(O_i) = |cos(|O_i - \overline{O}_i|)|. \tag{12}$$

This metric approximates to 1 whether the real gradient vector of p_i points towards or against the center of the circle, as long as its orientation approximates to either one of the two valid orientations of the concave and convex cases. On the other hand, the metric approaches 0 as the real gradient vector becomes perpendicular to either of the two valid orientations. According with this metric, the proposed algorithm classifies orientation alignment into three categories:

$$O_i \text{ alignment} = \begin{cases} \theta\text{-aligned} & \text{if } M(O_i) > cos(\theta) \\ \gamma\text{-aligned} & \text{else if } M(O_i) > cos(\gamma) \\ \text{not aligned} & \text{otherwise} \end{cases} \tag{13}$$

Within this classification, the orientation O_i is θ-aligned when it deviates less than θ radians from either of the two valid orientations. Similarly, the orientation O_i is γ-aligned when it deviates less than γ radians (but more than θ) from either of the two valid orientations. An illustration of the three classifications is shown in Fig. 3.

The gradient orientation pattern is preserved even in the presence of noise. In Fig. 4, the error between an ideal and a real orientation pattern is shown. To compute the error, the circle parameters are manually detected. Then, a valid gradient orientation $\overline{O}_i$ is computed for each edgel within the circumference. The real gradient orientation O_i, on the other hand, is obtained for the same edgels with derivative kernels. As shown in the figure, for this particular image with different circles of different intensities, the majority of edgels are θ-aligned even in the presence of different levels of Gaussian noise.

3.3.2 Objective-function definition

Having defined the orientation alignment, we now describe how it is used in the objective function computation. A well

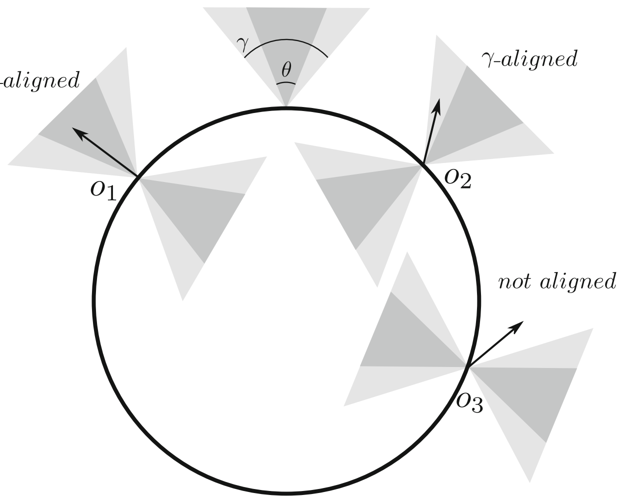

Fig. 3 Orientation alignment. The proposed algorithm categorizes the orientation alignment of each edge pixel into three classifications according to their gradient orientation

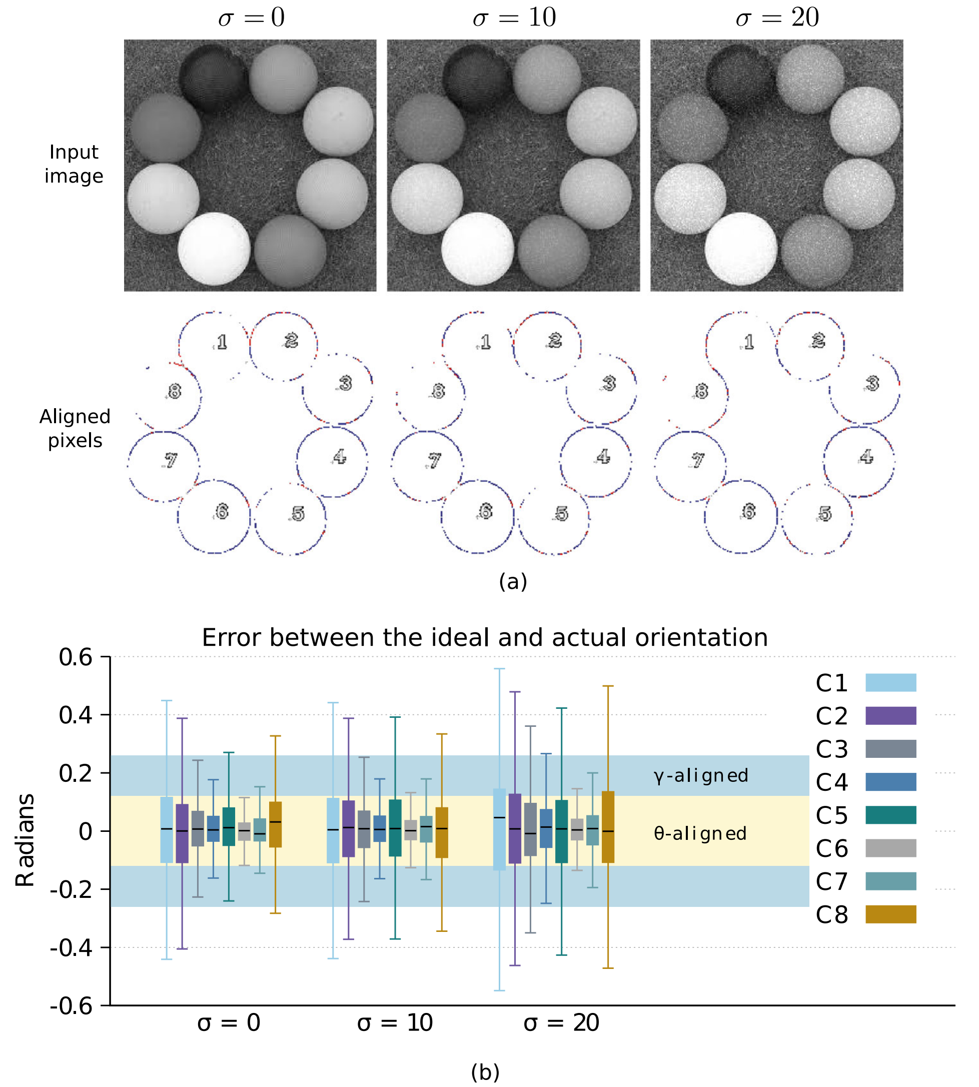

Fig. 4 Orientation alignment test in real images. **a** Original image corrupted with different levels of Gaussian noise. θ-aligned edgels are shown in blue, while γ-aligned edgels in red. **b** Boxplots showing the orientation alignment error of pixels within the circumference of each circle

fitted student is the one that encodes a candidate circle that actually exists within the edge map and has a valid gradient orientation pattern. To score a solution, reported meta-heuristic approaches for circle detection do not use gradient information and proceed as follows. For a candidate circle C, a certain amount of test points in the perimeter of C is generated. The value of the classical objective function, $F_{classical}$, is the ratio of the total number of test points appearing in the edge map to the total number of test points. Variations to this conventional objective function include different penalizations as the edge points recede from the test circle. Unfortunately, when in noisy conditions, the quantity of false positive circles may increase using this approach due to the fact that small circumferences may contain enough pixels in the edge map to give a high ratio. For this reason, in the proposed method, additional information provided by the gradient orientation is used in order to surmount the problem.

Given a candidate circle $C(x_0, y_0, r)$ encoded by a student X, N_c test triplets $T = \{t_1(x_1, y_1, \overline{O}_1), \cdots, t_{N_c}(x_{N_c}, y_{N_c}, \overline{O}_{N_c})\}$ are generated. In the interest of reducing the computational burden and improving the sub-pixel accuracy, the locations of test points are generated using the Midpoint Circle Algorithm (MCA) [38]. The MCA aims to minimize the error between the pixel discrete positions and the continuous candidate circumference. Additionally, it reduces the computational burden of the algorithm by only computing pixel positions within the first octant of the circumference and mirroring the rest. Gradient orientation of each test point, on the other hand, is computed using (11).

Candidate test points within the perimeter of C must exist in the edge map, and their gradient orientation must be aligned. Thus, the proposed fitness function is computed by:

$$F_{proposed}(X) = \frac{\sum_{i=1}^{N_c} E(x_i, y_i, \overline{O}_i)}{N_c}, \tag{14}$$

where $E(\cdot)$ is the function that verifies the pixel existence in $p_i(x_i, y_i)$, and its orientation alignment O_i w.r.t $\overline{O}_i$. E returns zero when the test point $p_i(x_i, y_i)$ is not an edgel. On the other hand, when the pixel p_i belongs to the edge map, E returns:

$$E(x_i, y_i, \overline{O}_i) = \begin{cases} \varphi_1 \cdot M(O_i) & \text{if } \theta\text{-aligned} \\ \varphi_2 \cdot M(O_i) & \text{if } \gamma\text{-aligned} \\ \kappa & \text{if not aligned,} \end{cases} \qquad (15)$$

where $M(O_i)$ is the metric for orientation alignment in (12), and the weights φ_1, φ_2 and κ can be set according to the desired accuracy. We set and fixed the weights empirically to: $\varphi_1 = 1$, $\varphi_2 = 0.5$ and $\kappa = 0.25$; and the threshold angles to $\theta = 0.12$ and $\gamma = 0.26$ radians. Different values modify the relevance of orientation alignment within the search process.

The example of Fig. 5 illustrates the evaluation of two test points p_{T1} and p_{T2} belonging to a circle encoded by student $X = \{i, j, k\}$. Since p_{T1} is θ-aligned, P_{T1} contributes to $F_{proposed}$ by a factor of $\varphi_1 M(o_{T1})$. On the contrary, since the test point P_{T2} is neither θ-aligned nor γ-aligned, it only contributes by a factor of κ. Recall that both P_{T1} and P_{T2} belong to the edge map, however P_{T1} meets the additional orientation test.

3.4 Implementation of TLBO-based circle detector

The whole circle detection algorithm is summarized in Fig. 6. First, the preprocessing step is performed in order to obtain the edge map, gradient information, and the search space encoded in V_c. Then, the TLBO algorithm is executed. The first step of the TLBO algorithm involves generating the initial population. All the solutions within the first population must consist of three non-collinear points. To achieve this, the algorithm first verifies if the denominator in (8) or 9 is different from zero. If this is not the case, at least two of the points are collinear, and that particular combination of points is not added to the population. Later in the process, in the Teacher an Learner phases, the algorithm prevents individuals consisting in collinear points by assigning a fitness equal to zero if the denominator in (8) or (9) is equal to zero. At the end, the teacher of the last iteration of the TLBO algorithm is selected as the solution.

Our proposed method is able to achieve multiple-circle detection. To this end, the proposed method is executed as many times as needed. After one execution, the detected circle is removed from the search space and the edge map. Then, the process is carried out again over the modified search space and edge map. Multiple executions of the whole process are carried out until the algorithm returns a final solution whose fitness value is below a predefined threshold f_{min}. Finally, every detected circle is validated by analyzing the continuity of the detected circumference segments as in [23].

4 Experimental results

In this section, results of the application of the proposed method on real images are reported, and compared with five different approaches that also use gradient information: 1) The OpenCV implementation of a HT-based circle detector; 2) The isophotes RCD algorithm with the same parameters as in [19]; 3) The on-line demo of EDCircles provided by the authors; 4) the Matlab implementation of CACD provided by the authors; and 5) The classical DE algorithm with the same search space and objective function as the TLBO algorithm. We have selected these methodologies in the interest of contrasting our method with other approaches that also use gradient information, while at the same time are based on different methodologies. The OpenCV implementation is a one-to-many CHT, while the CACD is a one-to-one CHT. Isophotes RCD, on its part, is based on an iterative sampling methodology. EDCircles is an arc-based circle detector; and the DE-based circle detector relies on a meta-heuristic.

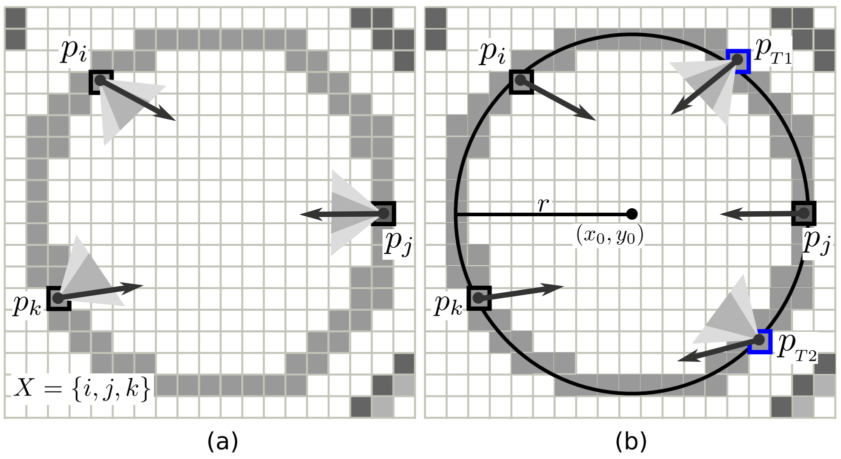

Fig. 5 Fitness evaluation. **a** A candidate solution $X = \{i, j, k\}$ represents a candidate circle $C(x_0, y_0, r)$. **b** Two pixels p_{T1} and p_{T2} within the perimeter of C are evaluated. Both pixels contribute to the fitness of X. However, p_{T1} contributes to a greater extent than p_{T2}, since the former meets the additional orientation test

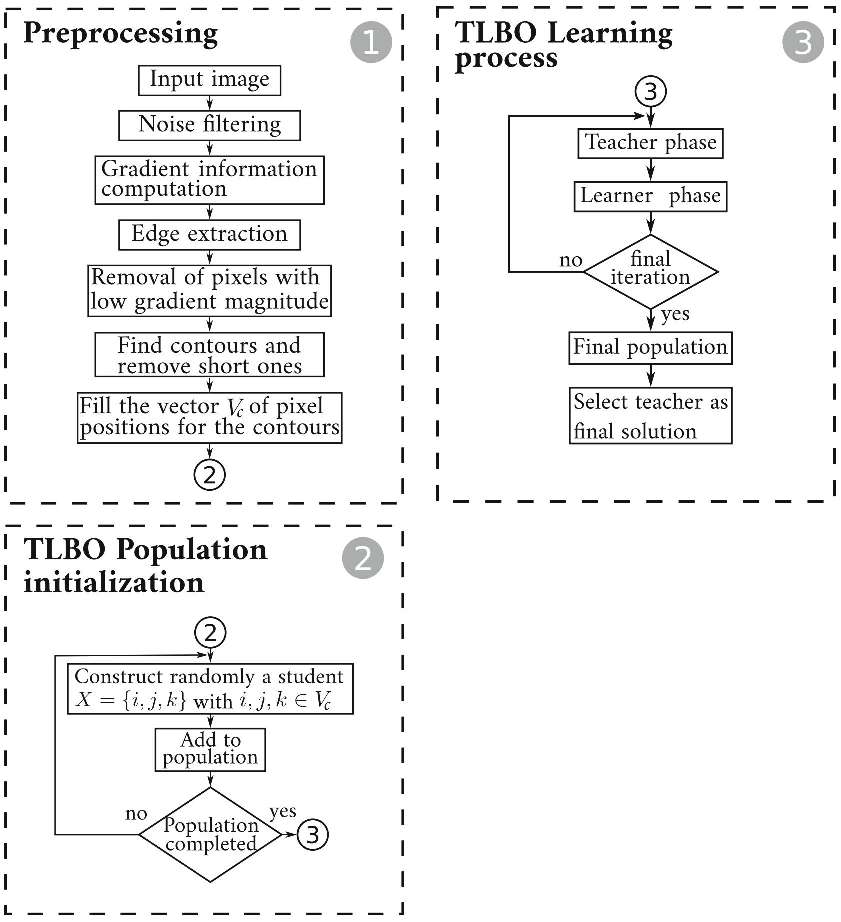
Preprocessing
1
Input image
Noise filtering
Gradient information computation
Edge extraction
Removal of pixels with low gradient magnitude
Find contours and remove short ones
Fill the vector V_c of pixel positions for the contours
2
TLBO Learning process
3
3
Teacher phase
Learner phase
no
final iteration
yes
Final population
Select teacher as final solution
TLBO Population initialization
2
2
Construct randomly a student $X = \{i, j, k\}$ with $i, j, k \in V_c$
Add to population
no
yes
Population completed
3

Fig. 6 General procedure of the proposed meta-heuristic-based circle detector

The experimental setup includes the use of real images commonly used in the literature: Streetlight (544 × 509), Plates (504 × 489), Bowling (467 × 480), Gobang (234 × 231), Cells (519×459), Cell (701×664), Watch (467×480), Cookies (295×292), Eye (190×143), and Insulator (492 × 553). All the experiments were executed on a 2.80GHz Intel Core i7-7700HQ CPU, with a C++ implementation of the algorithm. To perform edge extraction and border following, implementations from OpenCV were utilized.

4.1 Parameter setup

The sets of parameters for the meta-heuristics, shown in Tables 1 and 2, have been chosen empirically in favor of performing a fair comparison. First, the settings for the TLBO algorithm were experimentally defined. Then, the parameters of the DE algorithm that control the number of objective function evaluations (FEs) were set specifically to equate this number for both approaches. Finally, the rest of the DE parameters were set after experimentation. On the other hand, the thresholds for the preprocessing step and the multiple-circle detection process were fixed at $\tau = 0.05$, $T_c = 2\pi \cdot 5$ pixels, and $f_{min} = 0.20$, respectively. The parameters and thresholds for both meta-heuristics remained fixed for every experiment.

As stated above, when the number of FEs is the same for all methods, a fair comparison is guaranteed since all algorithms sample the search space an equal number of times. Otherwise, an algorithm with more FEs has, a priori, more chances to provide a better solution, since it gathers more information of the problem. The number of FEs for each approach is now described.

Table 1 Parameter setup for the DE-based circle detector

Parameter	Value
Number of epochs	200
Population size	50
Differential weight	0.25
Crossover probability	0.80

Table 2 Parameter setup for the TLBO-based circle detector

Parameter	Value
Iterations	100
Population size	50

Given M individuals and N iterations, the number of FEs for the TLBO algorithm is

$$FEs_{TLBO} = (2N + 1) \cdot M. \tag{16}$$

The above equation is easily deduced considering that in the initialization step for the TLBO algorithm, the objective function is evaluated once for each individual; and that at each iteration, two function evaluations are carried out for each individual (one per step: teacher and learner). Distinctively, the DE-based detector only requires one objective function evaluation at each epoch for each individual, and another within the initialization step. Thus, the number of FEs for the DE algorithm is

$$FEs_{ED} = (N + 1) \cdot M. \tag{17}$$

We now describe results related to the convergence of the algorithm, since our approach is based on a meta-heuristic. Then, in the following subsections, we discuss qualitative and quantitative results.

4.2 Convergence analysis

The evolution of the best individual, the mean fitness value and the similarity of the population are examined for every generation. To compute a measure of similarity S for a population P, the student $X = \{i, j, k\}$ is considered as a vector. Thus, the L2-norm of student X is $|X| = \sqrt{i^2 + j^2 + k^2}$. To calculate S, we first compute the L2-norm of each student within P. Then, each output value is divided by the maximum L2-norm value within P. The output of the last computation is a set of values between 0 and 1. We thus define S as the standard deviation of these values. For convenience, S is plotted along with the mean fitness value and the best individual in Fig. 7.

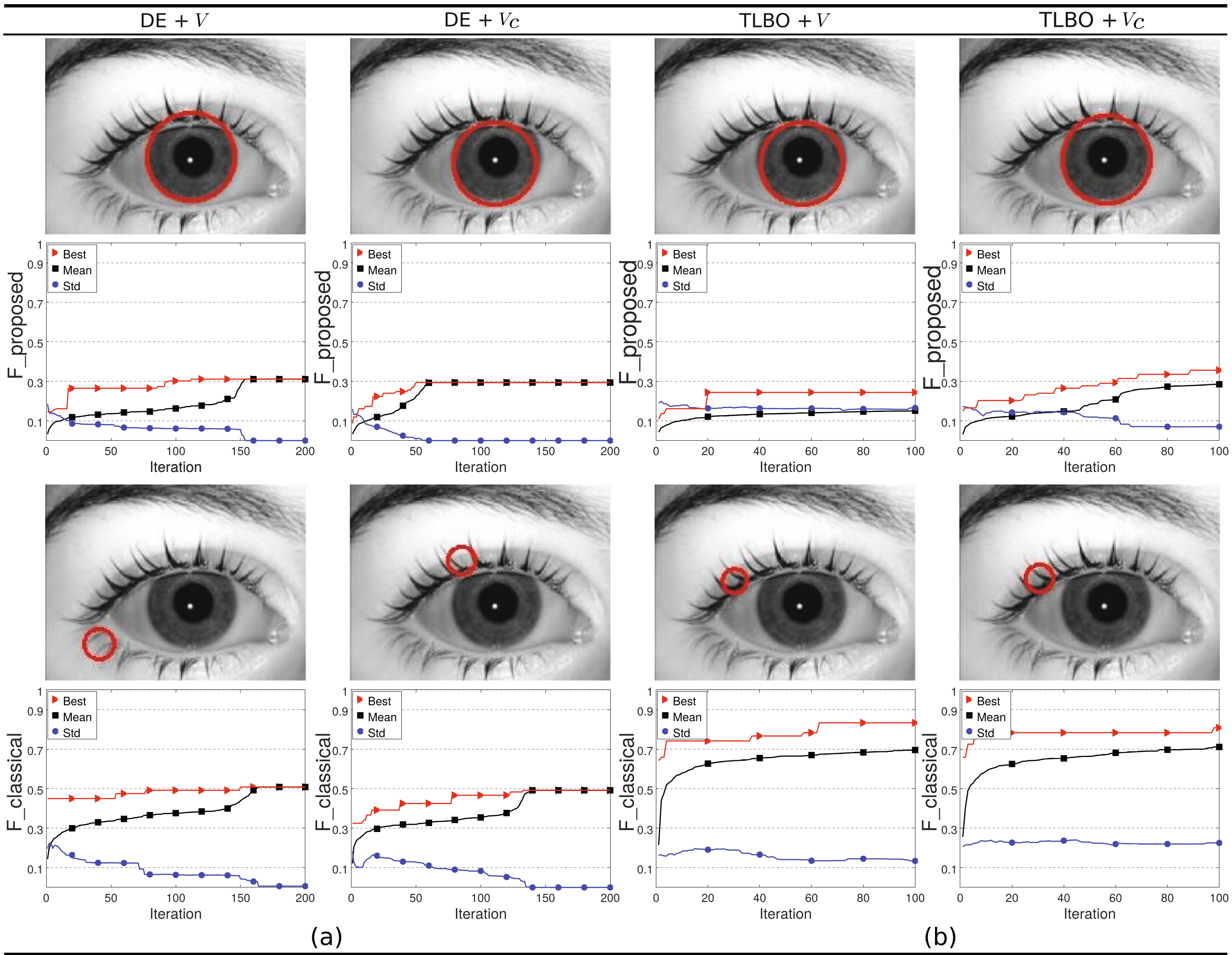

Fig. 7 Comparison between different combinations of meta-heuristic, search-space organization, and objective function. **a** Obtained results with DE optimizing over different search-space organizations and guided by different objective functions. **b** Results for the TLBO algorithm optimizing over different search-space organizations and guided by different objective functions

Within the convergence analysis, the TLBO and DE algorithms optimize over two different representations of the search space: 1) Edgels locations are indexed in a set V one after the other as they are visited within the edge map from left to right and top to bottom; and 2) Edgels locations are indexed in a set V_c with linked contours as proposed in Section 3.1. Further, to guide the search, two different objective functions are used: 1) The classical objective function $F_{clasical}$ that only considers edgel locations; and 2) The objective function $F_{proposed}$ as described in Section 3.3.

Results demonstrate that the proposed combination of an objective function that considers gradient orientation, an adequate search space, and the TLBO algorithm, results in an efficient circle detector. First, we discuss the performance of the proposed objective function. Figure 7 shows the detected circles after one execution of each metaheuristic. From the figure, it is clear that when $F_{classical}$ is used, false positives are detected because they have high fitness value. Conversely, when guided by the proposed objective function, a true circle is detected.

To better explain why our proposed objective function guides more efficiently the search, Fig. 8 depicts that, oppositely to $F_{clasical}$, $F_{proposed}$ clearly differentiates good from bad solutions, since it scores better a true circle than a false positive circle. The last statement is true due to the fact that $F_{proposed}$ does not only verifies edgels existence in the perimeter of a candidate circumference, but gradient orientation as well.

By additionally organizing the search space into linked contours, the fitness of the final solution is better, and the TLBO tends to converge. The graphs in Fig. 7b show for the TLBO algorithm that, when optimizing over the unorganized search space V and guided by either of the two objective functions, the search is random, and in consequence the best solution does not evolve. This is true since the similarity of the population remained almost equal throughout the iterations. Conversely, it is only when using the proposed search space representation V_c along with the proposed objective function $F_{proposed}$, that a clear evolution and convergence of the population is seen. As a result, the final solution has a higher fitness because the detected circle fits better a true circle. In other words, more edgels with a valid orientation pattern are truly present within the perimeter of the detected circle. Another advantage of the proposed search space representation is that it permits to obtain accuracy while keeping a simple coding.

It is clear that optimizing over V_c and guided by $F_{proposed}$, good accuracy is obtained. Furthermore, by also utilizing the TLBO algorithm, results are improved in comparison with the DE approach. We discuss the advantages of the TLBO algorithm by analyzing its exploration and exploitation behavior. In comparison with DE, TLBO explores more the search space as shown by the similarity of the populations in Fig. 7. The DE approach usually converges faster (the similarity becomes zero); this means that the DE algorithm stops searching. Differently, the TLBO algorithm makes more exploration of the search space, since there

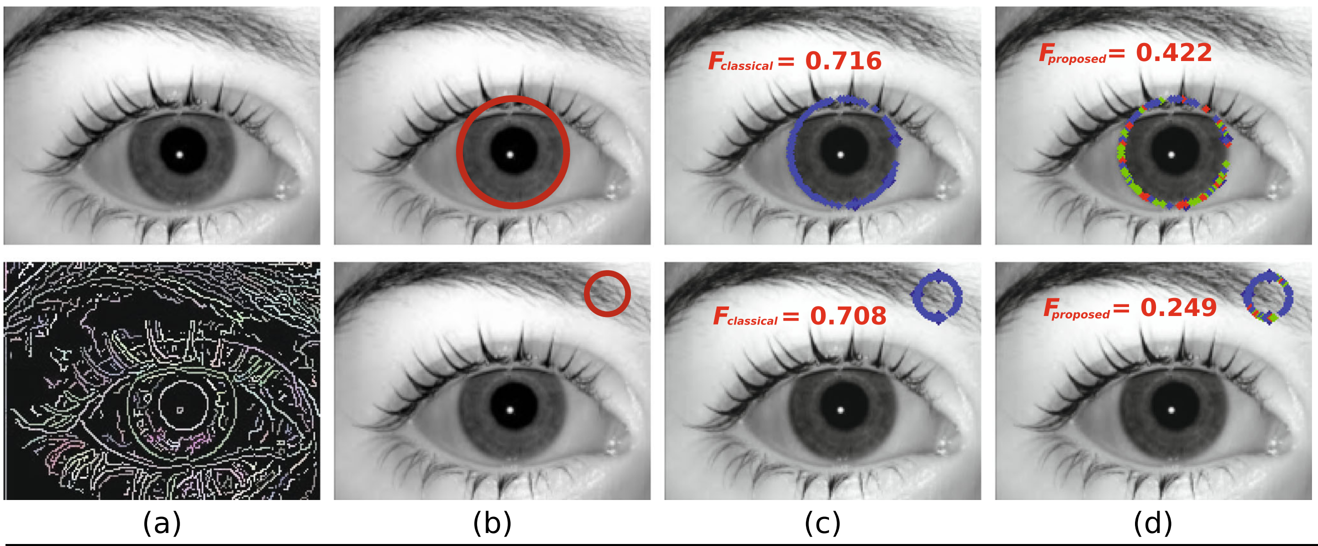

Fig. 8 Comparison between objective functions. **a** Test image and linked contours. **b** Two different candidate solutions are to be scored. **c** Fitness values computed with the classical objective function. **d** Fitness values obtained with the proposed objective function. $F_{proposed}$ truly differentiates good from bad solutions by additionally evaluating orientation alignment (θ-aligned, green; γ-aligned, red; and not aligned, blue), and not only the existence of edgels within the edge map like $F_{classical}$

are more variability within the population while keeping the best solution. By allowing more iterations to DE, no changes in the results would exist. To change the exploration and exploitation capabilities of the DE approach, a different setting of parameters is needed. Conversely, the TLBO algorithm does not require the tunning of algorithm-specific parameters to modify its convergence behavior.

4.3 Image performance

To evaluate the qualitative performance of the TLBO-based detector, several tests were carried out regarding different tasks involving:

1. Circle localization.
2. Multiple circle detection.
3. Blurred and low contrast circle detection.
4. Circular approximation.
5. Circle detection in presence of Gaussian noise.

The relevance of such tasks comes from the fact that they are commonly found in typical computer vision applications.

Results shown in Figs. 9 and 10 demonstrate that the proposed circle detector achieves efficiently every task. Our circle detector found all circles present in the test images despite cluttered textures, blurred and low-contrast circles, occluded circles, and Gaussian noise.

To test the capabilities of the proposed method for the first task, image Streetlight is used. For this task, the TLBO-based detector outperforms the OpenCV implementation and EDCircles. The OpenCV implementation does not fit accurately the circle present in the image; while EDCircles falsely detects a circular shape in the background.

Our method also detects multiple and concentric circles, as shown by the results of the second task. Images Plates, Bowling and Gobang show a better performance of our method in comparison with the Opencv implementation and EDCircles. For instance, EDCircles detects reflections as circles in Bowling and Gobang; while the Opencv implementation

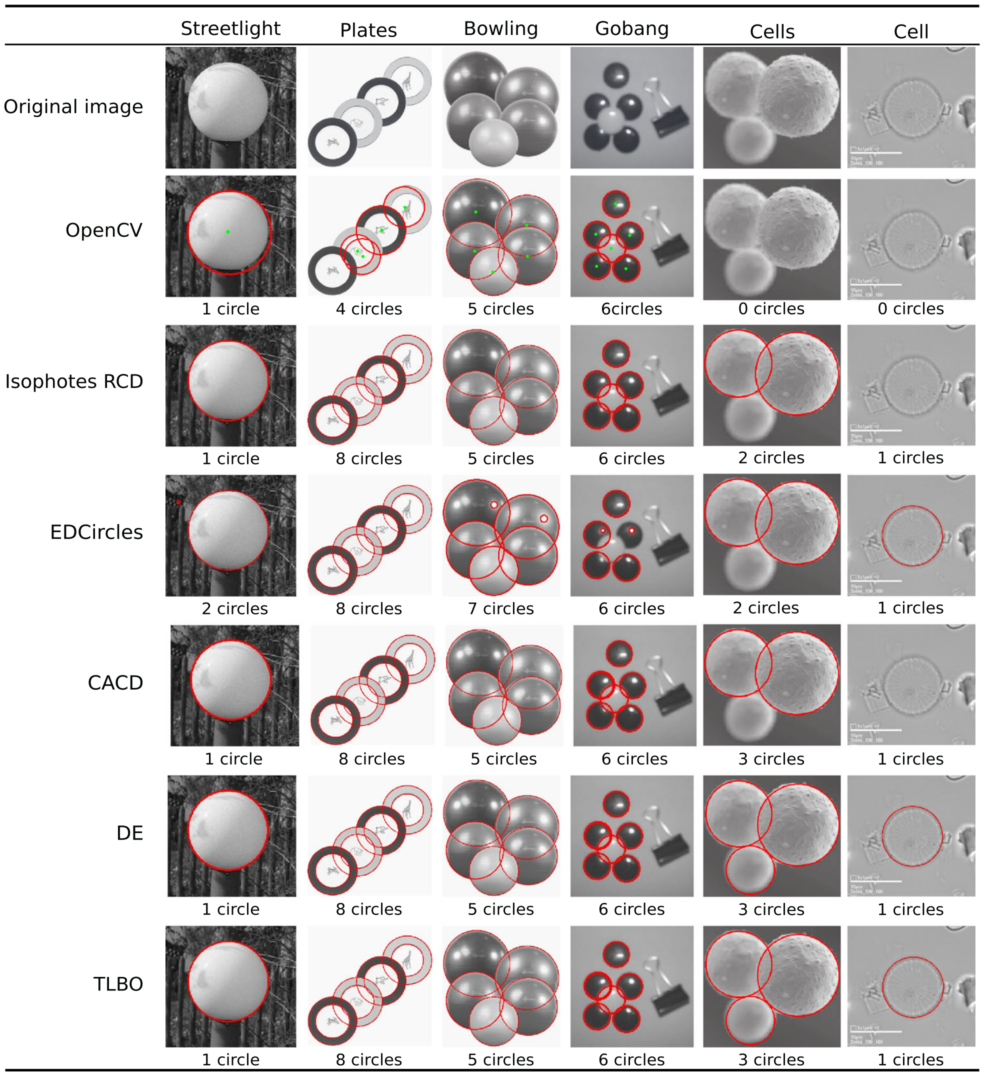

Fig. 9 Performance of the six methods for the tasks of circle localization with cluttered textures (column 1), multiple-circle detection (columns 2–4), and blurred or low-contrast circle detection (columns 5–6). The number of detected circles (including false positives) for the best case is indicated

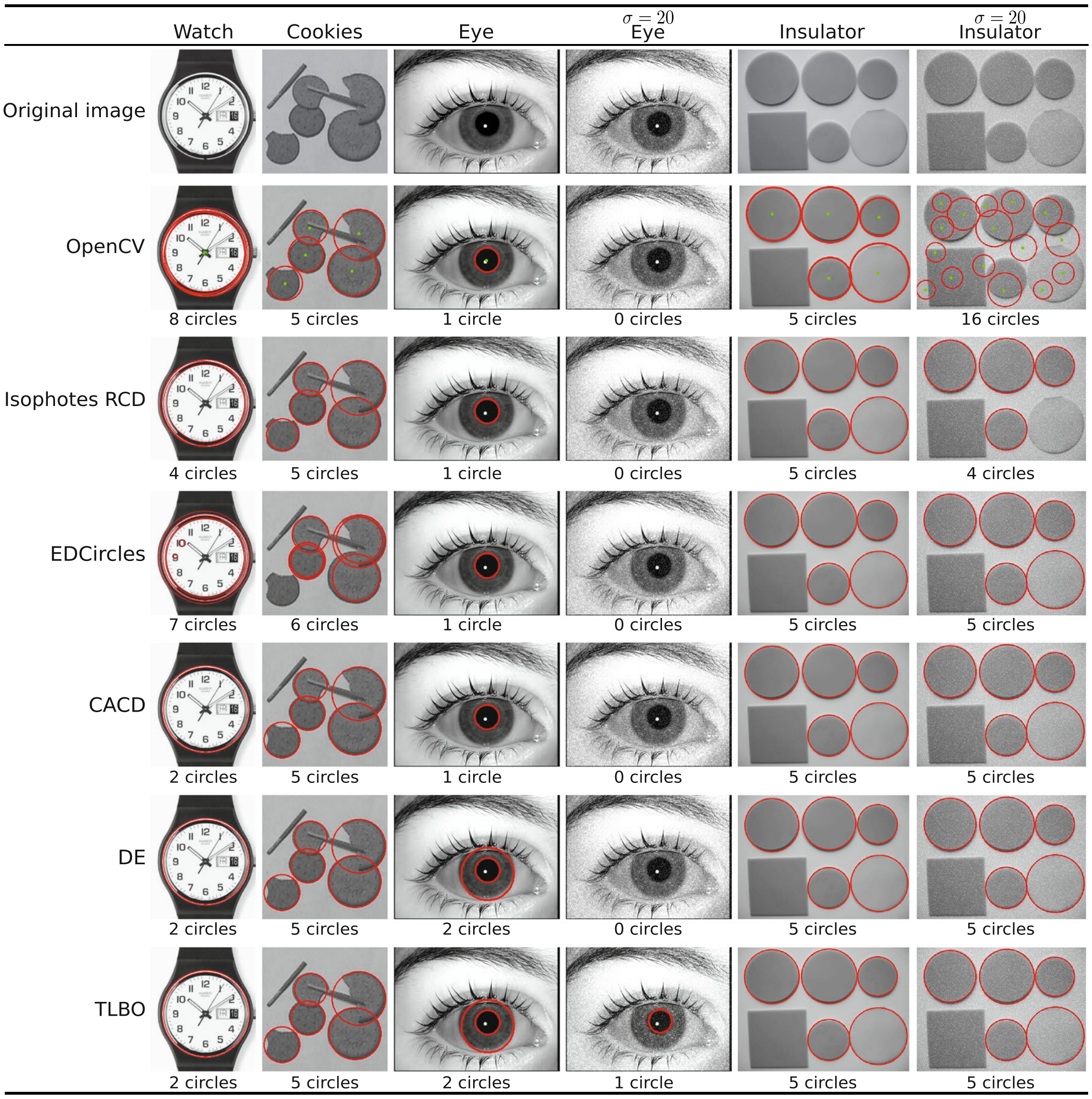

Fig. 10 Performance of the six methods in regard to the tasks of circular approximation (columns 1–2), and circle detection under Gaussian noise (column 3–6). The number of detected circles (including false positives) for the best case is indicated

does not detect all circles in Plates. Regarding the task of dealing with blurred and low-contrast circle detection, images Cells and Cell are presented. For this task, the TLBO-based detector outperforms the the Opencv implementation, and the CACD algorithm.

For the rest of the tasks, Fig. 10 shows the results involving circular approximation and noisy images. In comparison with other approaches, the proposed detector obtains better results. For instance, the proposed detector outperforms EDCircles in images Watch, Cookies and Eye. EDCircles tends to classify as circles small curves like the number “0” and “9” in image Watch. On the other hand, the Isophotes RCD approach is outperformed by our method in images Eye and Insulator with Gaussian noise.

It is noted that the DE algorithm has a similar performance to the TLBO for the all images, except the image Eye. This is due to the fact that they share the search space and objective function. However, quantitative results described in the next subsection show that the TLBO algorithm obtained better quantitative accuracy.

4.4 Performance evaluation

To obtain quantitative information of the accuracy of all detectors, a score metric is used to quantify their performance. Results from the methods are compared to a manually detected ground-truth circle using this score metric. Further, to compute a statistical analysis, the meta-heuristic-based

Table 3 Average time, average E_o error, and success rate for the meta-heuristic methods, considering the test images in Figs. 9 and 10

Image	DE			TLBO		
	Time	E_o eror	Succes rate	Time	E_o error	Succes rate
Streetlight	0.420	0.107	**100**	**0.396**	**0.088**	**100**
Plates	**1.260**	**0.073**	95.87	1.589	0.084	**96.55**
Bowling	0.527	**0.173**	95.32	**0.423**	0.179	**97.80**
Gobang	**0.443**	0.147	90.33	0.557	**0.062**	**92.35**
Cells	**0.794**	0.182	78.33	0.812	**0.175**	**72.66**
Cell	0.359	0.112	59	**0.318**	**0.098**	**65**
Watch	0.540	**0.091**	**97.50**	**0.417**	0.152	96.20
Cookies	0.473	0.057	95.60	**0.460**	**0.032**	**98.33**
Eye	0.276	0.191	76	**0.250**	**0.105**	**89.50**
Eye ($\sigma = 20$)	0.352	–	–	**0.367**	**0.112**	**43**
Insulator	0.808	0.135	98.66	**0.787**	**0.124**	**100**
Insulator($\sigma = 20$)	0.887	**0.187**	90.20	**0.792**	0.193	**92.35**

Bold values indicate the best result for a particular experiment

circle detectors were executed 100 times for each test image.

The metric to evaluate a quantitative performance looks at the percent overlap of the circle area. The overlap error E_o is calculated as:

$$E_o = 1 - \overline{\frac{C_1 \cap C_2}{C_1 \cup C_2}}, \tag{18}$$

where the second term is the overlap percentage, and is the area of the overlap between circle 1, C_1 and circle 2, C_2, divided by the area of the union of the circles. For multiple circle evaluation, the average value of the E_o errors, M_{E_o}, is calculated by:

$$M_{E_o} = \left(\frac{1}{NC}\right) \cdot \sum_{i=1}^{NC} E_{o_i}, \tag{19}$$

where NC is the total number of detected circles in the test image.

For the meta-heuristic approaches, we also compute a success rate S_r as now explained. Given an specific image, let C_{GT} be the set of ground truth circles, and C_D the set of circles detected by the method. For each circle in C_D, we find the match circle in C_{GT} subject to $E_o < 0.20$. Only one circle in C_D is accepted as a valid match per ground truth circle. Then, all the accepted circles are considered as true positives and stored in a set C_{TP}. Finally, we compute the success rate S_r as follows:

$$S_r = \frac{card(C_{TP})}{max(card(C_{GT}), card(C_D))}, \tag{20}$$

where $card(\cdot)$ returns the cardinality of a set. It is noted that S_r is equal to 1 when all the ground truth circles are

Table 4 Time and E_o error for the non-meta-heuristic methods, considering the test images in Figs. 9 and 10

	OpenCV		Isophotes RCD		EDCircles	CACD	
Image	Time	E_o error	Time	E_o error	E_o error	Time	E_o error
Streetlight	0.368	0.086	0.625	0.083	0.052	1.907	0.031
Plates	0.927	0.092	2.837	0.076	0.083	3.015	0.068
Bowling	1.017	0.318	1.605	0.152	0.065	2.216	0.105
Gobang	0.757	0.235	1.092	0.198	0.071	2.291	0.117
Cells	1.158	–	1.027	0.106	0.073	2.972	0.087
Cell	0.709	–	0.968	–	0.026	2.397	–
Watch	1.415	0.173	1.072	0.194	0.091	2.737	0.112
Cookies	1.157	0.025	2.142	0.177	0.124	2.471	0.089
Eye	0.831	0.104	1.793	0.052	0.087	1.916	0.062
Eye ($\sigma = 20$)	1.751	–	2.611	–	–	3.041	–
Insulator	0.872	0.016	1.595	0.069	0.098	2.795	0.092
Insulator($\sigma = 20$)	12.365	–	2.407	0.098	0.019	3.973	0.091

found and no false positives are detected. Differently, when all ground truth circles are found, but false positives are detected, S_r decreases. The more false positives are detected or the less ground truth circles are found, the more S_r decreases and approximates to 0. Also, S_r is equal to 0 when there are no found circles by the method that match the ground truth circles. S_r penalizes even multiple detections of circles with similar centers and radii. The succes rate S_r allows us to obtain a measure of the performance of the algorithm to detect all ground truth circles and discard false positives. The error E_o, on the other hand, when applied only to circles in C_{TP}, allow us to obtain a measure of the accuracy of the detected true-positives circles.

Quantitative results are shown in Tables 3 and 4. Table 3 exposes the average time, average error, and success rate for the meta-heuristic approaches. Table 4, on its part, shows the time and error for the non-meta-heuristic approaches. In the case of EDCircles, computational time is omitted since the on-line demo provided by the authors was used.

5 Conclusions

In this paper, we presented a meta-heuristic circle detector based on the TLBO algorithm. The proposed method differs from previous meta-heuristic approaches in the way that it uses additional information provided by the gradient to customize the search space, and to better guide the search with a novel objective function.

In comparison with other approaches, the results obtained by the TLBO-based detector required less effort from the user because there are less parameters to tune for this particular meta-heuristic. This is an advantage in comparison to other meta-heuristic approaches, and even to the OpenCV CHT and the isophotes RCD algorithms that also require the set of various parameters. To test the performance of the proposed approach, computation time and accuracy have been compared with five different approaches based on different principles: The OpenCV CHT, the isophotes RCD algorithm, EDCircles, CACD, and a meta-heuristic based on DE. We used a score metric to evaluate the mismatch between a ground-truth circle and the detected circle. Also, since our proposed circle detector is based on a meta-heuristic, an evaluation of the convergence of the algorithm was performed.

Our approach accurately detects circles in real images despite the presence of cluttered textures, circle occlusions, blurred effects, and Gaussian noise. Further, the detected circles hold a sub-pixel accuracy due to the use of the circle equation and the MCA method.

In future work, we will make efforts to expand our method to detect ellipses and reduce the computation time; and also to explore the use of gradient magnitude, and not only orientation in the objective function, since usually pixels of the same circumference have similar magnitude values.

References

1. Leo M, De Marco T, Distante C (2014) Highly usable and accurate iris segmentation. In: 2014 22nd international conference on pattern recognition (ICPR). IEEE, pp 2489–2494
2. Ngo HT, Rakvic RN, Broussard RP, Ives RW (2014) Resource-aware architecture design and implementation of hough transform for a real-time iris boundary detection system. IEEE Trans Consum Electron 60(3):485–492
3. Wen Z, Wang Y, Luo J, Kuijper A, Di N, Jin M (2017) Robust, fast and accurate vision-based localization of a cooperative target used for space robotic arm. Acta Astronaut 136:101–114
4. Saska M, Baca T, Thomas J, Chudoba J, Preucil L, Krajnik T, Faigl J, Loianno G, Kumar V (2017) System for deployment of groups of unmanned micro aerial vehicles in gps-denied environments using onboard visual relative localization. Auton Robot 41(4):919–944
5. Berkaya SK, Gunduz H, Ozsen O, Akinlar C, Gunal S (2016) On circular traffic sign detection and recognition. Expert Syst Appl 48:67–75
6. Scholz S, Mueller T, Plasch M, Limbeck H, Adamietz R, Iseringhausen T, Kimmig D, Dickerhof M, Woegerer C (2016) A modular flexible scalable and reconfigurable system for manufacturing of microsystems based on additive manufacturing and e-printing. Robot Comput Integr Manuf 40:14–23
7. da Fontoura Costa L, Cesar RM Jr (2010) Shape analysis and classification: theory and practice. CRC Press, Boca Raton
8. Ok AO, Başeski E (2015) Circular oil tank detection from panchromatic satellite images: a new automated approach. IEEE Geosci Remote Sens Lett 12(6):1347–1351
9. Li C, Huo H, Fang T (2016) Oil depots detection from high resolution remote sensing images based on salient region extraction. In: 2016 international conference on audio, language and image processing (ICALIP). IEEE, pp 285–288
10. Yadav VK, Trivedi MC, Rajput SS, Batham S (2016) Approach to accurate circle detection: multithreaded implementation of modified circular hough transform. In: Proceedings of international conference on ICT for sustainable development. Springer, Berlin, pp 25–34
11. Kumar V, Asati A, Gupta A (2018) Memory-efficient architecture of circle hough transform and its FPGA implementation for iris localization. IET Image Process 12(10):1753–1761
12. Yao Z, Yi W (2016) Curvature aided hough transform for circle detection. Expert Syst Appl 51:26–33
13. Manzanera A, Nguyen TP, Xu X (2016) Line and circle detection using dense one-to-one hough transforms on greyscale images. EURASIP J Image Video Process 2016(1):46
14. Jia L-Q, Peng C-Z, Liu H-M, Wang Z-H (2011) A fast randomized circle detection algorithm. In: 2011 4th international congress on image and signal processing (CISP), vol 2. IEEE, pp 820–823
15. Yu H, Wang T (2017) Vision-based technique for circle detection and measurement using lookup table and bitwise center accumulator. JOSA A 34(3):415–423
16. Gonzalez R (2015) Fast line and circle detection using inverted gradient hash maps. In: 2015 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, pp 1354–1358
17. Zhang H, Wiklund K, Andersson M (2016) A fast and robust circle detection method using isosceles triangles sampling. Pattern Recognit 54:218–228

18. Chung K-L, Huang Y-H, Shen S-M, Krylov AS, Yurin DV, Semeikina EV (2012) Efficient sampling strategy and refinement strategy for randomized circle detection. Pattern Recognit 45(1):252–263
19. De Marco T, Cazzato D, Leo M, Distante C (2015) Randomized circle detection with isophotes curvature analysis. Pattern Recognit 48(2):411–421
20. Fornaciari M, Prati A, Cucchiara R (2014) A fast and effective ellipse detector for embedded vision applications. Pattern Recognit 47(11):3693–3708
21. Li Y, Zhao C (2015) Fast ellipse detection by elliptical arcs extracting and grouping. In: Sixth international conference on graphic and image processing (ICGIP 2014), vol 9443. International Society for Optics and Photonics, p 94430C
22. Akinlar C, Topal C (2013) Edcircles: a real-time circle detector with a false detection control. Pattern Recognit 46(3):725–740
23. Cuevas E, Zaldivar D, Pérez-Cisneros M, Ramírez-Ortegón M (2011) Circle detection using discrete differential evolution optimization. Pattern Anal Appl 14(1):93–107
24. Oliva D, Cuevas E (2017) Detection of circular shapes in digital images. In: Advances and applications of optimised algorithms in image processing. Springer, pp 113–134
25. Cuevas E, Osuna V, Oliva D (2017) Multi-circle detection on images. In: Evolutionary computation techniques: a comparative perspective. Springer, pp 35–64
26. Cuevas E, González M (2013) Multi-circle detection on images inspired by collective animal behavior. Appl Intell 39(1):101–120
27. Fourie J (2017) Robust circle detection using harmony search. J Optim 2017. https://doi.org/10.1155/2017/9710719
28. Díaz-Cortés M-A, Cuevas E, Rojas R (2017) Clonal selection algorithm applied to circle detection. In: Engineering applications of soft computing. Springer, pp 143–164
29. Rao RV, Savsani VJ, Vakharia DP (2011) Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
30. Rao R (2016) Review of applications of tlbo algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems. Decis Sci Lett 5(1):1–30
31. Goyal RK, Kaushal S (2016) A constrained non-linear optimization model for fuzzy pairwise comparison matrices using teaching learning based optimization. Appl Intell 45(3):652–661
32. El Ghazi A, Ahiod B (2018) Energy efficient teaching-learning-based optimization for the discrete routing problem in wireless sensor networks. Appl Intell 48(9):2755–2769
33. Cuevas E, Wario F, Osuna-Enciso V, Zaldivar D, Pérez-Cisneros M (2012) Fast algorithm for multiple-circle detection on images using learning automata. IET Image Process 6(8):1124–1135
34. López-Martinez A, Cuevas FJ (2018) Automatic multi-circle detection on images using the teaching learning based optimization algorithm. IET Comput Vis 12(8):1188–1199
35. Davies ER (1987) The effect of noise on edge orientation computations. Pattern Recognit Lett 6(5):315–322
36. Kittler J (1983) On the accuracy of the sobel edge detector. Image Vis Comput 1(1):37–42
37. Suzuki S et al (1985) Topological structural analysis of digitized binary images by border following. Comput Vis Graph Image Process 30(1):32–46
38. Van Aken JR (1984) An efficient ellipse-drawing algorithm. IEEE Comput Graph Appl 4(9):24–35

A. Lopez-Martinez received a degree in Mechatronics in 2012 from the Universidad Tecnológica, campus León, México, and a M. Sc. Eng. Degree in Optomechatronics in 2014 from Centro de Investigaciones en Óptica, A.C., campus León, México. He is currently a Ph.D. candidate in Centro de Investigaciones en Óptica, A.C., campus León México. His research interests are mainly focused on robotics, computer vision, and artificial intelligence.

F. J. Cuevas received his BS degree in Computer Systems Engineering from Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) Campus Monterrey in 1984. He obtained his Master degree in Computer Science from ITESM Campus Edo. de México in 1995 and his PhD in Optics Metrology from Centro de Investigaciones en Óptica, A.C. in 2000, where he is currently a researcher. He made a posdoctoral stay in Centro de Investigación en Computacion of Instituto Politécnico Nacional of México, between 2001 and 2002. He has more than 40 publications in international journals with rigorous refereeing and more than 90 works in national and international conferences. He is a member of the Sistema Nacional de Investigación of México since 1995. He was the leader of the Optical Metrology Group of the Centro de Investigaciones en Óptica, A.C. from 2003 to 2006. From 2006 to 2013, he was the Academic Director of Centro de Investigaciones en Óptica, A.C. His research lines cover: Optical Metrology, Computer Vision, Digital Image Processing, Pattern recognition, Neural networks and Genetic Algorithms.