

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/266087332>

# Randomized circle detection with isophotes curvature analysis

Article in Pattern Recognition · August 2014

DOI: 10.1016/j.patcog.2014.08.007

---

CITATIONS

52

READS

1,577

4 authors:



**Tommaso De Marco**  
Italian National Research Council

28 PUBLICATIONS 329 CITATIONS

[SEE PROFILE](#)



**Dario Cazzato**  
University of Luxembourg

41 PUBLICATIONS 648 CITATIONS

[SEE PROFILE](#)



**Marco Leo**  
Italian National Research Council

179 PUBLICATIONS 4,382 CITATIONS

[SEE PROFILE](#)



**Cosimo Distante**  
Italian National Research Council

259 PUBLICATIONS 4,545 CITATIONS

[SEE PROFILE](#)



## Randomized circle detection with isophotes curvature analysis

Tommaso De Marco <sup>a,\*</sup>, Dario Cazzato <sup>a,b</sup>, Marco Leo <sup>a</sup>, Cosimo Distante <sup>a</sup>

<sup>a</sup> Istituto Nazionale di Ottica - CNR, Viale delle Libertà n. 3, 73010 Arnesano, LE, Italy

<sup>b</sup> Università del Salento, 73100 Lecce, Italy

### ARTICLE INFO

#### Article history:

Received 22 August 2013

Received in revised form

20 July 2014

Accepted 11 August 2014

#### Keywords:

Circle detection

Sampling strategy

Isophotes

Density estimation

### ABSTRACT

Circle detection is a critical issue in image analysis and object detection. Although Hough transform based solvers are largely used, randomized approaches, based on the iterative sampling of the edge pixels, are object of research in order to provide solutions less computationally expensive. This work presents a randomized iterative work-flow, which exploits geometrical properties of isophotes in the image to select the most meaningful edge pixels and to classify them in subsets of equal isophote curvature. The analysis of candidate circles is then performed with a kernel density estimation based voting strategy, followed by a refinement algorithm based on linear error compensation. The method has been applied to a set of real images on which it has also been compared with two leading state of the art approaches and Hough transform based solutions. The achieved results show how, discarding up to 57% of unnecessary edge pixels, it is able to accurately detect circles within a limited number of iterations, maintaining a sub-pixel accuracy even in the presence of high level of noise.

© 2014 Elsevier Ltd. All rights reserved.

### 1. Introduction

Circle detection is a critical issue in pattern recognition and computer vision [1]. Most of the current solutions are based on Hough transform [2], the well-known method to recognize complex models in computer vision. The Hough transform has been used for the first time to detect curves in images in [3] and since then different variants have been proposed [4,5]. The circular Hough transform (CHT) consists of a sequence of algorithmic steps: an edge map of the image is firstly computed and then the detected edge pixels are mapped into a three dimensional (Hough) space defined by the parameters necessary to represent univocally a specific circle (center coordinates and radius). The analysis of this Hough space is based on the construction of an accumulator array according to a specific voting strategy, which allows us to select circular objects with a number of edge pixels lying on the circumference higher than a threshold. CHT approaches are affected by relatively high memory requirements and computational load, but they can generally achieve an high degree of accuracy, anyway subject to the image quality and to the setup of the involved functional parameters (whose values have to be supplied by the user on the basis of the characteristics of the processed image). Additionally, CHT based approaches are generally affected by a large

number of false positive detections, especially in case of incorrect settings of the input parameters and in the presence of noise.

To overcome these limitations, over the years, different improvements of the original CHT approach have been proposed: size invariant formulation [6], fuzzy HT [7], hypothesis filtering [8], and randomized Hough transform [9,10]. In the randomized CHT iteratively three edge points are randomly chosen to determine the parameters of a candidate circle, and then parameters are collected in the 3D accumulator; candidate circles corresponding to maxima in the accumulator are chosen as detected circles. Maintaining this iterative approach, in different solutions the high memory-demanding accumulator is replaced by less computational expensive voting strategy, in order to select among a set of candidate circles the best one(s). In [11] a LUT-based voting scheme is proposed. In the Randomized Circle Detection (RCD) method [12] at each iteration four edge pixels are selected: three pixels are used to define a possible circle, while the fourth one is used to check if it can be considered as a valid candidate or not; then, a voting process based on counting the number of edge pixels lying on the circumference (namely *inliers*) determines whether the candidate circle is a true circle or not. In [13], an improved RCD method is presented, GRCD-R, where a multiple-evidence-based sampling strategy is used to determine a restricted candidate circles set (starting from the edge pixels extracted by the Sobel operator): a fourth pixel, as in RCD, with additional evidences derived by the gradient of the image is used to check the validity of each candidate circle. Then a refinement stage is performed on the recognized circles, to reduce a bias effect due

\* Corresponding author. Tel.: +39 0832321816; fax: +39 0832420553.

E-mail addresses: [tommaso.demarco@ino.it](mailto:tommaso.demarco@ino.it) (T. De Marco),  
[dario.cazzato@ino.it](mailto:dario.cazzato@ino.it) (D. Cazzato), [marco.leo@ino.it](mailto:marco.leo@ino.it) (M. Leo),  
[cosimo.distante@ino.it](mailto:cosimo.distante@ino.it) (C. Distante).

to the usage of only three edge pixels for the circle parameters' computation.

Different circle detectors have been developed based on other approaches, such as genetic algorithms [14,15] and nature-inspired computing [16]. Recently EDCircles has been presented, a parameter-free circle detection algorithm able to recognize circles and ellipses, which uses an *a contrario* validation step to reduce false positives [17]. EDCircles is based on the analysis of the edge segments (arcs) extracted by an edge segment detector: the Edge Drawing Parameter Free (EDPF) algorithm [18], that does not need parameters supplied by the user (different from traditional edge extractors).

In this work, we propose a randomized circle detection algorithm which uses an alternative multiple-evidence strategy to define valid circles set: maintaining the four edge pixels approach, we add 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 objects detection [19]. Preliminarily, edge pixels with too high or too low local isophote curvature are discarded, then the remaining pixels are classified into subsets with the same isophotes curvature (and consequently belonging to possible candidate circles with the same radius). This leads to three improvements: firstly, the sampling process can be limited on each subset, so increasing the probability to sample edge pixels belonging to the same circle; secondly, candidate circles with radius not compliant with the subset under exam (false positives) can be discarded before the voting process and finally, dependency of the results from the used edge map is reduced. This last feature is particularly relevant considering the aforementioned problem of the input parameters: the proposed work-flow is not parameter free, because a traditional edge extractor is used, but isophotes analysis allows us to obtain similar detection performance starting from edge maps differently detailed or affected by noise. For each candidate circle a kernel density based estimation voting process is performed; this provides better results than simple counting of edge pixels, because *inliers* are automatically defined according to the distribution of the distances between each edge pixel and the circle center. Then, detected circles parameters are refined with an error linear compensation algorithm, in order to provide a better fitting with the recognized circle and the *inliers*.

Each step of the proposed algorithm is described in Section 2, while in Section 3 the obtained results and performance are shown; the method is compared with other circle detectors in the literature using common test images. Finally, achieved conclusions are given in Section 4.

## 2. Algorithm description

The presented algorithm is based on the analysis of the curvature of isophotes, curves connecting pixels in the image with equal intensity. Isophote's properties make them particularly suitable for objects detection and image segmentation, e.g. they have been used for face detection [20], for ridges seeking in CT/MRI scans of the human brain [21], and, recently, for accurate eye center location [22]. In particular, it has been demonstrated that their shapes are independent of rotation and varying lighting conditions, and, in general, isophote features result in better detection performance than intensities, gradients or Haar-like features [19].

### 2.1. Isophotes curvature estimation

Curvature  $\kappa$  of an isophote, which is the reciprocal of the subtended radius  $r$ , can be computed as

$$\kappa = \frac{1}{r} = -\frac{L_x^2 L_{xx} - 2L_x L_{xy} L_y + L_y^2 L_{yy}}{(L_x^2 + L_y^2)^{3/2}} \quad (1)$$

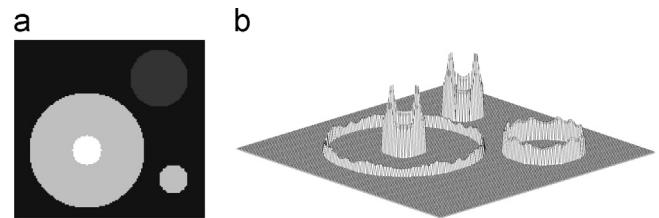


Fig. 1. (a) Original image. (b) 3D view of the isophote curvature at the edges.

where  $\{L_x, L_y\}$  and  $\{L_{xx}, L_{xy}, L_{yy}\}$  are the first- and second-order derivatives of the luminance function  $L(x,y)$  in the  $x$  and  $y$  dimensions respectively (for further details refer to [23]). For our purpose, isophotes curvature is restricted to edge pixels  $V$  extracted using a Canny operator [24], as shown in the sample image in Fig. 1. Restriction of the curvature values on  $V$  is performed using a median filter centered on each edge pixel, in order to reduce the aliasing effect due to the image discretization that otherwise produces a crown-like effect along the edge. This effect can be also reduced by preliminarily applying a Gaussian smoothing to the original image; the convolution of the original image with a Gaussian kernel improves, in fact, isophote curvature estimation, but with high scales, i.e. standard deviations, the important features of the image can be discarded.

The obtained curvature map is then processed by filtering values between a lower-bound  $T_{min}$  and an upper-bound  $T_{max}$  ( $T_{min} < \kappa < T_{max}$ ): this allows us to discard pixels with excessively low curvature (e.g. belonging to lines or in general to circles with radius bigger than the image size), or presenting a too high curvature (i.e. too small circles), shrinking  $V$  only to meaningful pixels. Eventually,  $(T_{min}, T_{max})$  can be chosen in order to limit the search to circles of fixed radius, or to a desired sign of  $\kappa$  (the sign of isophote curvature, in fact, depends on the relative intensity of the outer/inner sides of the curve).

### 2.2. Edge pixels classification

The filtered map of  $\kappa$  values is then analyzed to detect the occurrence of most probable values, as it was a probabilistic distribution. This way to proceed arises from the fact that if a circle is present in the image, then there is an accumulation of edge pixels with the corresponding curvature. Calling  $M$  the unknown number of local maxima in  $\kappa$  distribution, Mean Shift is employed to detect local maxima  $\kappa_i$ ,  $i = 1 : M$ , in  $\kappa$ , assigning at each edge pixel a probability weighted by a 1D Gaussian kernel. The edge map  $V$  is then divided into subsets  $V_i$ , given by pixels with the same isophote curvature  $\kappa_i$ . The basic idea is that edge pixels located on the same circle, or on circles with equal radius, have equal isophote curvature; consequently, in a randomized iterative circle detector, limiting the sampling at each  $V_i$  requires less iterations than sampling between all edge pixels without a specific criteria. Additionally, further evidence constraints can be considered in the analysis of candidate circles, discarding circles with parameters not compliant with the  $V_i$  subset under examination. Referring to Fig. 1, three principal local maxima can be found in  $\kappa$  (two circles have in fact equal radius), corresponding to local modes  $\kappa_i$ ,  $i = 1 : 3$ . Minor local maxima can be observed, due to the discretization aliasing previously introduced, but the relative number of edge points is generally very limited, so they can be easily detected and discarded.

### 2.3. Iterative circle detection

Each  $V_i$  is separately processed by an iterative randomized algorithm: following the four points approach [13], four pixels ( $v_j, v_k, v_l, v_m$ ) are randomly sampled on  $V_i$  at each iteration.

$(v_j, v_k, v_l)$  are used to compute the parameters of a candidate circle  $C_{jkl}$  (center coordinates and radius), and then it is checked if  $v_m$  is relatively close to  $C_{jkl}$ , otherwise the candidate circle is discarded; this evidence check allows us to reduce the number of necessary iterations up to 95% with respect to a three points sampling [13]. This checking can be extended to more than one pixel considering that sampling is restricted on each  $V_i$ , so probability of choosing further points belonging to the circle under examination is higher than sampling on all the edge map.

$C_{jkl}$  radius is compared to the expected  $r_i = 1/\kappa_i$ ; if the two values are not comparable, then  $C_{jkl}$  is discarded, because this means that the sampled edge points belong to different circles presenting equal radius.

Candidate circles found on each  $V_i$  are afterwards analyzed by a voting algorithm, firstly to select the best fitting one and secondly to verify if it lies in the image or not (false positive). This procedure is based on the analysis of distances  $\mathbf{d}_0$  between all edge pixels and center of the candidate  $C_{jkl}$ ; considering  $\mathbf{d}_0$  as a random variable, kernel density estimation  $f(d, \mathbf{d}_0)$  is performed using Parzen windows with Gaussian kernels  $K(\cdot)$  in order to detect the *inliers*:

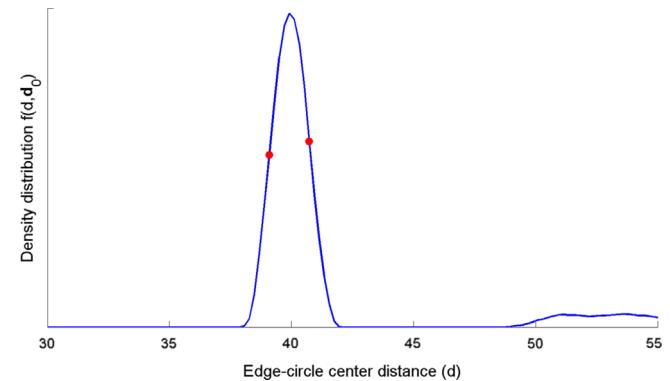
$$f(d, \mathbf{d}_0) = \frac{1}{2\pi r_i N} \sum_{p=1}^N \frac{1}{h_{\mathbf{d}_0}} K\left(\frac{d - d_{0p}}{h_{\mathbf{d}_0}}\right) \quad (2)$$

with  $d$  being the generic pixel-center distance,  $N$  the number of edge pixels and  $h_{\mathbf{d}_0}$  the bandwidth of Parzen windows. Bandwidth estimation is critical in Kernel Density estimation problems since estimating the scale of the inlier noise is an important issue for any robust estimation algorithm, especially for applications where it is hard for the user to provide the scale. Low bandwidth values may allow us to the formation of several local maxima in the density, whereas higher kernel window size will suffer from too little resolution. To this aim, this procedure does not require any parameter, since  $h_{\mathbf{d}_0}$  is computed directly from  $\mathbf{d}_0$  according to the MAD formulation [25]. The MAD formulation has been revisited based on the work in [26], which is purely data driven and provides a good asymptotic convergence property for one-dimensional kernel density estimation where distances are assumed to be homoscedastic; it is defined as follows:

$$h_{\mathbf{d}_0} = N^{-1/5} \text{med}_j |d_{0j} - \text{med}_i d_{0i}|. \quad (3)$$

Although this scale estimate assumes symmetric data relative to the center and that less than half of the data are *outliers*, in [26] this aspect has been proven to be not true, since it is far more robust to *outliers* exceeding 50% in robust estimation. Difference between MAD formulation and the plug-in rule of Eq. (3) is the extra factor  $N^{-1/5}$ , where  $N$  is the number of edge pixels. Other robust scale estimators have been proposed in the literature, such as *Kth Ordered Scale Estimator* (KOSE) and *Adaptive Least Kth order Square* (ALKS) [27] which are generalization of the MAD based method and among the first ones to address the problem of automatic scale estimation.

For each candidate circle, kernel density distribution is computed on  $N_v$  evaluation points, uniformly distributed between the minimum and the maximum distance between edge pixels and the candidate circle center  $C_{jkl}$ . The candidate with the highest absolute maximum of  $f$  is chosen, and pixels around it are taken as lying along its circumference (*inliers*); in particular, *inliers* are selected as the pixels between the peak and the two points where behavior of first derivative of  $f$  changes (Fig. 2). The density  $f$  is normalized to length  $L = 2\pi r_i$  of the expected circumference; this is equivalent to impose that, to obtain the same score, circles with high radius must have more *inliers* than circles with a lower radius. Finally, to avoid false positives, the detected circle is considered to be valid only if the ratio of *inliers* over  $L$  is higher than a threshold  $T_{cov}$ .



**Fig. 2.** Kernel density function  $f(d, \mathbf{d}_0)$  of the distance between the edge pixels and the candidate circle center. When estimate is correct,  $f$  has a peak around the radius value, and points around corresponding to edge pixels lying along its circumference (*inliers*). They are selected by determining the two points (red dots) around the peak where behavior of first derivative of  $f$  changes. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Using kernel density estimation as a voting process allows us to automatically take into account the variability that can occur among different images: in fact, a classic voting strategy requires the definition of threshold(s) to select edge pixels lying near the target circumference. Obviously, it is not possible to adopt a fixed threshold for all the images since the choice of the most appropriate value depends on the intrinsic characteristics of the processed image, for example, a strong threshold can be adopted in the case of circles with well defined borders, whereas a more relaxed one may be successful if circles with blurred contours are present in the image (since, in this last case, edge extraction can lead to irregular borders). Using kernel density estimation as voting strategy allows, instead, us to automatically take into account the distribution of the distances between all the edge points and the circle center, performing a more restrictive selection in the case of well defined borders and a more moderate one in the case of irregular borders.

#### 2.4. Refinement of the results

As previously introduced, the detected circle is affected by a bias effect, because parameters have been estimated only from three pixels: consequently if they are not exactly located on the circumference, the detected circle will not perfectly match the ideal one. In order to reduce this effect, a linear error compensation is used, by adjusting the found parameters to have a better fitting with the recognized *inliers*. Given an ideal circle with center  $(\bar{x}_c, \bar{y}_c)$  and radius  $\bar{R}_c$ , it is defined by the following equation:

$$(x - \bar{x}_c)^2 + (y - \bar{y}_c)^2 - \bar{R}_c^2 = 0. \quad (4)$$

Given the detected parameters  $(x_c, y_c, R_c)$ , affected by errors  $(\delta x_c, \delta y_c, \delta R_c)$ , Eq. (4) becomes

$$(x - (x_c - \delta x_c))^2 + (y - (y_c - \delta y_c))^2 - (R_c - \delta R_c)^2 = 0. \quad (5)$$

Reformulating Eq. (5), we obtain

$$((x - x_c) + \delta x_c)^2 + ((y - y_c) + \delta y_c)^2 - (R_c - \delta R_c)^2 = 0 \quad (6)$$

$$\Rightarrow (x - x_c)^2 + \delta x_c^2 + 2(x - x_c)\delta x_c + (y - y_c)^2 + \delta y_c^2 + 2(y - y_c)\delta y_c - R_c^2 - \delta R_c^2 + 2R_c\delta R_c = 0. \quad (7)$$

Neglecting the second order terms and considering the generic edge pixel coordinates  $(x_i, y_i)$ , we obtain

$$B\delta x_c + C\delta y_c + D\delta R_c = A \quad (8)$$

where

$$\begin{cases} A = -(x_i - x_c)^2 - (y_i - y_c)^2 + R_c^2 \\ B = 2(x_i - x_c) \\ C = 2(y_i - y_c) \\ D = 2R_c. \end{cases} \quad (9)$$

Applying Eq. (8) to all  $N_c$  inliers, the following over-determined linear system is obtained, which is able to provide an evaluation of the parameters errors (to solve this problem a Moore–Penrose pseudo-inverse has been used in our work-flow):

$$\begin{pmatrix} B_0 & C_0 & D_0 \\ B_1 & C_1 & D_1 \\ \vdots & \vdots & \vdots \\ B_{N_c} & C_{N_c} & D_{N_c} \end{pmatrix} \begin{pmatrix} \delta x_c \\ \delta y_c \\ \delta R_c \end{pmatrix} = \begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_{N_c} \end{pmatrix} \quad (10)$$

### 3. Experimental results

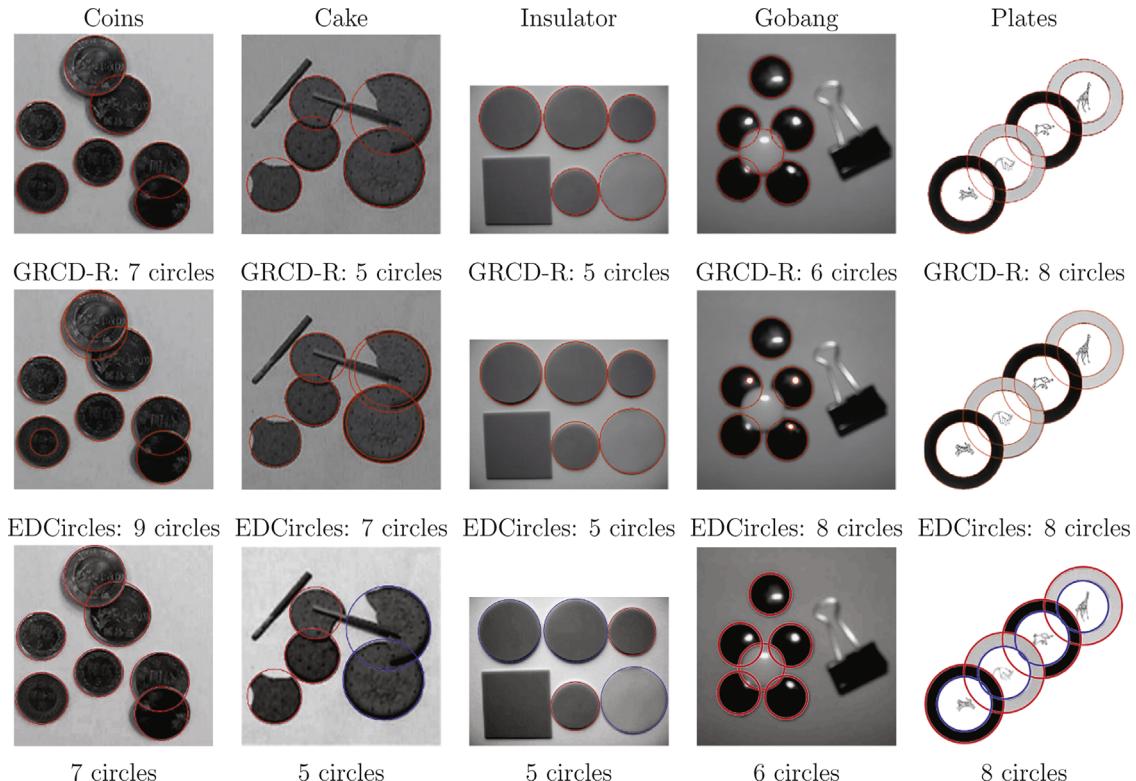
In this section, the results of the application of the proposed method on real images are reported and compared with those provided by some of the leading approaches in the literature. Notice that, concerning the application of the proposed approach, all the involved parameters have been considered as internal, in the sense that they have been set once and then kept constant along all the experiments reported in the following. In particular, the search was limited to those circles whose center lies into the image (if the estimated center is outside the image the circle is immediately discarded). Moreover, only circles with radius bigger than 5 pixels ( $T_{min} = 1/5$ ,  $T_{max} = 2/\text{max\_image\_size}$ ) and with a number of *inliers* larger than 40% of the detected circumference length ( $T_{cov} = 0.4$ ) were taken under consideration. Notice that the choice of the  $T_{cov}$  parameter was made in order to get the best

trade-off between the capabilities to find real circle (i.e. to limit the number of false positive occurrences) and the misdetections in the presence of partial, occluded or less contrasted circles. For example, by using a higher  $T_{cov}$  a more robust circle finder could be built, i.e. false positives would be very rare but, at the same time, a number of misdetections could arise.

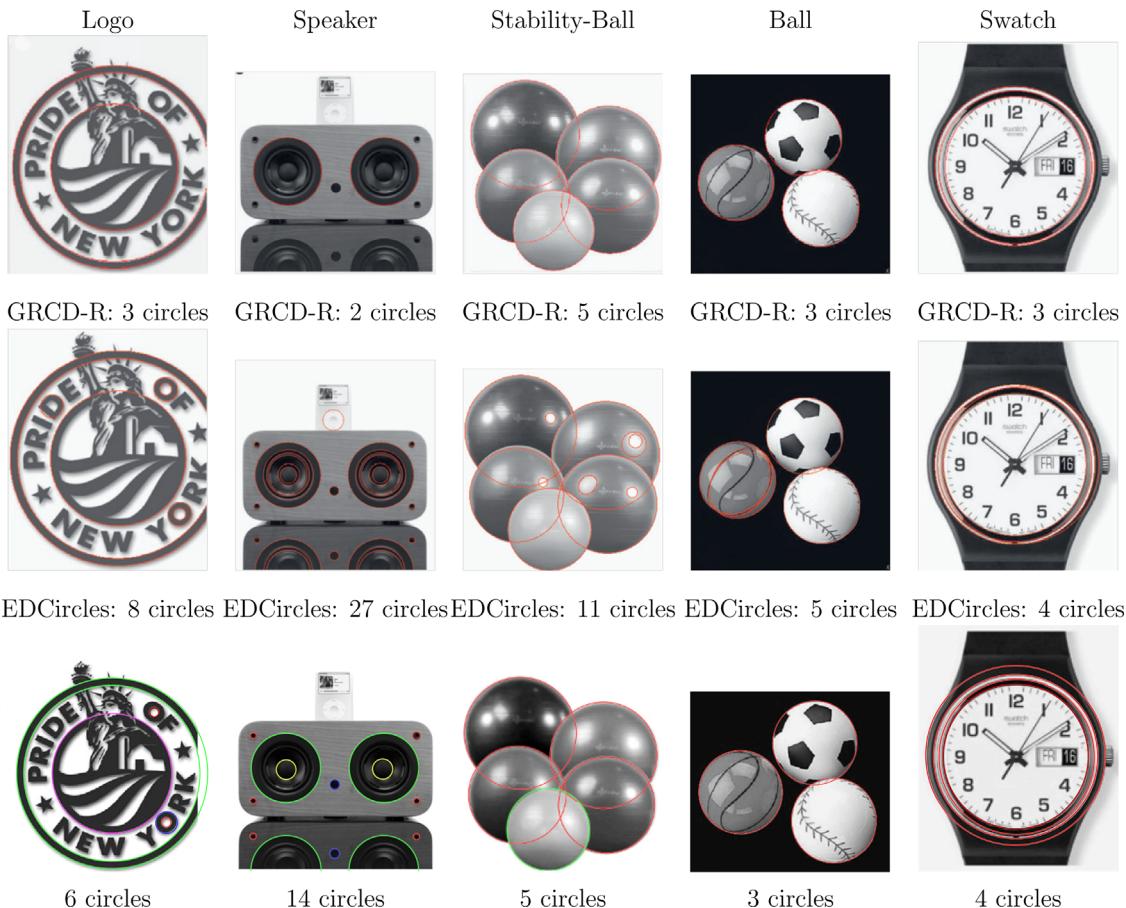
#### 3.1. Detection performance

The proposed method was applied on 10 test pictures extracted from [13]: coins (255 × 256 pixels), ball (340 × 340), plates (400 × 300), cake (256 × 256), stability-ball (209 × 210), gobang (256 × 256), speaker (437 × 393), insulator (256 × 192), logo (283 × 344) and swatch (309 × 356). The aforementioned images were chosen since they became over the years a common reference for testing the circle detection algorithms. The achieved circle detection results were also compared with those obtained by the GRCD-R method [11,13] and by EDCircles, which are the most successful algorithms recently proposed in this research field.

In Figs. 3 and 4 the test images with superimposed detection results obtained by the three comparing techniques (proposed, GRCD-R and EDCircles) are shown; in particular, referring to the results obtained by the proposed isophote based approach, the circles' obtained sampling pixels belonging to the same subset  $V_i$  are drawn with the same color. From figures, it is quite evident that the strength of the proposed method, exploiting the curvature based edge classification, was able to handle the presence of circles with different radii, also concentric, in the same image. In general, the three methods correctly detected most of the prominent circles, even in the presence of occlusions, but there were some instances in which the performance of the comparing approaches was very different. In general, the number of real circles detected by the GRCD-R method was lower than (or at best equal to) the number of circles detected by the proposed and



**Fig. 3.** Detected circles on five test images. (1st row) Circle detection results by GRCD-R, and (2nd row) circle detection results by EDCircles. Results obtained with the proposed method are shown in the 3rd row; in this case circles with the same color are obtained by sampling edge pixels belonging to the same subset of isophote curvature. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)



**Fig. 4.** Detected circles on five test images. (1st row) Circle detection results by GRCD-R, and (2nd row) circle detection results by EDCircles. Results obtained with the proposed method are shown in the 3rd row; in this case circles with the same color are obtained by sampling edge pixels belonging to the same subset of isophote curvature. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

EDCircles approaches. In other words, GRCD-R experienced a considerable number of misdetection, in particular for the ‘Logo’ and ‘Speaker’ images, where the proposed method and EDCircles detected almost all the circles (even having various radius sizes), whereas GRCD-R detected only the biggest ones.

Concerning the comparison between proposed and EDCircles methods, a twofold comment has to be made: on the one side, there were some images (e.g. the ‘Speaker’, ‘Logo’ and ‘Stability-ball’ images) where EDCircles was able to detect more real circles than the proposed approach. This was due to the Canny edge extractor (where a too selective threshold was chosen) which was not able to highlight the less prominent edges in these images. On the other side, EDCircles method highlights a number of false circles (e.g. ‘Logo’, ‘Ball’ and ‘Stability-ball’ images) arisen around noncircular shapes (e.g. the ‘d’ letter in the ‘Logo’ image) or smooth edges caused by reflections (e.g. in the ‘Stability-ball’ image). Moreover, it was also affected by various detections around circles having fuzzy boundaries: in the ‘Coins’ and ‘Cake’ images, in fact, shadows led EDCircles to recognize more circles around the same object.

The above drawbacks were, instead, completely overcome by the proposed method. Noncircular shapes were discarded by the accurate isophote based selection of the edge pixels, and the subsequent voting process, that makes use of Parzen windows on residuals, allowed us to also avoid multiple detections, since it naturally considered shadow edge pixels as *inliers* of the near most prominent circle.

To deeply investigate the effect of the exploitation of the geometrical properties of isophotes to reduce the edge pixel set,

**Table 1**  
Reduction of the edge pixels number in the test images.

Coins	21.2%
Cake	25.5%
Insulator	49.9%
Gobang	40.1%
Plates	21.5%
Logo	53.3%
Speaker	49.3%
Stability-ball	31.3%
Ball	20.7%
Swatch	57.5%
Average:	37.0%

**Table 1** reports the percentages of discarded pixels for each of the considered test images. In particular, from **Table 1**, it is possible to observe that the percentage of reduction depends on the complexity and the structural composition of the image under investigation. The isophotes curvature analysis allowed, in fact, us to discard up to 57% of unnecessary edge pixels in the most complex images, where non-circular objects exist, whereas it led to a minor reduction in all those cases where mainly circular structures are present in the images. In any case, the average reduction along all the considered images was of 37% of the edge pixels, and this is very crucial to better understand the encouraging results achieved by the proposed solution.

Another fundamental strength of the proposed algorithm is the reduction of iterations in the sampling stage, which is achieved by

**Table 2**

Number of possible circles ( $N_p$ ) and candidate circles ( $N_c$ ) supplied by RCD [12], GRCD-R [13] and the proposed isophotes based method.

Image	$N_p$			$N_c$		
	RCD	GRCD-R	Isophotes	RCD	GRCD-R	Isophotes
Coins (a)	49,051	10,810	785	2143	87	52
Ball (b)	29,254	7495	435	1011	27	18
Plates (c)	52,048	16,317	306	1562	58	27
Cake (d)	34,704	12,507	845	1871	85	30
Stability-ball (e)	28,259	7900	274	849	62	22
Gobang (f)	43,968	11,370	3508	2041	29	69
Speaker (g)	29,362	4454	2917	827	16	68
Insulator (h)	35,928	13500	1093	1809	87	56
Logo (i)	33,377	7307	199	930	13	55
Swatch (l)	23,910	6247	627	691	58	71
<b>Average</b>	35,986	9791	1099	1373	52	47

the classification of the remaining edge pixels into different

**Table 3**

Execution time in milliseconds of GRCD-R, EDCircles and the proposed approach for the test images in Figs. 3 and 4. GRCD-R execution time does not include the edge extraction step, while for EDCircles and our approach the whole algorithms were considered.

Image	GRCD-R	EDCircles	Isophotes
Coins	36	8.7	25.4
Cake	31	7.6	2.9
Insulator	35	4.2	5.6
Gobang	48	5.7	4.6
Plates	71	9.8	3.6
Logo	88	12.2	78.6
Speaker	38	15.7	59
Stability-ball	50	9.7	3.3
Ball	38	10.5	15.3
Swatch	86	10.5	7
<b>Average</b>	52	9.9	20.5

subsets of equal isophote curvature. This allows us to restrict the search only to pixels having higher probability to lie along a circle, for example, note that the results in Figs. 3 and 4 were obtained setting the maximum number of iterations  $T_f$  to 1000, whereas GRCD-R uses  $T_f=16,000$  [13]. To numerically prove this important achievement, in Table 2, the number of possible circles  $N_p$  (i.e. the number of circles that the considered algorithm finds) and the number of candidate circles  $N_c$  (i.e. the number of circles that the considered algorithm actually retains before the voting process) supplied by RCD [12], GRCD-R and the proposed isophotes based method are reported. Table 2 demonstrates that, by using the proposed method,  $N_p$  became much more smaller than by using the comparing randomized approaches, i.e. GRCD-R and RCD. Moreover, it is possible to observe that both isophotes based approach and GRCD-R provided a more significative reduction of the candidate circles  $N_c$  than RCD, even if this pleasant achievement is much more computationally expensive for GRCD-R, since it must handle a bigger initial  $N_p$ . This is better demonstrated in Table 3, where the execution time for GRCD-R, EDCircles and the proposed approach is shown (side by side for each of the ten test images). The proposed method was run on a 2 GHz Intel Core i7-3537U CPU. For GRCD-R and EDCircles, instead, the running times reported by the authors and obtained on a 3 GHz Intel E8400 CPU and a 2.2 GHz Intel E4500 CPU, respectively, were taken. From Table 3 it is possible to derive that the proposed approach was less time consuming than GRCD-R having, on average, a speedup factor of  $2.5\times$ . Additionally, for GRCD-R, execution time required to perform edge extraction was not

**Table 4**

Percentage distribution of the execution time reported in Table 3 on the stages of the proposed approach for the test images in Figs. 3 and 4. In the 'Curvature computation' field the median filtering of the curvature distribution is included.

Image	Edge detection	Curvature computation	Mean Shift	Iterative detection	Results refinement
Coins	5.4	2.2	30.0	58.5	4.0
Cake	6.4	6.0	43.1	38.1	6.4
Insulator	5.7	4.5	46.3	39.8	3.7
Gobang	5.8	3.1	18.3	70.1	2.8
Plates	6.2	7.6	59.1	21.0	6.1
Logo	5.6	2.3	55.6	30.8	5.7
Speaker	5.2	1.2	12.7	78.6	2.3
Stability-Ball	6.1	6.3	46.2	35.9	5.4
Ball	5.5	3.4	50.6	33.3	7.1
Swatch	5.2	3.4	25.6	60.8	5.0
<b>Average (%)</b>	5.7	4.0	38.8	46.7	4.8

considered, and thus the actual difference between the proposed approach and GRCD-R was greater than the one derivable from Table 3. These results further prove that the isophote based classification of the edge pixel set allows, in a randomized circle detector, us to improve its computational performance. On the other side, from the same table, we can derive that the isophote based approach was, on average, slower than EDCircles. In particular, with a more detailed analysis, it can be observed that our approach was particularly limited on the 'Logo' and 'Speaker' images, while the execution time spent to process the remaining images was comparable with that one required by EDCircles (sometimes the proposed approach resulted faster than EDCircles, such as in the 'Swatch', 'Stability-Ball' and 'Plates' case). This was due to the complexity of the 'Logo' and 'Speaker' images, where a large number of edge pixels made the Mean Shift analysis and the iterative step particularly time consuming; this is clearly shown in Table 4 where, for each image, the percentage distribution of the total execution time on the involved algorithmic steps is reported. For example, in the 'Logo' image the Mean Shift analysis and the subsequent iterative detection spent almost the 87% of the total execution time, while in the 'Speaker' image the 78% of the total time was spent for the iterative circle detection stage. In general, from the last row in Table 4, we can derive that, in the proposed method, the iterative circle detection and the Mean Shift analysis were the most time consuming steps (on average they represent the 46.7% and 38.8% of the required execution time respectively), while the other stages of the proposed method (edge extraction, curvature computation and refinement of the results) occupied overall 15% of the total execution time.

### 3.2. Accuracy of the estimated parameters

In Fig. 5 the biasing effect due to the adopted sampling strategy is demonstrated; without any compensation all circles were correctly detected (they are drawn in red), but the accuracy in the estimation of their parameters decreased in the case of lower luminance. In the same figure, the advantages of using the linear compensation (described in Section 2.4) are demonstrated by drawing the refined circles (represented by dotted blue lines); a better fit with the detected *inlier* pixels was obtained allowing an improvement of the estimated parameters up to 3% in the test images.

To quantitatively assess the accuracy of the proposed method, the Hough transform based circle detector (CHT) was taken as a reference [2], and used to process the 10 test images. As in the EDCircles and GRCD-R original papers, to make CHT results as accurate as possible the granularity of the x-coordinate, y-coordinate

and radius was set to one pixel precision. Since CHT needs many parameters supplied by the user, it was impossible to use the same parameters for all the test images. So, to obtain a valid ground-truth, CHT was applied several times on each image using different parameters and, this way, the best parameter configuration was determined for each image. Additionally, the large amount of false detections supplied by CHT were discarded. The average differences of the center coordinates ( $\Delta a, \Delta b$ ) and radii  $\Delta r$  between circles detected by CHT and those detected by the proposed approach, EDCircles and GRCD-R are shown in [Table 5](#). All the three methods demonstrated a sub-pixel accuracy; in particular the proposed solver was more accurate than EDCircles, but less than GRCD-R. In the above analysis, we considered CHT results as ground-truth, mainly to compare our method with the GRCD-R and EDCircles; however it is necessary to consider that, in some case, CHT results are not necessarily the optimal ones (traditional CHT, for example, can provide only integer radius values). To perform a more detailed numerical evaluation, an additional experiment was performed using the Atherton improved CHT formulation (MCHT) [6], which is radius size invariant and more tolerant to noise than traditional

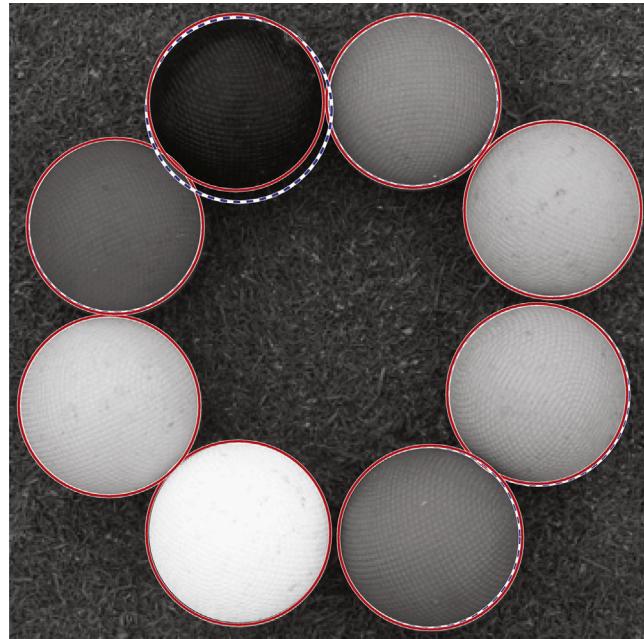
CHT. Following the same trial-and-error procedure used for CHT, with various parameter sets for each image and discarding false positives, we obtained more accurate ground-truth. In the last two columns of [Table 5](#), the average differences of the parameters detected by the proposed method and MCHT are shown; it is evident that average errors in the proposed approach were the lowest ones, in particular for the radius value, whereas the accuracy of EDCircles was not optimal.

### 3.3. Noise analysis

In order to evaluate the robustness of the proposed method, a set of simulations using different noisy images was performed. In this analysis, we decided to follow two different ways: in the first experiment, we added different levels of noise to the edge map of the 'Plates' image while, in the second set of simulations, different amounts of Gaussian noise were applied directly to the 'Plates' original image. This way, the robustness of the sampling strategy supported by the isophotes analysis (independent of the used edge extractor) and of the overall algorithm separately can be evaluated.

In [Fig. 6](#), the detection results of the first analysis are shown, i.e. in the case of uniformly distributed false edge pixels added to the binary edge map. The amount of added pixels was one, two and three times the number of real edge pixels. For each configuration 100 runs were performed, by increasing the maximum number of iterations for circle  $T_f$  to 3000. All the circles were correctly detected, providing results similar to the noise-free case; this was due to the isophotes based classification of edge pixels: most of the added pixels, in fact, corresponded to non-compliant isophote curvature values, and they were discarded before the edge sampling stage. The method maintained a sub-pixel accuracy, as shown in [Table 6](#), where detected parameters are compared with the parameters provided by CHT and MCHT in the noise free image.

A further set of parametric simulations was performed in order to evaluate the tolerance of the overall work-flow if the original image is directly affected by noise. [Fig. 7](#) demonstrates the performance when Gaussian white noise was applied. Noise was added by using the MATLAB function `imnoise(img, 'gaussian', mean, variance)` with zero mean and variance ranging from 0.01 (1% added noise) to 0.3 (30% added noise). The increasing noise led to a reduction of detected circles, especially for those having a lower gradient along the circumference. It is interesting to note that the average number of false positives was zero up to 20% of added noise: in order to obtain at least an invalid circle (on average) for each processed image, it was necessary to apply more than 30% of noise.

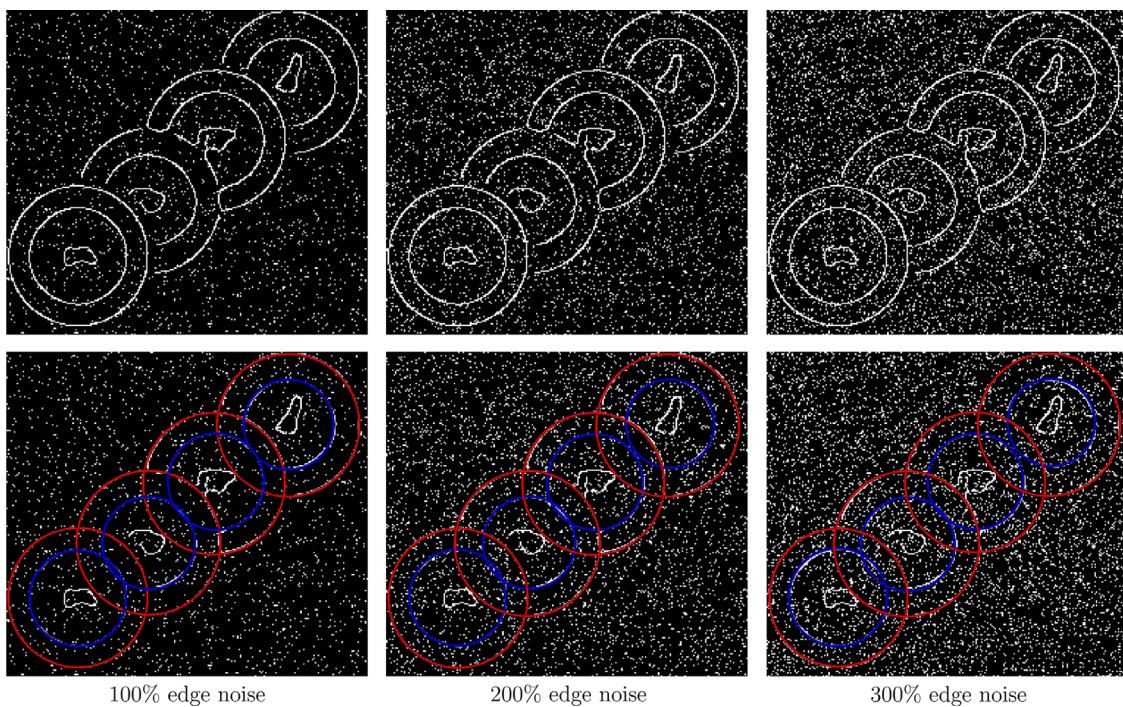


**Fig. 5.** Detected circles with (red) and without (dotted blue) bias compensation. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

**Table 5**

Average differences on the parameters detected by traditional CHT and GRCD-R, EDCircles and our approach. In the last two columns are shown the average differences between our method and EDCircles versus an improved Modified CHT[6].

Image	GRCD-R		EDC		Isophotes		EDCircles-MCHT		Isophotes-MCHT	
	$(\Delta a, \Delta b)$	$\Delta r$								
Coins	(0.55, 0.34)	0.47	(0.81, 1.01)	0.66	(0.96, 0.61)	0.77	(1.16, 0.90)	0.48	(0.88, 0.75)	0.64
Ball	(0.28, 0.35)	0.50	(0.66, 1.10)	0.85	(0.20, 0.49)	0.48	(0.63, 0.55)	0.30	(0.07, 0.60)	0.27
Plates	(0.50, 0.54)	0.52	(0.43, 0.26)	0.62	(0.41, 0.50)	0.34	(1.06, 1.02)	0.19	(0.35, 0.56)	0.06
Cake	(0.43, 0.67)	0.46	(0.83, 0.75)	1.18	(0.48, 0.28)	0.76	(1.04, 1.09)	0.57	(0.46, 0.49)	0.36
Stability-ball	(0.42, 0.65)	0.51	(0.75, 1.02)	0.84	(0.65, 0.89)	0.64	(0.89, 1.29)	1.06	(0.63, 0.66)	0.18
Gobang	(0.38, 0.43)	0.50	(0.45, 0.33)	0.25	(0.45, 0.54)	0.45	(1.12, 1.02)	0.37	(0.37, 0.57)	0.40
Speaker	(0.26, 0.01)	0.46	(0.59, 0.47)	0.66	(0.34, 0.64)	0.59	(0.91, 0.68)	0.14	(0.21, 0.55)	0.41
Insulator	(0.31, 0.57)	0.45	(0.50, 1.13)	0.70	(0.57, 0.90)	0.37	(1.17, 0.95)	0.36	(0.75, 0.87)	0.46
Logo	(0.02, 0.40)	0.06	(0.61, 0.57)	0.49	(1.05, 1.00)	0.61	(1.05, 0.63)	1.52	(0.35, 0.21)	0.45
Swatch	(0.41, 0.63)	0.71	(0.65, 0.64)	0.68	(0.38, 0.16)	0.44	(0.50, 1.27)	1.46	(0.04, 0.03)	0.65
<b>Average</b>	(0.36, 0.46)	0.46	(0.63, 0.73)	0.66	(0.55, 0.60)	0.54	(0.95, 0.94)	0.65	(0.41, 0.53)	0.39



**Fig. 6.** Detected circles with noisy edge maps of the 'Plates' image.

**Table 6**

Average differences on the parameters estimated by our approach versus traditional CHT and MCHT. Data are referred to noisy edge maps obtained adding uniformly distributed random edge pixels. Number of pixels added is one, two and three times the number of true edge pixels.

Edge noise ratio (%)	CHT		MCHT	
	$(\Delta a, \Delta b)$	$\Delta r$	$(\Delta a, \Delta b)$	$\Delta r$
100	(0.85, 0.50)	0.63	(0.81, 0.57)	0.50
200	(0.93, 0.57)	0.65	(0.89, 0.52)	0.48
300	(0.87, 0.64)	0.58	(0.86, 0.51)	0.47

#### 3.4. On the used edge map

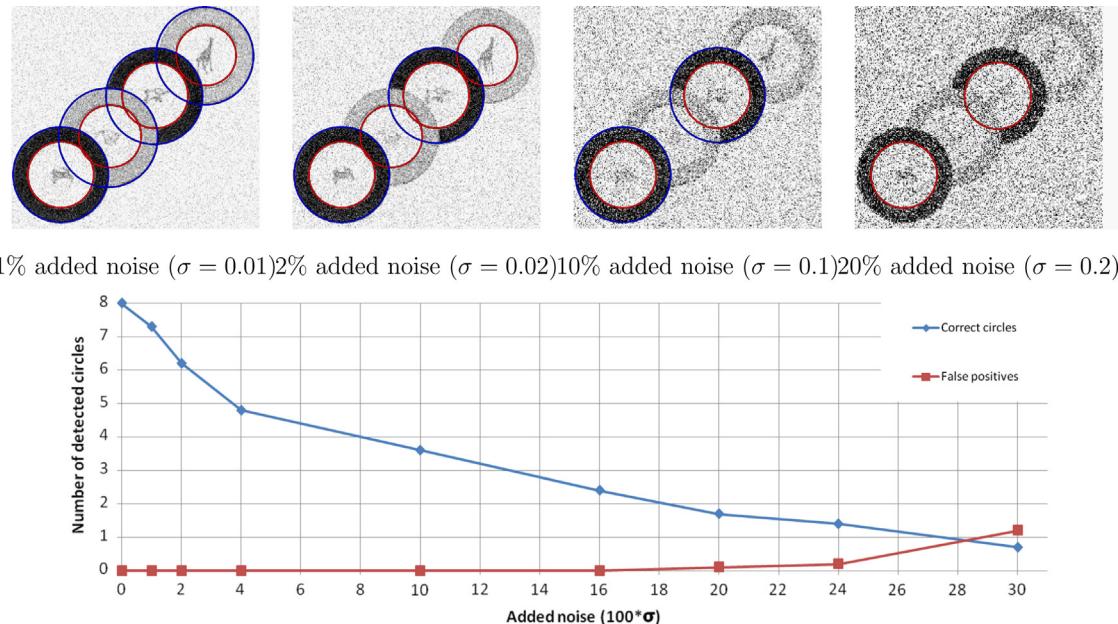
As for the GRCD-R, the proposed approach is performed after a preliminary edges extraction step. This point is particularly important, because, in a randomized algorithm, the chosen edge extractor affects directly the performance of the overall work-flow: a highly selective edge extractor can discard less prominent edges, which consequently will not be considered in the effective circle detection step. On the contrary, too weak edge extraction can lead the user to increment the maximum number of iterations in order to fix the reduced probability of sampling edge pixels lying on valid circles. EDCircles uses ad hoc parameters free edge extractor (EDPF) [18], which detects arcs in the image; however, as the same authors stated, it is penalized in case of fuzzy images and small divided circles. In these challenging contexts, the proposed isophotes based approach is instead able to reduce the dependency to the edge extraction: in fact, as previously explained, isophotes curvature analysis allows us to improve the probability at each iteration to sample meaningful edge pixels. To prove this feature, two additional test images used in [17] and in which the EDCircles was unable to detect any circle are taken into account: the 'Drain cover' and 'Blood cells'. On these images, three different experimental tests are performed, consisting in executing the proposed circle detection giving as input three different edge maps: two of them were obtained using Canny edge extractor with different

setting parameters, while the third one was extracted by EDPF algorithm. In Fig. 8 circles detected when different edge maps are used are reported; in the first column the two considered test images are shown. The corresponding edge maps extracted by Canny algorithm with different thresholds are shown in the second and fourth columns of the first two rows while, in the third and fifth columns, the relative detected circles are presented. Similarly, in the third row edge maps extracted by EDPF algorithm and the circles detected by our algorithm using these EDPF edge maps are shown. Note that the proposed method detected almost all circles in the images and its performance appeared to be quite independent of the used edge map. This is an important feature since, in the proposed approach, the only parameters to be supplied by the user are those relative to the edge extractor. This experiment highlights that isophotes curvature classification limits the influence of these parameters allowing an easier usage of the proposed circle detector, which can be then considered to be parameters free. It is finally important to note that EDPF is a parameter-free edge detector, so circles obtained using the EDPF edge map did not require any parameter to be supplied by the users.

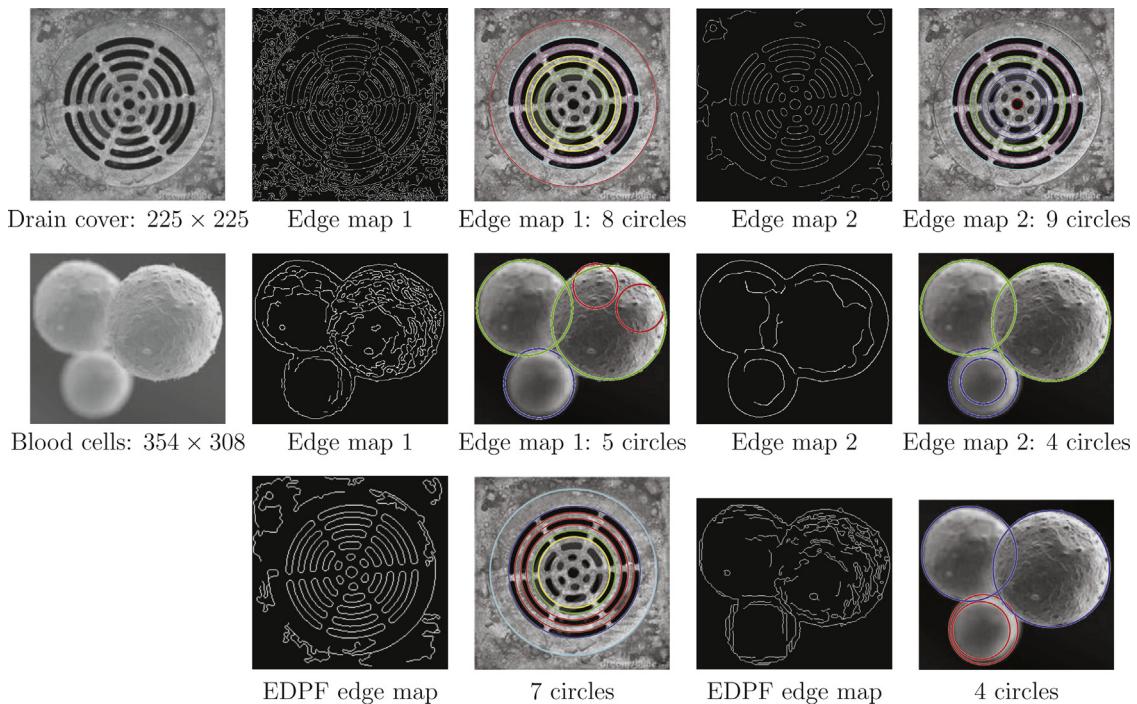
A final consideration is related to the 'Blood cells' image: the occurrence of objects with fuzzy or smoothed borders deceived the algorithm causing false positive detections; however the algorithm was still able to detect valid circles. This is caused both by the limits of the used edge extractors and by the increase of isophote lines in smoothed regions. Also in this case, detection results obtained with Canny edge extractor and EDPF were similar.

#### 3.5. Computational remarks

About computational complexity of the proposed solution, the overall system complexity can be derived considering each single stage that composes the algorithm. Given an image of size  $m \times n$ , Canny edge extraction has asymptotic complexity of  $O(mn \log(mn))$ . The smoothing operation, necessary to compute the isophotes curvature distribution, makes use of a Gaussian filter, therefore the asymptotic complexity is  $O(mn)$ . Median filtering, necessary to restrict the curvature values on the  $N_e$  edge pixel, has a complexity of



**Fig. 7.** Detected circles in 'Plates' image with added Gaussian noise with zero mean and variance  $\sigma$ . In first row detected circles in the image affected by a noise of 1%, 2%, 10% and 20% are shown. In the second row the behavior of the average number of detected circles and false positives with noise varying from 1 to 30% is represented; values were obtained performing 100 simulations for each considered level of noise.



**Fig. 8.** Detected circles with different edge maps; in the first column test images are shown. Referring to the first and second rows, in the second and fourth columns edge maps extracted by Canny algorithm with different thresholds are shown, while in the third and fifth columns detected circles corresponding to the different edge maps are presented. Similarly, in the third row edge maps extracted by EDPF algorithm and the circles detected by our algorithm using these EDPF edge maps are shown.

$O(N_m^2 N_e) \approx O(N_e)$ , since the size of the kernel window,  $N_m$ , is negligible.

The remarkable part of the algorithm is the Mean Shift and the iterative circle detection, as can be observed in Table 4. Mean Shift analysis can be summarized as following: firstly an edge pixel is randomly selected, then an iterative procedure is performed starting from its curvature value  $\kappa_{in}$ . This procedure allows us to determine the belonging cluster  $V_i$  and the relative mode  $\kappa_i$ , representing the curvature of the cluster. Then all edge pixels with a curvature value between  $\kappa_{in}$  and  $\kappa_i$  are selected as belonging

to  $V_i$ . The computational cost of this procedure is  $O(N_e N_{max})$ , where  $N_{max}$  is the imposed maximum number of iterations. This procedure is repeated until all edge pixels have been classified. Consequently, in a worst case analysis where  $N_e$  modes are present, Mean Shift has a complexity of  $O(N_e^2 N_{max})$ ; note that this case is highly unlikely, since it represents an unreachable boundary case.

Iterative circle detection is applied on each cluster  $V_i$ ; the relative computational cost is dominated by the kernel density estimation processes. In particular, to detect a circle, a maximum number of iterations  $T_f$  is performed; however only a little number

of candidate circle are selected, as shown in [Table 2](#). For each candidate circle, density estimation is performed, with a  $O(GN_e)$  complexity, where  $G$  is an internal parameter of the method. Finally, the best candidate circle is chosen, and the refinement process is performed, which has a complexity of  $O(4N_{inliers}) \approx O(N_{inliers})$ , with  $N_{inliers}$  being the number of the *inliers* of the detected circle.

#### 4. Conclusions

The paper introduces a novel iterative algorithm for circle detection, which uses both local information related to the concept of isophotes curvature and a kernel density estimation to robustly detect the *inliers* of each structure describing a real circle. Results are compared with the classical Hough based solvers and the latest circle detectors in the literature. Referring to randomized circle detectors as GRCD-R, the proposed approach provides a reduction of the necessary computational power given by the reduced number of necessary iterations, resulting in two time faster than GRCD-R on a set of test images, and maintaining a sub-pixel accuracy in the circle parameters estimation, even in the presence of noise in both the image and the edge map. With respect to the techniques not based on the randomized circle detection, the proposed approach has been compared with EDCircles, a parameter-free solution based on the edge segments. The proposed approach resulted in a higher estimated parameters' accuracy, although slower than EDCircles – however, maintaining execution times of the same order.

It has been proven that the proposed algorithm maintains detection performance in the presence of different edge maps of the same image, limiting the influence of the edge extractor on the results and so allowing a simpler usage for automated applications. It is interesting, to this end, to highlight how the method provides similar results using edge maps extracted by Canny extractor and EDPF, the parameter-free segment detector used in EDCircles, giving correct detections also with test images where EDCircles fails. In conclusion, the proposed approach can be used exploiting the accuracy of the estimated parameters, the robustness in the presence of noise and the limited dependency from the used edge extractor, eventually in a multi-core implementation able to exploit the level of parallelism achievable during the iterative step. As shown, the iterative step represents the more time consuming stage of all the computation; so, considering that each iteration is independent of the others, a parallel execution can lead to a considerable reduction of the execution times.

#### Conflict of interest

None declared.

#### References

- [1] E. Davies, *Machine Vision: Theory, Algorithms, practicalities*, Morgan Kaufmann, Burlington, Massachusetts, 2004.
- [2] P. Hough, Method and Means for Recognizing Complex Patterns, US Patent 3,069,654, 1962.
- [3] R.O. Duda, P.E. Hart, Use of the hough transformation to detect lines and curves in pictures, *Commun. ACM* 15 (1972) 11–15.
- [4] E. Davies, A modified hough scheme for general circle location, *Pattern Recognit. Lett.* 7 (1988) 37–43.
- [5] P. Kierkegaard, A method for detection of circular arcs based on the hough transform, *Mach. Vis. Appl.* 5 (1992) 249–263.
- [6] T.J. Atherton, D.J. Kerbyson, Size invariant circle detection, *Image Vis. Comput.* 17 (1999) 795–803.
- [7] J.H. Han, L. Kóczy, T. Poston, Fuzzy hough transform, *Pattern Recognit. Lett.* 15 (1994) 649–658.
- [8] W. Lam, S. Yuen, Efficient technique for circle detection using hypothesis filtering and hough transform, in: *IEEE Proceedings on Vision, Image and Signal Processing*, vol. 143, IET, Stevenage, England, 1996, pp. 292–300.
- [9] L. Xu, E. Oja, P. Kultanen, A new curve detection method: randomized Hough transform (RHT), *Pattern Recognit. Lett.* 11 (1990) 331–338.
- [10] L. Xu, E. Oja, Randomized Hough transform (RHT): basic mechanisms, and computational complexities, *CVGIP: Image Underst.* 57 (1993) 131–154.
- [11] K. Chung, Y. Huang, Speed up the computation of randomized algorithms for detecting lines, and ellipses using novel tuning- and LUT-based voting platform, *Appl. Math. Comput.* 190 (2007) 132–149.
- [12] T. Chen, K. Chung, An efficient randomized algorithm for detecting circles, *Comput. Vis. Image Underst.* 83 (2001) 172–191.
- [13] K. Chung, Y. Huang, S. Shen, A. Krylov, D. Yurin, E. Semeikina, Efficient sampling strategy and refinement strategy for randomized circle detection, *Pattern Recognit.* 45 (2012) 252–263.
- [14] P.-Y. Yin, A new circle/ellipse detector using genetic algorithms, *Pattern Recognit. Lett.* 20 (1999) 731–740.
- [15] V. Ayala-Ramirez, C.H. García-Capulin, A. Perez-Garcia, R.E. Sanchez-Yanez, Circle detection on images using genetic algorithms, *Pattern Recognit. Lett.* 27 (2006) 652–657.
- [16] E. Cuevas, D. Oliva, D. Zaldivar, M. Pérez-Cisneros, H. Sossa, Circle detection using electro-magnetism optimization, *Inf. Sci.* 182 (2012) 40–55.
- [17] C. Akinlar, C. Topal, Edcircles: a real-time circle detector with a false detection control, *Pattern Recognit.* 46 (2013) 725–740.
- [18] C. Akinlar, C. Topal, Edpf: a real-time parameter-free edge segment detector with a false detection control, *Int. J. Pattern Recognit. Artif. Intell.* 26 (2012).
- [19] J. Lichtenauer, E. Hendriks, M. Reinders, Isophote properties as features for object detection, in: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, vol. 2, 2005, pp. 649–654.
- [20] B. Froba, C. Kublbeck, Robust face detection at video frame rate based on edge orientation features, in: *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition*, Washington D.C., New York, 2002, pp. 342–347.
- [21] J.B.A. Maintz, P.A. van den Elsen, M.A. Viergever, Evaluation of ridge seeking operators for multimodality medical image matching, *IEEE Trans. Pattern Anal. Mach. Intell.* 18 (1996) 353–365.
- [22] R. Valenti, T. Gevers, Accurate eye center location through invariant isocentric patterns, *IEEE Trans. Pattern Anal. Mach. Intell.* 34 (2012) 1785–1798.
- [23] M. Van Ginkel, J. Van de Weijer, L. Van Vliet, P. Verbeek, Curvature estimation from orientation fields, in: *Proceedings of the Scandinavian Conference on Image Analysis*, vol. 2, 1999, pp. 545–552.
- [24] J. Canny, A computational approach to edge detection, *IEEE Trans. Pattern Anal. Mach. Intell.* 8 (1986) 679–698.
- [25] P. Rousseeuw, C. Croux, Alternatives to the median absolute deviation, *J. Am. Stat. Assoc.* 88 (1993) 1273–1283.
- [26] C. Haifeng, P. Meer, Robust regression with projection based m-estimators, in: *Proceedings of the Ninth IEEE International Conference on Computer Vision*, Nice, France, vol. 2, New York, 2003 pp. 878–885.
- [27] K.-M. Lee, P. Meer, R.-H. Park, Robust adaptive segmentation of range images, *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (1998) 200–205.

**Tommaso De Marco** received the M.Sc. and Ph.D. degrees in Electrical Engineering from the University of Bologna, Italy, in 2006 and 2010, respectively. In 2007, he worked as an independent consultant for STMicroelectronics and then as a researcher for the Advanced Research Center on Electronic Systems “E. De Castro” of the University of Bologna, Italy. Currently, he is with the National Research Council - Institute of Optics (CNR-INO), Lecce, Italy. His research interests include image processing, pattern recognition, biomedical imaging and parallel computing.

**Dario Cazzato** received the master's degree in Computer Engineering from the University of Salento, Lecce, Italy, in 2011. From May 2012 to May 2013 he has been a Research Associate with the Institute of Optics (INO-CNR) in Lecce, Italy. Since June 2013 he is a Ph.D. student in Information Engineering at the University of Salento. He is responsible for the SARACEN project under social innovation funding of the Italian Minister MIUR. His fields of expertise are in computer vision and pattern recognition, particularly for visual human–robot and human–computer interaction.

**Marco Leo** received the Honors degree in Computer Science Engineering from the University of Salento, Lecce, Italy, in 2001. From May 2001 to August 2012, he was a Researcher with the Institute of Intelligent Systems for Automation (ISSIA-CNR) in Bari, Italy. Since September 2012 he has been a Researcher with the Institute of Optics (INO-CNR) in Lecce, Italy. His main research interests are in the fields of image processing, image analysis, computer vision, pattern recognition, digital signal processing,

neural networks, graphical models, wavelet transform, and independent component analysis. He participated in a number of national and international research projects facing automatic video surveillance of indoor and outdoor environments, human attention monitoring, real-time event detection in sport contexts, and nondestructive inspection of aircraft components. He is the author of more than 100 papers in national and international journals and conference proceedings. He is also a coauthor of three international patents on visual systems for event detection in sport contexts.

**Cosimo Distante** received the degree in Computer Science, in 1997, at the University of Bari. He received the Ph.D. in Engineering, in 2001, by working in the field of artificial intelligent systems. He has been a Visiting Researcher at the Computer Science Department of the University of Massachusetts (Umass at Amherst, MA) where has worked in the context of robot learning. In 1998, he served as a Teaching Assistant of the Artificial Intelligence course of the Master in Science at the University of Massachusetts. In 2000, he received the Ph.D. in Engineering from the University of Salento. Since 2003 he is a Contract Professor for the courses of Pattern Recognition and Image Processing in Computer Engineering at the University of Salento. He is the founder of Taggalo startup company which develops audience metrics for Digital Signage and a patented proof-of-Display technology. In 2011 and 2012 he has been awarded with the National Innovation Prize from the Senate President of the Republic of Italy. He is the expert member for the panel in Innovation Technology of the Ministry of the Economic Development. He is currently with the National Institute of Optics of the CNR where his main research interests are in the field of computer vision and pattern recognition applied to the following sectors: video surveillance, digital signage, robust estimation, medical imaging and manufacturing.