

Automatic multi-circle detection on images using the teaching learning based optimisation algorithm

Alan López¹✉, Francisco J. Cuevas¹

¹Optical Metrology, Centro de Investigaciones en Óptica. A.C., Lomas del Bosque 115, León, Guanajuato, México

✉ E-mail: alan.lopez@cio.mx

Abstract: Circle detection has numerous applications towards industry, robotics, and science in general. Therefore, a significant effort has been made in order to develop an accurate and fast method for circle extraction. Commonly, different techniques such as the ones based on the Hough transform have been widely used because of their robustness. However, these techniques usually demand a considerable computational load and large storage, and therefore meta-heuristic approaches such as evolutionary and swarm-based algorithms have been studied as an alternative. This study introduces a circle-detection method based on a recently proposed meta-heuristic technique: the teaching learning based optimisation algorithm, which is a population-based technique that is inspired by the teaching and learning processes. The algorithm uses the encoding of three points as candidate circles over the edge image. To evaluate if such candidate circles are actually present within the edge map, an objective function is used to guide the search. To validate the efficacy of the proposed approach, several tests using noisy and complex images as input were carried out, and the results were compared with different approaches for circle detection.

1 Introduction

Regarding pattern recognition and computer vision, circle detection is of utmost practical importance since its many applications in biometrics, industrial inspection, space science, autonomous navigation among others. Examples of applications for circle detection within these areas include and are not limited to traffic sign recognition [1], iris detection [2, 3], automatic industrial inspection [4], visual robot orientation [5], and on-orbit service technologies [6]. Given the relevance to this problem, different techniques for circle detection have been extensively investigated to offer an adequate solution. State-of-the-art techniques for circle extraction can be classified into three categories: Hough transform (HT) based techniques, arc-segment-based techniques, and meta-heuristic techniques. One of the foremost differences between these categories is the computational efficiency concerning processing time and memory usage. This is due to the approach these techniques used to explore the search space.

Circle detectors based on the HT explore the search space in a non-probabilistic or probabilistic manner. The classic HT is an example of a non-probabilistic and exhaustive technique. This method aims to find geometrical entities, such as circles, in a binary image by mapping every point in the input image into a parametric space, thus converting the difficult pattern detecting in the image space into local peak searching in the parameter space. This approach makes the HT robust to noise, yet computationally inefficient. In order to surmount this problem, some new approaches based on the HT have been proposed to improve both storage and computational requirements. For example, the adaptive HT [7] reduces the storage requirement by decomposing the circle finding problem into two stages. First, it uses a two-dimensional array to accumulate votes along the normal of each edge point to propose candidate centres for circles. Then, to identify the radius, the distance of each point from a candidate centre is calculated and a radius histogram is produced. The adaptive HT is also considered as a non-probabilistic technique since it considers all points within the edge map. Other examples of non-probabilistic techniques based on the HT are the fast HT [8] and the combinatorial HT [9]. On the other hand, the probabilistic HT-based techniques use random sampling for selecting only a small subset of the data, which makes the search less exhaustive. Among the probabilistic

techniques based on the HT are the Monte Carlo HT [10], the probabilistic HT [11], and the randomised HT [12].

More recently, an advanced variant of the randomised HT, called randomised circle detection (RCD), has been proposed in [13]. This method randomly selects four non-collinear pixels in the edge map. Three of them are used to compute a circle, and the last one determines whether the circle is a possible circle, then a voting procedure is performed to decide whether possible circles are to be upgraded to detected circles. To improve the efficiency and accuracy of the RCD, a novel sampling and refinement strategy for the RCD, called gradient-based RCD with refinement (GRCD-R), is proposed in [14]. This method additionally uses the gradient direction to validate if the four randomly selected pixels can compute a valid circle. Furthermore, to validate candidate circles, the GRCD-R algorithm uses the gradient direction to test whether the pixels of a possible circle point to its centre. There are some other recent approaches [15–18] that also exploit the gradient information along with other geometrical properties of circles in order to lower the computational and memory usage required for the circle detection task.

Other approaches do not use points to find circles as the aforementioned techniques. Instead, these approaches use arc segments to find circles and ellipses in binary images. The work in [19], for instance, proposed an algorithm for segmenting connected points into lines, circular, elliptical, and super-elliptical arcs. Other example of an arc-segment-based technique for circle extraction is the work in [20]. In this work, line segments are extracted from the edge map. Then, the line segments that are potential candidates of elliptic arcs are linked to form arc segments according to connectivity and curvature conditions. Finally, arcs are grouped together if they belong to the same ellipse.

Heuristics have been used in combination with arc-segment-based techniques in order to improve the search. For instance, a real-time circle detection algorithm named EDCircles [21] makes use of two heuristic algorithms to detect candidate circles and near-circular ellipses. EDCircles uses a contiguous set of edge segments produced by a parameter-free edge segment detector called edge drawing parameter free (EDPF). Then the detected segments are converted into line segments. After that, the detected line segments are converted into circular arcs, which are joined together using the heuristic algorithms to fit circles and ellipses. To eliminate false detections, EDCircles uses a contrario validation step based on the

Helmholtz principle. The methods described above are efficient, but sometimes they produce false detections and missing detections. Thence, meta-heuristic techniques have been studied as an alternative.

Evolutionary and swarm-based algorithms have been studied as a meta-heuristic manner to perform circle detection. Such nature-inspired optimisation algorithms work on the principles of different natural phenomena. Examples of meta-heuristic algorithms include the genetic algorithm (GA), differential evolution (DE), the particle swarm optimisation (PSO), the artificial bee colony (ABC), the learning automata (LA) algorithm, the harmony search (HS), and the grenade explosion method. The utilisation of meta-heuristic algorithms for circle detection has been reported, for example, using a GA [22], the PSO [23], an ABC [24], DE [25], the HS algorithm [26], and the LA algorithm [27]. For the aforementioned work, a set of parameters must be tuned appropriately to accomplish good accuracy and high detection rate in the circle detection task. This disadvantage becomes important when dealing with different image conditions, such as variations in illumination, low contrast, distortion, and blurring boundaries.

In the interest of improving the autonomy of the meta-heuristic-based circle detector, in this paper we propose the use of another natural phenomena inspired meta-heuristic optimiser, namely, the teaching learning based optimisation (TLBO) algorithm. The TLBO algorithm does not require the tuning of many parameters in comparison with other meta-heuristics. For instance, GA needs the tuning of the elite size, crossover rate, and mutation rate. The same is the case with PSO, which uses inertia weight, social and cognitive parameters. Similarly, ABC requires a proper tuning of parameters such as the number of bees (employed, scout, and onlookers), limit, and others. HS, on the other hand, requires the harmony memory consideration rate, pitch adjusting rate, and the number of improvisations. Conversely, for the TLBO only the number of iterations and the size of the population need to be provided to the algorithm.

The TLBO algorithm, first proposed by Rao *et al.* in [28], is inspired in the classical school learning process where students knowledge increases by two means: (i) the effects of teaching of a teacher upon a student and (ii) the knowledge transfer through the interaction of a student with her or his schoolmates. The TLBO algorithm is a successful heuristic method for solving complex optimisation problems, still keeping a simple structure and an easy implementation. Thence, several engineering and scientific applications have been published [28–32].

In this work, an automatic multi-circle detection on images is presented using the TLBO algorithm that, to our knowledge, has not been applied to the circle detection task. To demonstrate the performance of the proposed method, a comparison was performed with four different approaches based on the probabilistic and non-probabilistic HT, arc-segment, and meta-heuristic techniques.

The rest of the paper is organised as follows. Section 2 describes the TLBO algorithm as implemented for this work. Section 3 presents the use of the TLBO algorithm for circle detection. Section 4 shows the quantitative and qualitative experimental results. Finally, Section 5 states the conclusions and future work.

2 TLBO algorithm

The TLBO algorithm is a recently developed population-based heuristic algorithm that is inspired by the teaching and learning process. The principle of the algorithm is based on two main ideas: one, a student is able to improve its knowledge by learning from the teacher. Two, a student knowledge could be improved through the interaction with other students. In this algorithm, a group of students is considered as the population and the different subjects offered are compared to the problem variables. Similarly, the objective function value is compared with student grades. The most qualified student, i.e. the best solution, is viewed as the teacher.

The TLBO starts with an initialisation procedure, where N random numbers of initial solutions (students) $X_i \quad i \in \{1, \dots, N\}$ are generated within the search space. The algorithm continues with two consecutive phases: the teacher phase and the learner

phase. During the first phase, the teacher X_{teacher} (best solution) tries to increase the knowledge level of the whole class and to help students individually to get better grades. To achieve this, the algorithm attempts to shift the value of students in the group towards the teacher value. After the phase consisting in learning from the teacher is completed, the second phase is performed. This final phase attempts knowledge increase through the interaction between students. During this phase any student can interact with any other student for knowledge transfer. The algorithm is terminated after a certain number of iterations are completed.

As explained above, the element X_i of the population represents a single possible solution to a particular optimisation problem. X_i 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 within the TLBO context. The algorithm then tries to improve certain individuals by changing these individuals during teacher and learner phase.

During the teacher phase, the best individual is assigned as the 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

$$X_{\text{new}} = X_i + r(X_{\text{teacher}} - \text{TF}X_{\text{mean}}) \quad (1)$$

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. In (1), 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. X_i is replaced by X_{new} if the latter gives better fitness value.

As stated in [28], the intuition behind the teaching factor is that in practice a teacher can only improve the quality of a class up to some extent depending on the capability of the class. Thus, the teaching factor is decided randomly for heuristic purposes, and can be either 1 or 2, thus emphasising the importance of student quality. Also, in (1), r and TF contribute to the exploration capabilities of the algorithm.

The teacher phase aims to increase the mean of the class through the teacher, who put maximum effort into teaching its students, but students gain knowledge according to the quality of the imparted teaching. However, the possibility for students to improve their knowledge is not completely lost. During the second and final phase, the learner phase, a student learns with the help of other students. In general terms, the quantity of knowledge transferred to a student does not only depend on its teacher but also on interactions amongst students.

During the learner phase, 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

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

Otherwise, it is moved away from X_j according to

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

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 process above, involving the two phases, is repeated until a certain termination criterion is met. Fig. 1 describes the simplest form of the TLBO algorithm. The proposed method adopts this process for circle extraction as explained in the following section.

3 Circle detection using the TLBO algorithm

The proposed approach seeks different objectives: (i) to detect multiple small and large circles; (ii) to work with real and noisy images; (iii) to have high detection rate and good accuracy in

```

1 Algorithm TLBO
2 Initialize the population of 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    $r \leftarrow \text{rand}(0,1)$ 
8    $TF \leftarrow \text{round}(1+\text{rand}(0,1))$ 
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      end
15    Start learner phase:
16      Select randomly another student  $X_j$  with  $i \neq j$ 
17      if ( $F(X_i) > F(X_j)$ ) then
18         $X_{new} \leftarrow X_i + r \cdot (X_i - X_j)$ 
19      else
20         $X_{new} \leftarrow X_i + r \cdot (X_j - X_i)$ 
21      end
22      if ( $F(X_{new}) > F(X_i)$ ) then
23         $X_i \leftarrow X_{new}$ 
24      end
25    end
26 end
27 Find out the best solution: ( $X_{teacher}$ ).
28 Select  $X_{teacher}$  as the final solution.

```

Fig. 1 TLBO algorithm

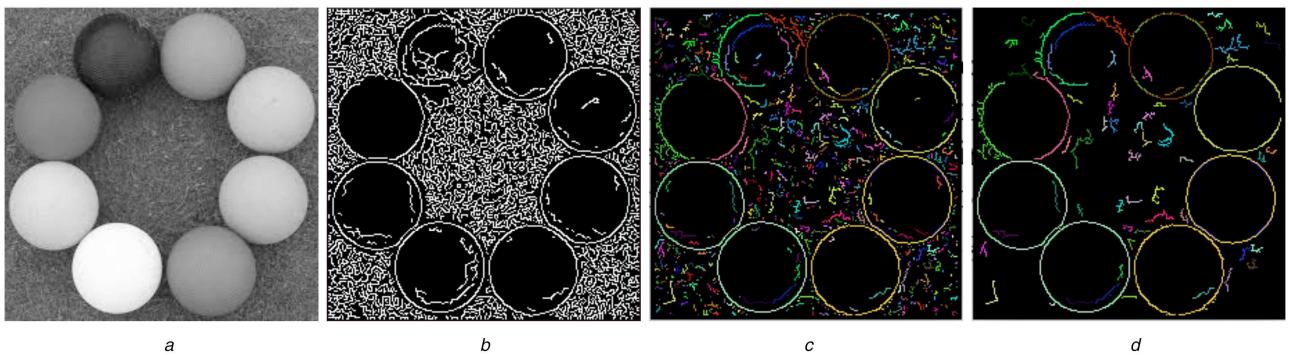


Fig. 2 Preprocessing steps for the proposed approach

(a) Input image for the algorithm is a grey-scale image, (b) First preprocessing step is noise filtering followed by edge extraction using the Canny algorithm, (c) Pixels within the edge map are connected into contours, (d) Contours having size below a predefined threshold are not considered to be part of the search space, and are removed. Thus, the input information to the proposed method is organised into contours large enough to be part of a circle

images with low contrast, distortion, and blurring boundaries; and (iv) to produce a few or no false detections.

In order to attain the aforementioned objectives, the TLBO algorithm is used for circle detection. Additionally, the input data for the algorithm is preprocessed to obtain better results. In this section, we first describe the image preprocessing, and then, we define the elements of the TLBO implementation for circle detection.

3.1 Preprocessing

The main purpose of the preprocessing step is to enhance the search space for the TLBO algorithm. This improvement is achieved by reducing the size of the search space and by customising its organisation.

Concerning the search space organisation, in previous works [22–25], the search space is organised as a set containing all edgels (edge pixel) locations. The edgels locations are indexed in the set one after the other as the edgels are visited within the edge image from left to right and top to bottom. The work in [27], on the other hand, store edgels locations randomly within the set. These particular organisations, as remarked in [26], do not ensure that nearby pixels within the edge image are neighbours in the search space. This is an important issue for our method since the TLBO

algorithm locates better individuals in each iteration by shifting them towards the position of the best one. This means that neighbouring individuals within the search space are expected to belong to the same circle and have similar fitness value. Based on these considerations, we preprocess the input image differently in order to reduce the size of the search space and customise its organisation.

The enhancement of the search space is attained by the following steps. First, to reduce the image noise, a 3×3 Gaussian smoothing filter is applied to the input grey-scale image. After which, the Canny operator is employed to extract the edge image I . Finally, the pixels in I are connected into contours using the border-following algorithm outlined in [33] that constructs the adjacency tree of the image. Then, all the contours that are considered too small (i.e. below a threshold T_c) 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:

$$T_c = \text{ceiling}(2\pi r_{min}). \quad (4)$$

Finally, the search space is organised in a set V_c with edgel locations stored together according to their respective contour c . Hence, the search space is stored together and ordered in the set

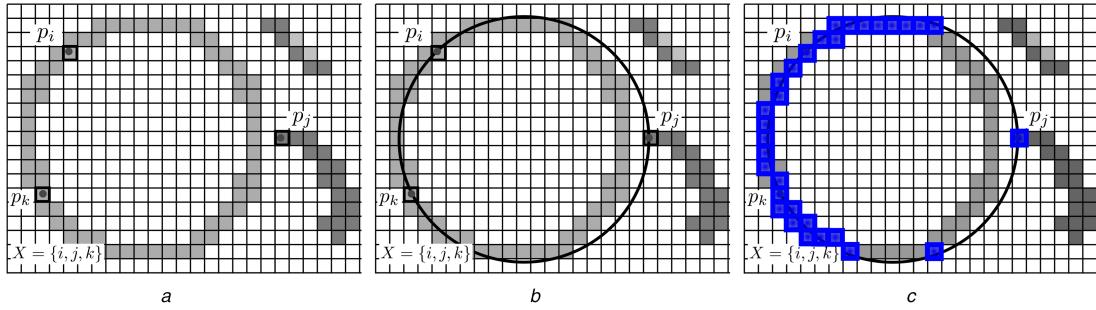


Fig. 3 Objective FE

(a) Student under test $X = \{i, j, k\}$ encodes the pixel positions $p_i(x_i, y_i)$, $p_j(x_j, y_j)$, $p_k(x_k, y_k)$ within the edge map, (b) Candidate circle passing by the three points is proposed, (c) All the pixels (highlighted with bigger blue squares) that exists in both the edge map and the perimeter of the candidate circle contribute to the fitness value

$V_c = \{c_1, c_2, \dots, c_n\}$ with n the number of linked contours, and $c_i = \{p_{i1}(x_{i1}, y_{i1}), \dots, p_{im_i}(x_{im_i}, y_{im_i})\}$ with m_i the number of pixels connected in the i -contour. One result of this organisation is an increase of the probability that nearby pixels in the search space belong to the same circle since connected pixels are stored together. Fig. 2 summarises the operations of the preprocessing step in the proposed approach.

3.2 Individual representation

Each student X of the population encodes a candidate circle C that passes through three different edgels $p_i(x_i, y_i)$, $p_j(x_j, y_j)$, and $p_k(x_k, y_k)$. To construct student X , three different elements are randomly chosen from V_c . Thus, $X = \{i, j, k\}$ 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 image, respectively.

From the student X , the parameters of the candidate circle C , i.e. its centre (x_0, y_0) and radius r , are calculated by the following equations:

$$x_0 = \frac{x_j^2 + y_j^2 - (x_i^2 + y_i^2) \quad 2(y_j - y_i)}{4((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i))}, \quad (5)$$

$$y_0 = \frac{2(x_j - x_i) \quad x_j^2 + y_j^2 - (x_i^2 + y_i^2)}{4((x_j - x_i)(y_k - y_i) - (x_k - x_i)(y_j - y_i))}, \quad (6)$$

and

$$r = \sqrt{(x_0 - x_d)^2 + (y_0 - y_d)^2}, \quad (7)$$

where $d \in \{i, j, k\}$. Hence, it is possible to represent a candidate circle $C(x_0, y_0, r)$ as a transformation T of the edge vector indexes i , j , and k belonging to the student $X = \{i, j, k\}$. With T being the transformation calculated using x_0 , y_0 , and r from 4–6. Each student is graded according to a fitness function as explained as follows.

3.3 Objective function

A well-fitted student is the one that encodes a candidate circle that actually exists within the edge image. In order to evaluate that, our proposed approach for circle detection proceeds as follows. Given a candidate circle, N_c test points within the perimeter of the candidate circle are generated. The value of the fitness function is the ratio of the total number of test points appearing in the edge image to the total number of test points

$$f(X) = \frac{\sum_{i=1}^{N_c} E(x_i, y_i)}{N_c}, \quad (8)$$

where N_c representing the number of test points in the perimeter of the circle corresponding to the student X under test, and $E(x_i, y_i)$ being the function that evaluates whether the edgel $p_i(x_i, y_i)$ exists within the edge map. Hence, the function $E(x_i, y_i)$ is defined as

$$E(x_i, y_i) = \begin{cases} 1 & \text{if } p_i(x_i, y_i) \text{ is an edgel} \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

For the objective function in (8), a value near to one implies a better grade for the student under test. A graphic example concerning the evaluation of the objective function is depicted in Fig. 3. Given a student $X = \{i, j, k\}$, a candidate circle (x_0, y_0, r) is proposed. Then, in the interest of reducing the computational burden and improving the subpixel accuracy, the set of test points to evaluate the objective function is generated using the midpoint circle algorithm (MCA) [34] as described below.

3.4 Midpoint circle algorithm

The MCA is an established algorithm used to determine the required points to draw a circle in a digital image. The algorithm entries are the parameters of the circle, i.e. the centre (x_0, y_0) and radius r . The MCA aims to minimise 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. For these reasons, the MCA is considered as the quickest providing a subpixel precision.

The MCA starts at point $(x_0 + r, 0)$ and proceeds upwards-left by using integer additions and subtractions. Let p_i be a pixel within the first octant, i.e. within the part of the circumference starting at $(x_0 + r, 0)$ and ending at (x, y) with $x < y$. If the pixel coordinates of p_i are (x_k, y_k) , the coordinates of the next pixel p_{i+1} to be chosen are either (x_k, y_{k+1}) or (x_{k-1}, y_{k+1}) . The selection is conducted as

$$p_{i+1} = \begin{cases} (x_k, y_{k+1}) & \text{if } (x_{k-0.5} - x_0)^2 + (y_{k+1} - y_0)^2 - r^2 \leq 0 \\ (x_{k-1}, y_{k+1}) & \text{otherwise} \end{cases} \quad (10)$$

The above selection is performed iteratively until the coordinates of the chosen pixel satisfy $y > x$, which indicates that the whole first octant has been visited. The MCA process is outlined in Fig. 4.

3.4 Implementation of TLBO

The whole circle detection algorithm is summarised in Fig. 5. First, the preprocess stage is performed to obtain the edge map and the set of edgel positions V_c . Then, the TLBO algorithm is executed as previously explained. First a random population is generated. All the solutions within the first population must consist of three non-collinear points. To achieve this, the algorithm simply verifies if the denominator in (5) or (6) is different from zero. If this is not the case, that particular combination of points is not added to the population. Later in the process, in the teacher and learner phases, the algorithm prevents individuals consisting in collinear points by assigning a fitness equal to zero if the denominator in (5) or (6) is

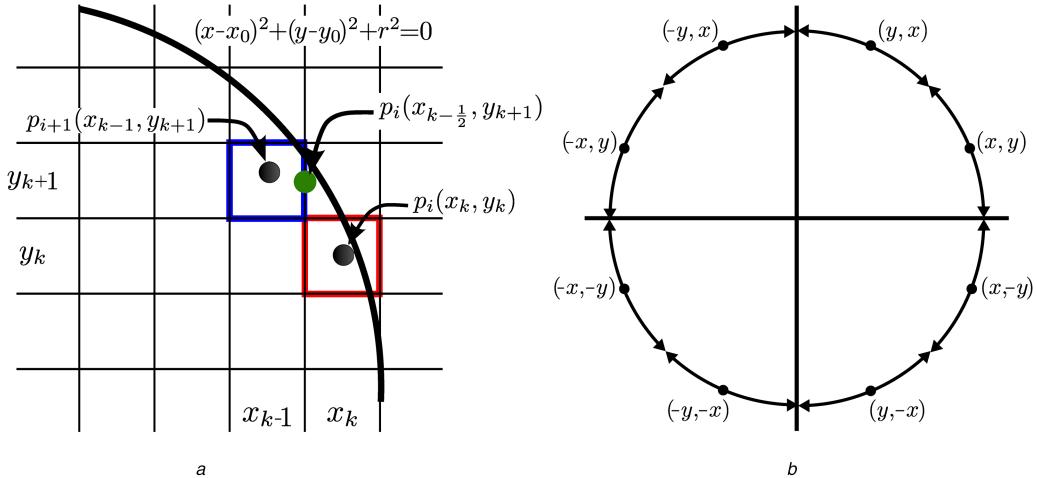


Fig. 4 Test point to evaluate the objective function are generated with the MCA

(a) Given a pixel $p_i(x_k, y_k)$ (shown in red and pointed out with an arrow), the two possibilities for the next chosen pixel p_{i+1} are the upper-middle and the upper-left neighbours. Since the midpoint $M_p(x_{k-1/2}, y_{k+1})$ (shown in green and pointed out with an arrow) lies inside the candidate circle, the selected pixel is the upper-left neighbour (shown in blue and pointed out with an arrow), (b) To reduce computational burden, every point (x, y) in the first octant is mirrored on the other seven

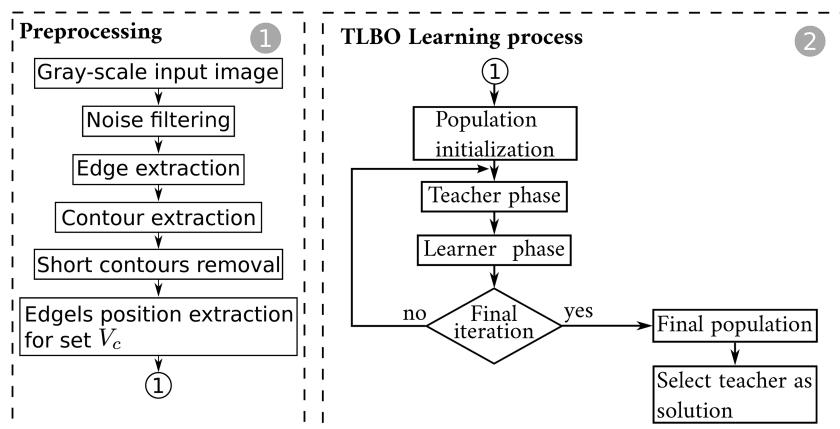


Fig. 5 General procedure of the proposed algorithm. The first stage involves preprocessing steps to construct an adequate search space. The second stage uses the TLBO algorithm to optimise over the previously constructed search space

Table 1 Parameter setup for the DE-based method

| Parameter | Value |
|-----------------------|-------|
| number of epochs | 200 |
| population size | 50 |
| differential weight | 0.25 |
| crossover probability | 0.80 |

equal to zero. At the end, the teacher of the last iteration of the TLBO algorithm is selected as the solution.

In order to achieve multiple-circle detection, the proposed method is executed as many times as needed. After one execution, the detected circle is removed from the search space and then the process is carried out again over the modified search space. Multiple executions of the whole process are carried out until the algorithm returns a solution whose fitness value is below a predefined threshold f_{\min} . After a set of solutions with fitness value greater than f_{\min} is obtained, a validation of all detected circles is performed by analysing continuity of the detected circumference segments as proposed in [25].

4 Experimental results

In this section, results of the application of the proposed method on real images are reported and compared with approaches belonging to each of the three categories of techniques as described in the introduction: HT-based, arc-segment-based, and meta-heuristics-based techniques. Additionally, a convergence analysis is made to

Table 2 Parameter setup for the TLBO-based method

| Parameter | Value |
|-----------------|-------|
| iterations | 100 |
| population size | 50 |

explore the evolution of the search process of the meta-heuristics-based detectors.

Comparisons with our proposed method are performed with non-probabilistic and probabilistic HT-based techniques, the adaptive HT and the GRCD-R algorithm, respectively. The TLBO-based detector is also compared with an arc-segment-based technique, EDCircles. For completeness, the proposed approach is compared with another meta-heuristic technique based on DE. For the experiments, the adaptive HT utilised was the implementation of the OpenCV library. The GRCD-R algorithm was implemented as in [14] since there is no publicly available implementation of the algorithm. Finally, the online demo of EDCircles provided by the authors was utilised.

4.1 Parameter setup

Tables 1 and 2 present the parameters for the DE-based method and the TLBO-based method, respectively. The set of parameters for each meta-heuristic have been chosen manually in favour of performing a fair comparison. First, the settings of the TLBO algorithm (i.e. number of iterations and population size) 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.

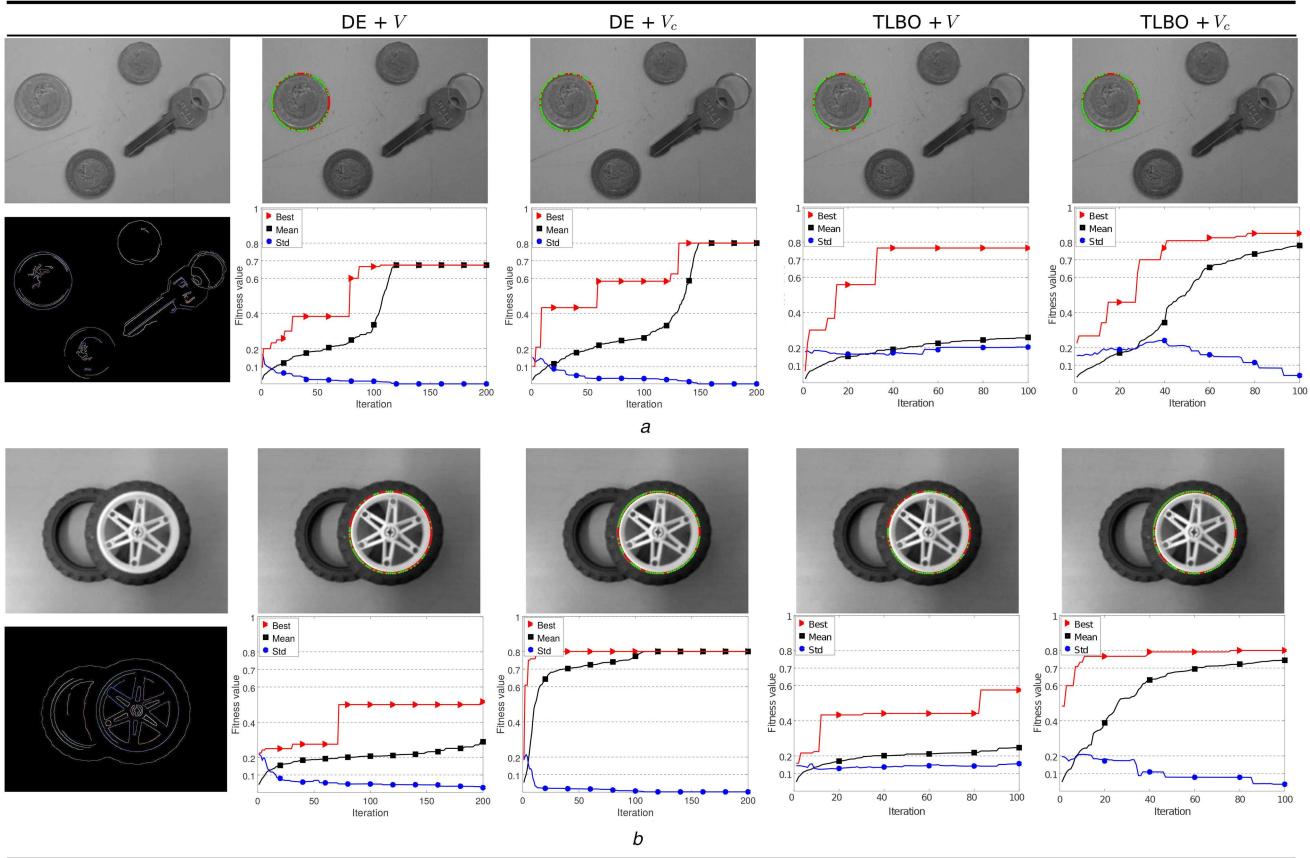


Fig. 6 Evolution of the population for the meta-heuristic approaches

(a) Results from image Coins, (b) Results for image Wheels. The first column shows the input image and the edge map. The second and third columns show the performance of the DE-based detector optimising over V and V_c , respectively. On the other hand, the fourth and fifth columns show the performance of the TLBO-based detector optimising over V and V_c , respectively. The detected circle is indicated along with the pixels actually present in the edge map

Finally, the rest of the parameters for the DE algorithm were tuned experimentally.

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 about the problem. The number of FEs for each approach is now described.

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

$$FES_{TLBO} = (2N + 1) \cdot M. \quad (11)$$

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

$$FES_{DE} = (N + 1) \cdot M. \quad (12)$$

The parameters for both meta-heuristics were the same for the whole set of experiments. The thresholds for the preprocessing step and the multiple circle detection were fixed at $T_c = 2\pi 20$ and $f_{\min} = 0.4$, respectively.

4.2 Convergence analysis

We first examine the behaviour of the meta-heuristic approaches. To this end, we examine for each generation the evolution of the best individual, the mean fitness value, and the similarity of the population. We also analyse how different representations of the

search space contribute to the convergence of the meta-heuristic-based detectors. In the analysis, the algorithms optimise over two different representations of the search space: (i) edgels locations are indexed in a set V one after the other as the edgels are visited within the edge image from left to right and top to bottom; and (ii) edgels locations are indexed in a set V_c with linked contours as described in Section 3.1. Images Coins (711×521) and Wheels (618×459), shown in Fig. 6, were given as input to the DE-based detector and the TLBO-based detector. The two detectors optimise over V and V_c separately. We compare the search process each algorithm performs for detecting the same circle. In consequence, the analysis includes the data regarding the detection of only one circle per image.

In order to compute a measure of similarity S for a population P , we consider student $X = \{i, j, k\}$ 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 of the student in P . The output of the last process is a set of values between 0 and 1. We thus define S as the standard deviation of these values. For convenience, the standard deviation is plotted along with the best individual in Fig. 6.

As shown in Fig. 6, both detectors (DE-based and TLBO-based) output a circle that is actually present in the image when optimising over V or V_c . However, it is clear from the graphs that both meta-heuristic-based detectors benefit when the search space is organised according to connected contours. For the optimisation over V_c , the two approaches detect a circle that better adjusts the circle that is actually present in the edge map, i.e. more edgels are truly present within the circumference of the detected circle. Thereby, the fitness value of the best individual increases. In Fig. 6, the detected circle is shown as a continuous line, whilst the pixels that actually exist within the edge map are shown as dots.

Organising the search space using connected contours permits to obtain accuracy while keeping a simpler coding than, for instance, a 6D vector with three Cartesian coordinates of the solution. A more complicated encoding of the solution with a 6D vector could lead, later in the evolution process, to vectors whose coordinates do not exist in the edge map. This would make necessary an extra validation step. For this reason, our method adopts the encoding of previous work as in [22–25], but organising the search space according to linked contours.

The accuracy of the proposed approach is the result of a combination of an appropriate preprocessing, an organisation of the search space, and also the behaviour of the TLBO algorithm regarding the exploration and exploitation of the search space. As shown by the graphs in Fig. 6, the TLBO algorithm explores more the search space than the DE. This is represented by the similarity of the population in each generation. The DE approach, for instance, usually converges around iteration 150, and thus the standard deviation is equal to zero. In other words, the DE stopped searching by the iteration 150. Differently, the TLBO algorithm makes more exploration of the search space, since there are more variability within the population while keeping the best solution. With the particular settings in Tables 1 and 2, if only more iterations are given to the DE, no changes in the results would exist. To change the exploration and exploitation capabilities of the DE approach, the differential weight and the crossover probability need to be tuned. Conversely, the TLBO algorithms do not require the tuning of algorithm-specific parameters to change the convergence.

Intuitively, the factors TF and r in the teacher phase of the TLBO algorithm in (1) contribute to the exploration and exploitation of the search space. When $TF = 1$ and r approximates to 1, the individuals tend to approach the teacher, thus exploiting the search space near the teacher. Conversely, when $TF = 2$, the individuals tend to explore far from the teacher depending on the value of r . Furthermore, the learner phase also contributes to a guided search, since students tend to approach better students.

In the next subsections, the obtained results for different tasks and comparisons against other methods for circle extraction are presented.

4.3 Image performance

In order to evaluate the performance of the TLBO-based circle detector, several tests have been carried out addressing the following tasks:

- i. Circle localisation.
- ii. Multiple circle detection.
- iii. Blurred circle detection.
- iv. Circular approximation from occluded circles.
- v. Circle detection under Gaussian noise conditions.

The experimental setup includes the use of several real images commonly used in the literature [14, 18, 21]: Streetlight (344×320), Moon (393×296), Baseball (538×402), Balls (315×313), Footballs (690×694), Plates (504×489), Objects (743×623), Speaker (502×543), Bowling (467×480), Gobang (234×231), Watch (509×601), Book (315×416), Cookies (295×292), and Insulator (451×331). All images, shown in Figs. 7 and 8, contain a different amount of circles and serve to evaluate distinct issues about the proposed approach. The proposed method and the methods for comparison were executed 100 times for each test image (except for the non-probabilistic approaches), and a statistical analysis was performed with the resulting data. This analysis is discussed in the next subsection. All the experiments were executed on a 2.80 GHz Intel Core i7-7700HQ CPU, with a C++ implementation of the algorithm. To perform edge extraction and border following, implementations from OpenCV were utilised.

As stated above, different tests have been performed to evaluate the proposed method. We first discuss in this subsection a qualitative evaluation, and then in the next subsection, we discuss two different metrics used to quantitatively evaluate the

performance of the algorithms. Within this and the next subsection, it is shown that the TLBO-based detector obtains good results, considering the existing noise, irregularities in the shape and including the partial occlusion of circles. Furthermore, since the proposed method aims to find the best approximation for a circle, it is possible for the method to successfully detect circular shapes even when dealing with noisy, incomplete, or partially occluded circular shapes. These results are due to the fact that an efficient meta-heuristic was used with a combination of an appropriate fitness function and a suited search space.

First, we investigated the efficiency regarding circle localisation and shape discrimination. Images Streetlight, Moon, and Baseball, in Fig. 7, are used to demonstrate the capability of the proposed method to accurately locate circles, and to differentiate circular patterns over any other shapes which might be present in the image. It is clear from the figure that the proposed method performs accurately, and even outperforms the adaptive HT. The other three methods equate the results of the proposed approach. However, in the case of the Baseball image, EDCircles detects two different circles. It is clear that the DE-based detector and the TLBO-based detector benefit from the preprocessing step.

The edge map shown in the third row of images 7 and 8 is computed as explained in Section 3.1. This edge map is the input for the TLBO-based and DE-based methods. Conversely, the adaptive HT, GRCD-R and the EDCircles perform over a different edge map. For the adaptive HT, the input image is preprocessed with Gaussian filtering and Canny edge extraction. For the GRCD-R, Sobel operators are used to detect edges. EDCircles, on its part, uses a different approach for edge extraction as remarked in the introduction.

Regarding the second task, the proposed method can also detect multiple and concentric circles on a real image. Results from images Balls, Footballs, Plates, Objects and Speaker, shown in Fig. 7, demonstrate that the TLBO performance can be compared with the GRCD-R, EDCircles and DE methods. EDcircles, however, benefits from its edge extraction approach with these particular images and detects more true circles in the Speaker image. The adaptive HT, on its part, does not obtain good results with these particular images.

For the third and fourth tasks, regarding blurred circle detection and circular approximation, we present the images Bowling, Gobang, Watch, Book and Cookies that are shown in Fig. 8. The test images present occluded and blurred circles. The five approaches succeed in the case of the Bowling image, although the EDCircles algorithm detects light reflections as circles. Also, EDCircles detect less circles in the Gobang and Book images due to the fact that the algorithm better fits ellipses in these images. However, EDCircles falsely detect the letters ‘e’, ‘D’, and ‘c’ as circles.

The final task to evaluate is the performance of the method under Gaussian noise. For this reason, we tested the proposed approach on a real image contaminated with Gaussian noise of $\mu = 0$, and σ values of 10 and 20. As demonstrated in Fig. 8, in the case of the Insulator image, it can be said that the proposed circle detector based on the TLBO algorithm is robust under noisy conditions, and that outperforms the adaptive HT and the GRCD-R algorithms.

4.4 Performance evaluation

Real images rarely contain perfect circles because of image discretisation. Therefore, in the interest of testing the accuracy of the algorithms, the results are compared to a ground-truth circle which is manually detected from the original image. To evaluate all circle detectors, two score metrics were used to quantify their performance. We next discuss the metrics, and then show the performance of the algorithms.

The first metric looks at the per cent overlap of the circle area. This starts with the overlap percentage calculated by

$$O_v = (C_1, C_2) = \frac{C_1 \cap C_2}{C_1 \cup C_2}, \quad (13)$$

| | Streetlight | Moon | Baseball | Balls | Footballs | Plates | Objects | Speaker |
|----------------------|-------------|------|----------|-------|-----------|--------|---------|---------|
| Original image | | | | | | | | |
| Ground truth circles | | | | | | | | |
| Edge map | | | | | | | | |
| Adaptive HT | | | | | | | | |
| GRCD-R | | | | | | | | |
| EDCircles | | | | | | | | |
| DE | | | | | | | | |
| TLBO | | | | | | | | |

Fig. 7 Performance of the five methods on test images regarding the task of circle localisation (columns 1–3), and the task of detecting multiple circles (columns 4–8). The second row shows the number of manually detected ground-truth circles; while the third row shows the edge map used for the meta-heuristics. For each method, the number of detected circles (including false positives) for the best case is indicated

where O_v 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. The overlap error E_o is then calculated as

$$E_o = 1 - O_v. \quad (14)$$

For multiple circle evaluation, the average value of the E_o errors, M_{E_o} , is calculated by

$$M_{E_o} = \left(\frac{1}{NC} \right) \cdot \sum_{i=1}^{NC} E_{o_i}, \quad (15)$$

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

The second metric, as proposed in [35], measures the deviation of the parameters of circle 1, $C_1(x_1, y_1, r_1)$, as compared with the parameters of circle 2, $C_2(x_2, y_2, r_2)$, by

$$E_s = \eta \cdot (|x_1 - x_2| + |y_1 - y_2|) + \mu \cdot |r_1 - r_2|, \quad (16)$$

where the first term represents the shift of the centre of circle 1 as it is compared to the circle 2, and the second term accounts for the difference between their radii. The two associated weights η and μ may be chosen according to the required accuracy. Finally, similar to the first metric, when multiple circle evaluation is estimated, the error average is calculated by

$$M_{E_s} = \left(\frac{1}{NC} \right) \cdot \sum_{i=1}^{NC} E_{s_i}. \quad (17)$$

In this metric evaluation, η has been set to 0.05 and μ to 0.1. This particular choice of parameters ensures that the radii difference is strongly weighted than the difference of centre position between the manually detected circle and the circle detected by the algorithm. Notice that with this setting, $E_s < 1$ for a maximum tolerated difference of radius length of 10 pixels or a maximum deviation in the location of the centre of 20 pixels.

For the stochastic 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_T the set of circles detected by the

| | Bowling | Gobang | Watch | Book | Cookies | Insulator | $\sigma=10$ Insulator | $\sigma=20$ Insulator |
|----------------------|---------|--------|-------|------|---------|-----------|--------------------------|--------------------------|
| Original image | | | | | | | | |
| Ground truth circles | | | | | | | | |
| Edge map | | | | | | | | |
| Adaptive HT | | | | | | | | |
| GRCD-R | | | | | | | | |
| EDCircles | | | | | | | | |
| DE | | | | | | | | |
| TLBO | | | | | | | | |

Fig. 8 Performance of the five methods on test images regarding to the task of detecting occluded or blurred circles (columns 1–2), the task of circular approximation (columns 3–5), and the task of circle detection under different levels of Gaussian noise (columns 6–8). The second row shows the number of manually detected ground-truth circles; while the third row shows the edge map used for the meta-heuristics. For each method, the number of detected circles (including false positives) for the best case is indicated

method. For each circle in C_T , we find the match circle in C_{GT} that minimises the sum $E_o + E_s$ subject to $E_o < 0.20$ and $E_s < 1$. Only one circle in C_T 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:

$$S_r = \frac{\text{card}(C_{TP})}{\max(\text{card}(C_{GT}), \text{card}(C_T))}, \quad (18)$$

where $\text{card}(\cdot)$ is the function that returns the cardinality of a set. It is noted that S_r is equal to 1 when all the ground-truth circles are 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. In other words, any

circle within C_T when compared with circles in C_{GT} meets the selected matching criteria of $E_o < 0.20$ and $E_s < 1$. The computation of the success rate as in (18) penalises even multiple detections of circles with similar centres and radii.

On the other hand, the computations of (14)–(17) are done, for the stochastic approaches, with the detected circles in C_{TP} ; and for the non-probabilistic approaches, with the detected circles that match the criteria of $E_o < 0.20$ and $E_s < 1$ when compared with the ground-truth circles. This enables us to obtain a measure of the accuracy of the true positives detected circles, whilst the success rate as computed in (18) allows us to obtain a measure of the performance of the algorithm to detect all ground-truth circles and discard false positives.

The whole set of original images was processed by each of the five circle detection algorithms. Then, the performance evaluation was computed with the aforementioned metrics. Tables 3–6 expose the computed average errors for the three stochastic approaches, as well as the average time, and the success rate. Table 7, on its part, shows the errors for the non-probabilistic approaches. In the case

Table 3 Average time, average E_o error and E_s error, and success rate, considering the test images in Fig. 7 for the circle localisation task

| Image | Average time \pm std, s | | | Average E_o \pm std | | | Average E_s \pm std ($\eta = 0.05, \mu = 0.1$) | | | Success rate | | |
|-------------|---------------------------|------------------|------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|---|-------------------------------------|-------------------------------------|--------------|------------|------------|
| | GRCD-R | DE | TLBO | GRCD-R | DE | TLBO | GRCD-R | DE | TLBO | GRCD-R | DE | TLBO |
| Streetlight | 0.352 ± 0.08 | 0.279 ± 0.08 | 0.253 \pm 0.04 | 0.068 ± 0.021 | 0.055 ± 0.002 | 0.047 \pm 0.002 | 0.258 ± 0.022 | 0.291 ± 0.095 | 0.250 \pm 0.070 | 92 | 93 | 95 |
| Moon | 0.282 ± 0.06 | 0.232 ± 0.05 | 0.249 ± 0.03 | 0.071 \pm 0.033 | 0.097 ± 0.075 | 0.083 ± 0.035 | 0.211 \pm 0.013 | 0.278 ± 0.020 | 0.221 ± 0.060 | 100 | 100 | 100 |
| Baseball | 0.393 ± 0.11 | 0.293 ± 0.01 | 0.285 \pm 0.04 | 0.262 ± 0.082 | 0.215 \pm 0.091 | 0.237 ± 0.004 | 0.293 ± 0.050 | 0.223 \pm 0.035 | 0.291 ± 0.011 | 97 | 98 | 100 |

Bold values indicate the best result for a particular experiment.

Table 4 Average time, average E_o error and E_s error, and success rate, considering the test images in Fig. 7 for the multiple circle detection task

| Image | Average time \pm std, s | | | Average E_o \pm std | | | Average E_s \pm std ($\eta = 0.05, \mu = 0.1$) | | | Success rate | | |
|-----------|------------------------------------|------------------|------------------------------------|-------------------------|-------------------------------------|-------------------------------------|---|-------------------------------------|-------------------------------------|--------------|-------|--------------|
| | GRCD-R | DE | TLBO | GRCD-R | DE | TLBO | GRCD-R | DE | TLBO | GRCD-R | DE | TLBO |
| Balls | 0.354 \pm 0.07 | 0.597 ± 0.09 | 0.492 ± 0.01 | 0.121 ± 0.007 | 0.089 \pm 0.015 | 0.109 ± 0.011 | 0.299 ± 0.029 | 0.193 \pm 0.042 | 0.206 ± 0.050 | 93.33 | 96.66 | 100 |
| Footballs | 0.578 ± 0.03 | 0.426 ± 0.02 | 0.418 \pm 0.01 | 0.097 ± 0.008 | 0.093 ± 0.007 | 0.075 \pm 0.009 | 0.288 ± 0.038 | 0.295 ± 0.012 | 0.238 \pm 0.071 | 96.25 | 93.42 | 98.60 |
| Plates | 2.174 ± 0.06 | 1.776 ± 0.07 | 1.376 \pm 0.04 | 0.092 ± 0.007 | 0.106 ± 0.019 | 0.098 \pm 0.011 | 0.255 ± 0.023 | 0.275 ± 0.033 | 0.223 \pm 0.095 | 92.33 | 94.87 | 96.65 |
| Objects | 1.472 ± 0.01 | 1.050 ± 0.07 | 0.987 \pm 0.04 | 0.094 ± 0.003 | 0.068 \pm 0.001 | 0.077 ± 0.014 | 0.282 ± 0.011 | 0.193 \pm 0.078 | 0.221 ± 0.062 | 92.50 | 94.66 | 95.55 |
| Speaker | 1.174 ± 0.06 | 1.651 ± 0.01 | 1.145 \pm 0.07 | 0.075 ± 0.002 | 0.078 ± 0.004 | 0.071 \pm 0.025 | 0.289 ± 0.018 | 0.268 ± 0.011 | 0.257 \pm 0.045 | 30.13 | 30.11 | 34.16 |

Bold values indicate the best result for a particular experiment.

Table 5 Average time, average E_o error and E_s error, and success rate, considering the test images in Fig. 8 for the blurred circle detection and circular approximation tasks

| Image | Average time \pm std, s | | | Average E_o \pm std | | | Average E_s \pm std ($\eta = 0.05, \mu = 0.1$) | | | Success rate | | |
|---------|---------------------------|-------------------------------------|------------------------------------|-------------------------|-------------------------------------|-------------------------------------|---|-------------------------------------|-------------------------------------|--------------|--------------|--------------|
| | GRCD-R | DE | TLBO | GRCD-R | DE | TLBO | GRCD-R | DE | TLBO | GRCD-R | DE | TLBO |
| Bowling | 1.095 ± 0.005 | 1.059 \pm 0.028 | 1.121 ± 0.02 | 0.135 ± 0.002 | 0.125 \pm 0.010 | 0.132 ± 0.022 | 0.287 ± 0.042 | 0.213 \pm 0.010 | 0.273 ± 0.005 | 93.12 | 97.40 | 97.80 |
| Gobang | 0.889 ± 0.07 | 0.940 ± 0.091 | 0.738 \pm 0.04 | 0.143 ± 0.012 | 0.112 \pm 0.007 | 0.155 ± 0.012 | 0.238 ± 0.012 | 0.226 \pm 0.007 | 0.293 ± 0.001 | 90.33 | 85.83 | 91.16 |
| Watch | 1.397 ± 0.01 | 1.589 ± 0.030 | 1.243 \pm 0.03 | 0.175 ± 0.003 | 0.196 ± 0.017 | 0.164 \pm 0.007 | 0.297 ± 0.051 | 0.280 ± 0.005 | 0.215 \pm 0.007 | 60.66 | 65.33 | 68.66 |
| Book | 1.174 ± 0.035 | 1.125 ± 0.017 | 1.086 \pm 0.04 | 0.153 ± 0.007 | 0.189 ± 0.015 | 0.119 \pm 0.004 | 0.282 ± 0.001 | 0.274 ± 0.023 | 0.253 \pm 0.010 | 47.5 | 55.83 | 48.50 |
| Cookies | 2.174 ± 0.06 | 1.776 ± 0.07 | 1.376 \pm 0.04 | 0.173 ± 0.010 | 0.152 ± 0.010 | 0.132 \pm 0.001 | 0.279 ± 0.052 | 0.278 ± 0.031 | 0.243 \pm 0.007 | 93.60 | 96.80 | 96.40 |

Bold values indicate the best result for a particular experiment.

Table 6 Average time, average E_o error and E_s error, and success rate, considering the test images in Fig. 8 used to demonstrate the performance of the methods under Gaussian noise

| Noise | Average time \pm std, s | | | Average E_o \pm std | | | Average E_s \pm std ($\eta = 0.05, \mu = 0.1$) | | | Success rate | | |
|---------------|---------------------------|------------------------------------|------------------------------------|-------------------------|-------------------------------------|-------------------------------------|---|-------------------|-------------------------------------|--------------|-------|--------------|
| | GRCD-R | DE | TLBO | GRCD-R | DE | TLBO | GRCD-R | DE | TLBO | GRCD-R | DE | TLBO |
| insulator | 0.795 ± 0.08 | 0.653 \pm 0.03 | 0.682 ± 0.08 | 0.097 ± 0.011 | 0.089 ± 0.001 | 0.073 \pm 0.005 | 0.201 ± 0.011 | 0.194 ± 0.024 | 0.132 \pm 0.043 | 97 | 97.60 | 98.66 |
| $\sigma = 10$ | 1.268 ± 0.04 | 1.801 ± 0.01 | 1.638 \pm 0.06 | 0.093 ± 0.001 | 0.095 ± 0.008 | 0.085 \pm 0.001 | 0.203 ± 0.021 | 0.199 ± 0.034 | 0.139 \pm 0.018 | 90.20 | 90.80 | 91.45 |
| $\sigma = 20$ | 2.379 ± 0.02 | 1.971 ± 0.05 | 1.832 \pm 0.02 | 0.106 ± 0.001 | 0.069 \pm 0.003 | 0.077 ± 0.009 | 0.216 ± 0.032 | 0.131 ± 0.012 | 0.213 ± 0.005 | 87.60 | 83.40 | 90.20 |

Bold values indicate the best result for a particular experiment.

Table 7 Average time, average E_o error and E_s error, and success rate, considering the test images in Figs. 7 and 8 for the non-probabilistic approaches regarding the whole set of tasks

| Image | Adaptive HT | | | | EDCircles | | | | Adaptive HT | | | | EDCircles | | | | |
|-------------|-------------|-------|-------|-------|-----------|---------|-------|-------|-------------|-------|-------|-----------|-----------|-------|-------|-------|-------|
| | Time | E_o | E_s | Time | E_o | E_s | Time | E_o | E_s | Time | E_o | E_s | Time | E_o | E_s | Time | |
| Streetlight | 0.230 | 0.193 | 0.578 | 0.053 | 0.089 | Objects | 0.327 | 0.377 | 0.935 | 0.019 | 0.126 | Cookies | 0.157 | 0.109 | 0.351 | 0.131 | 0.325 |
| Moon | 0.135 | 0.277 | 0.985 | 0.021 | 0.108 | Speaker | 0.457 | 0.253 | 0.587 | 0.022 | 0.141 | Insulator | 0.872 | 0.016 | 0.108 | 0.019 | 0.124 |
| Baseball | 0.140 | 0.035 | 0.123 | 0.089 | 0.093 | Bowling | 0.368 | 0.049 | 0.128 | 0.029 | 0.018 | Insulator | 2.681 | — | — | 0.019 | 0.124 |
| Balls | 0.260 | 0.102 | 0.718 | 0.109 | 0.182 | Gobang | 0.287 | 0.036 | 0.107 | 0.164 | 0.268 | Insulator | 12.365 | — | — | 0.019 | 0.124 |
| Footballs | 0.271 | 0.699 | 1.253 | 0.025 | 0.125 | Watch | 0.381 | 0.115 | 0.150 | 0.034 | 0.137 | | | | | | |
| Plates | 0.381 | 0.876 | 1.927 | 0.183 | 0.139 | Book | 0.278 | 0.109 | 0.086 | 0.289 | 0.421 | | | | | | |

of EDCircles, computational time is omitted since the used implementation provided by the authors is an online application.

As shown by the results presented in the tables, for almost every test image, our proposed method accurately detect all ground-truth circles, since the success rate is high, and the average E_o and E_s errors are low in comparison with the other methods. However, in the case of image Speaker, the proposed method obtains lower success rate, since less ground-truth circles were detected. On the other hand, in the case of images Watch and Book, regarding the circular approximation task, our algorithm performs less accurately, but so do the other algorithms. This is due to the fact that within the edge map, ellipses are a better fit.

5 Conclusions

This work has presented a novel application of the TLBO algorithm to the task of automatic multi-circle detection. In our approach, the detected circles hold a sub-pixel accuracy due to the use of the circle equation and the MCA method. The proposed circle detector finds circular shapes among cluttered and noisy images with no consideration of the conventional HT principles. Rather, the proposed detector searches for the best approximation to circular shapes within the edge image using a meta-heuristic approach. The algorithm uses the encoding of three non-collinear edgels to propose a solution (student). Then, different solutions are graded using an objective function that evaluates if the candidate circle under test is actually present in the edge image. After some iterations are performed, the final solution is the one with the higher score.

In order to test the performance of the proposed approach, computation time and accuracy have been compared with four different approaches based on different principles. These are the adaptive HT, the GRCD-R algorithm, EDCircles algorithm, and a meta-heuristic detector based on DE. Our approach accurately detects circles in complex images with little visual distortion despite the presence of non-circular shapes, circle occlusions, blurred effects, and noise. Furthermore, two score metrics, which can effectively evaluate the mismatch between a ground-truth circle and a machine-detected circle were utilised. The metrics allow to evaluate and compare quantitatively the performance of the method. Also, since our proposed circle detector is based on a meta-heuristic, an evaluation of the convergence of the algorithm was performed. In the analysis, we tested different representations of the search space, and compared the TLBO convergence with the convergence of the DE-based method.

The aim of our work is to demonstrate that the TLBO algorithm can be effectively considered as an alternative approach for detecting circular shapes when an adequate preprocess is done for the input image. In comparison with other approaches, the results obtained by the TLBO-based detector required less effort by the user because there are less parameters to tune, which is a clear advantage in comparison to other meta-heuristic approaches, and even to the adaptive HT and the GRCD-R algorithms that also require the set of various parameters.

Regarding future works, we will make efforts to expand our method to detect ellipses and reduce the computation time. Furthermore, we believe more effort has to be done into exploring different objective functions for circle detection since, until now, only edge pixels positions from the input image have been used.

Providing additional data to the method, like gradient information, might represent an improvement in accuracy.

6 References

- [1] Greenhalgh, J., Mirmehdi, M.: ‘Real-time detection and recognition of road traffic signs’, *IEEE Trans. Intell. Transp. Syst.*, 2012, **13**, pp. 1498–1506
- [2] Ngo, H., Rakvic, R., Brousseau, R., et al.: ‘Resource-aware architecture design and implementation of Hough transform for a real-time iris boundary detection system’, *IEEE Trans. Consum. Electron.*, 2014, **60**, pp. 485–492
- [3] Wan, H.L., Li, Z.C., Qiao, J.P., et al.: ‘Non-ideal iris segmentation using anisotropic diffusion’, *IET Image Process.*, 2013, **7**, pp. 111–120
- [4] Costa, L., Marcondes, C.R.: ‘Shape analysis and classification: theory and practice’ (CRC Press, Boca Raton, FL, 2001)
- [5] Zhang, S., Zhang, Z., Wang, H.T.: ‘Research of robot color logo orientation based on Hough transform’. Proc. Int. 2nd Conf. on Intelligent Human-Machine Systems and Cybernetics IHMSC, Nanjing, China, 2010, vol. **1**, pp. 56–60
- [6] Cai, J., Huang, P., Chen, L., et al.: ‘An efficient circle detector not relying on edge detection’, *Adv. Space Res.*, 2016, **57**, pp. 2359–2375
- [7] Illingworth, J., Kittler, J.: ‘The adaptive Hough transform’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 1987, **5**, pp. 690–698
- [8] Li, H., Lavin, M., Le-Master, R.: ‘Fast Hough transform: a hierarchical approach’, *Comput. Graph. Image Process.*, 1986, **36**, pp. 139–161
- [9] Ben-Tzvi, D., Sandler, M. B.: ‘A combinatorial Hough transform’, *Pattern Recognit.*, 1990, **3**, pp. 167–174
- [10] Shvaysiter, H., Bergen, J. R.: ‘Monte Carlo Hough transform’. Proc. 7th Scandinavian Conf. Image anal., Aalborg, Denmark, 1991, pp. 80–87
- [11] Kiryati, N., Eldar, Y., Bruckstein, A.: ‘A probabilistic Hough transform’, *Pattern Recognit.*, 1991, **24**, pp. 303–316
- [12] Xu, L., Oja, E., Kultanen, P.: ‘A new curve detection method: randomized Hough transform (rht)’, *Pattern Recognit. Lett.*, 1990, **11**, pp. 331–338
- [13] Chen, T. C., Chung, K. L.: ‘An efficient randomized algorithm for detecting circles’, *Comput. Vis. Image Underst.*, 2001, **83**, pp. 172–191
- [14] Chung, K. L., Huang, Y. H., Shen, S. M., et al.: ‘Efficient sampling strategy and refinement strategy for randomized circle detection’, *Pattern Recognit.*, 2012, **45**, pp. 252–263
- [15] Jia, L. Q., Peng, C. Z., Liu, H. M., et al.: ‘A fast randomized circle detection algorithm’. 2011 4th Int. Congress on Image and Signal Processing (CISP), Shanghai, China, 2011, vol. **2**, pp. 820–823
- [16] De Marco, T., Cazzato, D., Leo, M., et al.: ‘Randomized circle detection with isophotes curvature analysis’, *Pattern Recognit.*, 2015, **48**, pp. 411–421
- [17] Zhang, H., Wiklund, K., Andersson, M.: ‘A fast and robust circle detection method using isosceles triangles sampling’, *Pattern Recognit.*, 2016, **54**, pp. 218–228
- [18] Yu, H., Wang, T.: ‘Vision-based technique for circle detection and measurement using lookup table and bitwise centre accumulator’, *JOSA A*, 2017, **34**, pp. 415–423
- [19] Rosin, P. L., West, G. A. W.: ‘Nonparametric segmentation of curves into various representations’, *IEEE Trans. Pattern Anal. Mach. Intell.*, 1995, **17**, pp. 1140–1153
- [20] Mai, F., Hung, Y. S., Zhong, H., et al.: ‘A hierarchical approach for fast and robust ellipse extraction’, *Pattern Recognit.*, 2008, **41**, pp. 2512–2524
- [21] Akınlar, C., Topal, C.: ‘EDCircles: a real-time circle detector with a false detection control’, *Pattern Recognit.*, 2013, **46**, pp. 725–740
- [22] Ayala-Ramirez, V., Garcia-Capulin, C.H., Pérez-García, A., et al.: ‘Circle detection on images using genetic algorithms’, *Pattern Recognit. Lett.*, 2006, **27**, pp. 652–657
- [23] Dong, N., Wu, C.H., Ip, W.H., et al.: ‘An opposition-based chaotic GA/PSO hybrid algorithm and its application in circle detection’, *Comput. Math. Appl.*, 2012, **64**, pp. 1886–1902
- [24] Cuevas, E., Sencio-Echauri, F., Zaldivar, D., et al.: ‘Multi-circle detection on images using artificial bee colony (ABC) optimization’, *Soft Comput.*, 2012, **16**, pp. 281–296
- [25] Cuevas, E., Zaldivar, D., Pérez-Cisneros, M., et al.: ‘Circle detection using discrete differential evolution optimization’, *Pattern Anal. Appl.*, 2011, **14**, pp. 93–107
- [26] Fourie, J.: ‘Robust circle detection using harmony search’, *J. Optim.*, 2017, **2017**, pp. 1–11
- [27] Cuevas, E., Wario, F., Zaldivar, D., et al.: ‘Circle detection on images using learning automata’, *IET Comput. Vis.*, 2012, **6**, pp. 121–132

- [28] Rao, R.V., Savsani, V.J., Vakharia, D.: ‘Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems’, *Comput. Aided Des.*, 2011, **43**, pp. 303–315
- [29] Rao, R.V., Savsani, V.J., Vakharia, D.: ‘Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems’, *Inf. Sci.*, 2012, **183**, pp. 1–15
- [30] Rao, R.: ‘Review of applications of TLBO algorithm and a tutorial for beginners to solve the unconstrained and constrained optimization problems’, *Decis. Sci. Lett.*, 2016, **5**, pp. 1–30
- [31] Amiri, B.: ‘Application of teaching-learning-based optimization algorithm on cluster analysis’, *J. Basic Appl. Sci. Res.*, 2012, **2**, pp. 11795–11802
- [32] Nayak, M., Nayak, C., Rout, P.: ‘Application of multi-objective teaching learning based optimization algorithm to optimal power flow problem’, *Proc. Technol.*, 2012, **6**, pp. 255–264
- [33] Suzuki, S.: ‘Topological structural analysis of digitized binary images by border following’, *Comput. Graph. Image Process.*, 1985, **30**, pp. 32–46
- [34] Van Aken, J.: ‘An efficient ellipse-drawing algorithm’, *IEEE Comput. Graph. Appl.*, 1984, **4**, pp. 24–35
- [35] Cuevas, E., Osuna, V., Oliva, D.: ‘Multi-circle detection on images’, in ‘*Evolutionary computation techniques: a comparative perspective*’ (Springer International Publishing, Berlin, Heidelberg, 2017), pp. 35–64