

# Soccer Ball Detection with Isophotes Curvature Analysis

Tommaso De Marco, Marco Leo, and Cosimo Distante

CNR - INO. Istituto Nazionale di Ottica, via della Libertà 3, Arnesano (LE)

{[tommaso.demarco](mailto:tommaso.demarco),[marco.leo](mailto:marco.leo),[cosimo.distante](mailto:cosimo.distante)}@ino.it

<http://www.ino.it>

**Abstract.** Circle detection is a critical issue in image analysis: it is undoubtedly a fundamental step in different application contexts, among them one of the most challenging is the detection of the ball in soccer game. Hough Transform based circle detector are largely used but there is a large open research area that attempt to provide more effective and less computationally expensive solutions based on randomized approaches, i.e. based on iterative sampling of the edge pixels. To this end, this work presents an ad-hoc randomized iterative work-flow, which exploits geometrical properties of isophotes, the curvature, to identify edge pixels belonging to the ball boundaries; this allow to consider a large amount of edge pixels, but limiting most of the time-consuming computation only on a restricted subset given by pixels with an high probability to lie on a circular structure. The method, coupled with a background suppression algorithm, has been applied to a set of real images acquired by fixed camera providing performances higher than a standard circular Hough transform solver, with a detection rate  $> 86\%$ .

**Keywords:** Ball detection, Isophote, Image sampling.

## 1 Introduction

Circle detection is a critical issue in pattern recognition and computer vision [1], as in the automatic sport video analysis. In particular in the soccer game circle detection is necessary in order to detect and localize the ball on each video frame. This is a critical task, and, although different techniques have been proposed, further solutions need to be investigated in order to achieve the accuracy required for a real usage during the matches [2]. The most powerful methods in literature address this problem dividing the computation in two phases: firstly, on each frame the regions with an high probability to contain the ball are selected (*ball candidate extraction*), secondly, selected regions are processed to discover the real presence of the ball (*ball candidate validation*) [3]. Candidate ball extraction is performed with circle detection methods, as the circular Hough transform (CHT) or randomized techniques. Randomized approaches are particularly interesting in this context because they reduce the computational power required by Hough based approaches, and in addition make relatively easy to add further

constraints and validity checks to increase the detection performances. In a randomized approach three random edge points are iteratively chosen to determine the parameters of a candidate circle, then the best one(s) is selected according to a voting strategy. However, before the voting process, a series of validation checks are necessary to discard possible invalid circles: in the Randomized Circle Detection (RCD) method [4] 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 a valid candidate or not; then an evidence-collecting process is performed to select best circle(s) among the candidates, without using any accumulator.

In this work we propose an alternative multiple-evidence strategy to discard invalid circles: 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 [5]. Analysis of isophote curvature allows to select pixels with a curvature compliant with the ball radius, and consequently limiting the sampling process only to edge pixel with an high probability to lie on the ball circumference. A voting process, based on kernel density estimation, is performed for each candidate circle, determining if it is a real circle or not. Then, results are refined with an error linear compensation algorithm, in order to provide a better fitting with the edge points lying on the recognized circle.

## 2 Algorithm Overview

The proposed algorithm is based on analysis of the curvature of isophotes, curves connecting pixels in the image with equal intensity. Isophotes curvature is computed for each pixel in the image, then data are restricted to edge pixels extracted using a Canny operator [6]; accuracy of edge extraction is improved performing on each frame an adaptive background suppression [7].

Obtained curvature distribution is analyzed to detect the occurrence of most probable values, providing a clusterization of the edge pixels into subsets of equal curvature; then subset(s) with curvature compliant with the expected ball radius are selected.

Each subset is separately processed by an iterative randomized algorithm; candidate circles parameters are computed by randomly sampling three pixels at each iteration, then invalid circles are discarded by performing a series of evidence checks.

Best circles are then selected between all the candidates by a voting algorithm, and, finally a linear error compensation process is applied to improve accuracy of the recognized circles.

### 2.1 Adaptive Background Subtraction

At first all the moving regions are detected making use of a background subtraction algorithm. The procedure consists of a number of steps. At the beginning of

the image acquisition a background model has to be generated and later continuously updated to include lighting variations in the model. Then, a background subtraction algorithm distinguishes moving points from static ones. Finally, a connected components analysis detects the blobs in the image.

The implemented algorithm uses the mean and standard deviation to give a statistical model of the background. Formally, for each frame  $I(x, y)$  at the time  $t$  the algorithm evaluates an average  $\mu^t(x, y)$  and a standard deviation  $\sigma^t(x, y)$  image:

$$\mu^t(x, y) = \alpha I^t(x, y) + (1 - \alpha)\mu^{t-1}(x, y) \quad (1)$$

$$\sigma^t(x, y) = \alpha |I^t(x, y) - \mu^t(x, y)| + (1 - \alpha)\sigma^{t-1}(x, y) \quad (2)$$

It should be noted that (2) is not the correct statistical evaluation of standard deviation, but it represents a good approximation of it, allowing a simpler and faster incremental algorithm that works in real time.

The background model described above is the starting point of the motion detection step. The current image is compared to the reference model, and points that differ from the model by at least two times the correspondent standard deviation are marked. Formally, the resulting motion image can be described as:

$$M(x, y) = \begin{cases} 1 & \text{if } |I(x, y) - \mu^t(x, y)| > 2 \cdot \sigma^t(x, y) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where  $M(x, y)$  is the binary output of the subtraction procedure. An updating procedure is necessary to have a consistent reference image at each frame, a requirement of all motion detection approaches based on background. The particular context of application imposed some constraints. First of all, it is necessary to quickly adapt the model to the variations of light conditions, which can rapidly and significantly modify the reference image, especially in cases of natural illumination. In addition, it is necessary to avoid including players who remain in the same position for a certain period of time (goalkeepers are a particular problem for goal detection as they can remain relatively still when play is elsewhere on the pitch) in the background model. To obtain these two opposite requirements, we chose to use two different values for  $\alpha$  in the updating equations (1) (2). The binary mask  $M(x, y)$  allows us to switch between these two values, and permits us to quickly update static points ( $M(x, y) = 0$ ) and to slowly update moving ones ( $M(x, y) = 1$ ). Let  $\alpha_S$  and  $\alpha_D$  be the two updating values for static and dynamic points respectively:

$$\alpha(x, y) = \begin{cases} \alpha_S & \text{if } M(x, y) = 1 \\ \alpha_D & \text{otherwise} \end{cases} \quad (4)$$

In our experiments we used  $\alpha_S = 0.02$  and  $\alpha_D = 0.5$ . The choice of a small value for  $\alpha_S$  is owed to the consideration that very sudden changes in light conditions can produce artifacts in binary mask 3: in such cases these artifacts would be slowly absorbed into the background, while they would have remained permanent if we had used  $\alpha_S = 0$ . The binary image of moving points is the input of the circle detection algorithm based on the analysis of the curvature of isophotes.

## 2.2 Pixel Subset Detection by Isophotes Curvature

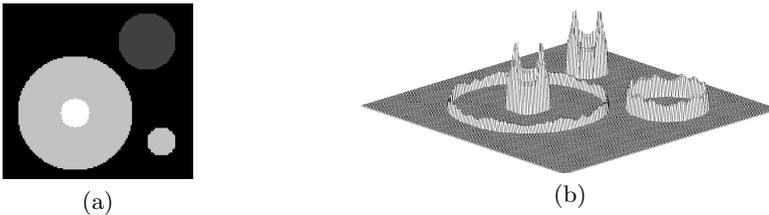
Isophotes properties make them particularly suitable for objects detection and image segmentation, e.g. they are successfully used for accurate eye center location [8]; in particular, it has been demonstrated that their shapes are independent to rotation and varying lighting conditions [5]. Analysis of the curvature can improve the ball detection performance for different reasons. First of all, edge extraction can be performed with a low threshold, in order to detect a large number of edge pixels. This ensures to preserve the ball boundary in the edge map, which otherwise can be lost with severe thresholds in the edge extraction, for example in case of varying environmental conditions (natural or artificial lights, weather changes, etc); this feature is fundamental considering that circle detection implements the *ball candidate extraction* phase. A large edge pixels number does not affect the performances because successively only pixels with a curvature complaint with the ball target radius (with a certain tolerance of few pixels, which takes into account variations in the ball radius given by the non-constant distance between it and the camera) are selected and elaborated. In addition, the sampling process results so limited to a restricted subset given by pixels with an higher probability to lie on the ball boundary; consequently the number of iterations necessary is reduced.

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

$$\kappa = \frac{1}{r} = -\frac{L_y^2 L_{xx} - 2L_x L_{xy} L_y + L_x^2 L_{yy}}{(L_x^2 + L_y^2)^{3/2}} \quad (5)$$

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 [8]). Isophotes curvature is restricted to edge pixels  $V$ , as shown in the simplistic image in fig. 1 (in order to reduce the aliasing effect due to the image discretization a median filter is applied). Obtained  $\kappa$  distribution is analyzed to detect the occurrence of most probable values, as it was a probabilistic distribution.

The basic idea of this procedure is that, if a circle is in the image, there is an accumulation of edge pixels with the corresponding curvature. We use Mean Shift to detect local maxima in  $\kappa$  distribution, assigning at each edge pixel a



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

probability weighted by a 1D gaussian kernel;  $V$  is then divided into subsets  $V_i$ , given by pixels with the same isophote curvature  $\kappa_i$ . Finally subset(s)  $V_b$  with curvature compliant with the target radius of the balloon are selected.

### 2.3 Iterative Computation of the Candidate Circles

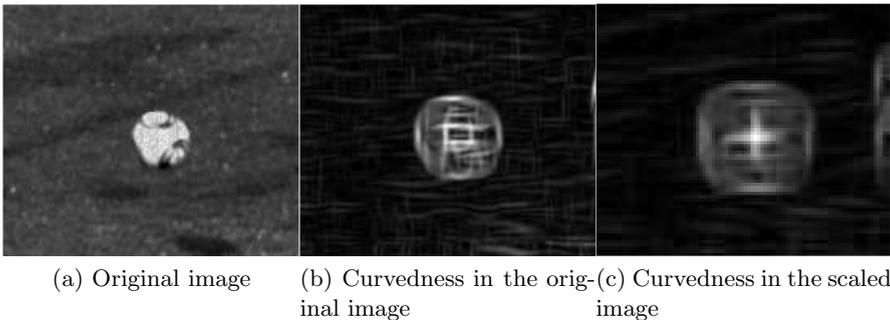
Edge pixels in  $V_b$  are processed by an iterative randomized algorithm: at each iteration three pixels  $(v_j, v_k, v_l)$  are randomly sampled and used to compute the parameters of a circle  $C_{jkl}$  (center coordinates and radius). A series of evidence checks are performed to determine if  $C_{jkl}$  can be considered a valid circle or not. Firstly it is verified that  $C_{jkl}$  center is located in the image and the radius is compared to the expected  $r_i = 1/\kappa_i$ ; if the difference is too high (i.e.  $> 1$  pixel),  $C_{jkl}$  is discarded, because this means that the sampled edge points belong to different circles of equal radius.

A further check is performed applying the curvedness operator to the image [9]. Curvedness is a rotational invariant gradient operator, able to provide a measurement of the level of steepness of the gradient image:

$$\text{curvedness} = \sqrt{L_{xx}^2 + 2L_{xy}^2 + L_{yy}^2}. \quad (6)$$

Curvedness is strictly related to the density of the isophotes: in particular it is higher where isphote are denser, e.g. around the boundaries-edge of the objects. Referred to circle detection this means that  $C_{jkl}$ , to be a valid closed circle, must necessarily contain at least a local maximum of the curvedness; intuitively, if the circle is relatively small (radius of few pixels), the maximum will be located on circle center (fig. 2). Local maxima are so localized in the curvedness distribution by a non-local maxima suppression filter, with a square window of size equal to the expected ball diameter. When  $C_{jkl}$  does not contain any curvedness local maximum is discarded.

In the last check, it is verified that at least an other edge pixel in  $V_b$  is close to  $C_{jkl}$ , following the four points approach proposed in [10], where it is shown



**Fig. 2.** Curvedness values of a real ball

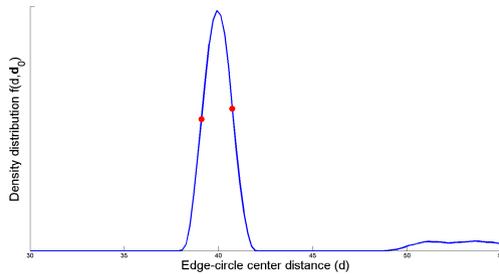
as this allows to reduce the number of necessary iterations until 95%. However, while in [10] the fourth point is randomly sampled from the edge pixel set, in our approach we process all points in  $V_b$ , exploiting the reduced edge pixel set.

## 2.4 Voting Algorithm

For each subset  $V_b$ , found candidate circles are analyzed by a voting algorithm, firstly to select the best fitting one, secondly to verify if it is actually 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  $K(\cdot)$ :

$$f(d, \mathbf{d}_0) = \frac{1}{2\pi r_i N} \sum_{p=1}^N \frac{1}{h_{\mathbf{d}_0}} K\left(\frac{d - d_{0p}}{h_{\mathbf{d}_0}}\right) \quad (7)$$

with  $d$  the generic pixel-center distance,  $N$  the number of edge pixels and  $h_{\mathbf{d}_0}$  bandwidth of Parzen windows. This procedure does not require parameters, because  $h_{\mathbf{d}_0}$  is computed directly from  $\mathbf{d}_0$  according to the MAD formulation [11]. Candidate with the highest absolute maximum of  $f$  is chosen, and pixels around it are taken as lying along its circumference (namely *inliers*, fig. 3).  $f$  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 lower radius. Finally, to avoid false positives, detected circle is considered valid only if the ratio of *inliers* over  $L$  is higher than a threshold  $T_{cov}$ .



**Fig. 3.** Kernel density function  $f(d, \mathbf{d}_0)$  of the distance between edge pixels and the candidate circle center. When estimate is correct,  $f$  has a peak around the radius value; points around correspond to edge pixels lying along its circumference (*inliers*). They are selected determining the two points (red dots) around the peak where behaviour of first derivative of  $f$  changes.

## 2.5 Linearization

Detected circle(s) 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, detected circle does not perfectly match the ideal one. To reduce this effect we use a linear error compensation, adjusting 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:

$$(x - \bar{x}_c)^2 + (y - \bar{y}_c)^2 - \bar{R}_c^2 = 0. \quad (8)$$

Given the detected parameters  $(x_c, y_c, R_c)$ , affected by errors  $(\delta x_c, \delta y_c, \delta R_c)$ , (8) becomes:

$$(x - (x_c - \delta x_c))^2 + (y - (y_c - \delta y_c))^2 - (R_c - \delta R_c)^2 = 0. \quad (9)$$

Linearizing (9), and considering the generic edge pixel coordinates  $(x_i, y_i)$ , we obtain:

$$B\delta x_c + C\delta y_c + D\delta R_c = A \quad (10)$$

where

$$\begin{aligned} A &= -(x_i - x_c)^2 - (y_i - y_c)^2 + R_c^2 \\ B &= 2(x_i - x_c) \\ C &= 2(y_i - y_c) \\ D &= 2R_c. \end{aligned} \quad (11)$$

Applying (10) to all  $N_c$  *inliers*, we obtain the following over-determined linear system, which is able to provide an evaluation of the parameters errors (to solve this problem we use a Moore-Penrose pseudo-inverse):

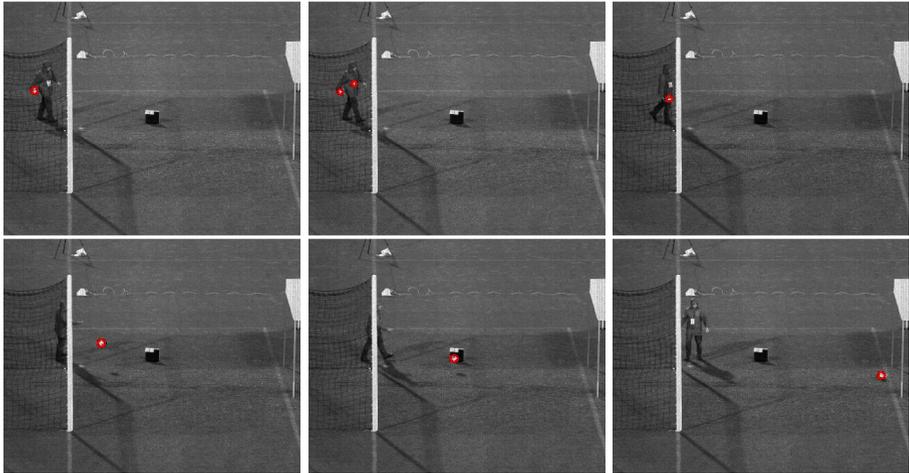
$$\begin{pmatrix} B_0 & C_0 & D_0 \\ B_1 & C_1 & D_1 \\ \vdots & \vdots & \vdots \\ B_{N_c} & C_{N_c} & D_{N_c} \end{pmatrix} \begin{pmatrix} \delta x_c \\ \delta y_c \\ \delta R_c \end{pmatrix} = \begin{pmatrix} A_0 \\ A_1 \\ \vdots \\ A_{N_c} \end{pmatrix} \quad (12)$$

## 3 Experimental Results

The tests are executed on image sequences acquired from fixed cameras placed on the stands of the stadium.

The first experiment is performed on a video sequence acquired during a training session by a camera with a spatial resolution of  $1024 \times 768$  pixels at 504 fps. In Fig. 4 six frames are shown, with superimposed the detection results of the algorithm. In this experiment ball is always correctly detected, and number of false positives is limited and referred to moving structures in the images. Computation has been performed with a threshold  $T_{cov}$  of 90%, and a maximum number of iteration of 1000 for each valid subset  $V_b$ .

To evaluate the detection performances of the algorithm, we have extracted from the video under consideration 500 frames where the ball is always present.



**Fig. 4.** Ball detection on training video

On these extracted frames we compare the performances of the proposed approach with a reference solver; in particular we use the Matlab implementation of the circular Hough transform, coupled with the same algorithm of background suppression and a level of sensitivity of 0.97; for both solvers we have used a tolerance of two pixels for the target ball radius. Obtained results are shown in table 1. The proposed randomized solver is able to achieve a detection rate higher than circular Hough transform method (more than 90 % versus 64 %), providing in addition a smaller number of false positives (less than a false positive for frame).

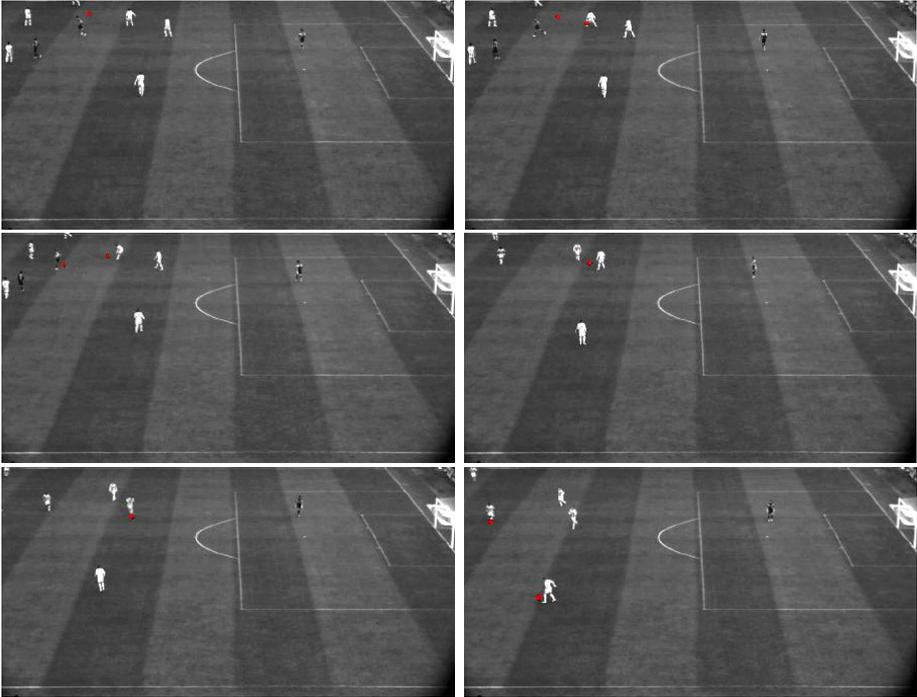
**Table 1.** Detection performances obtained with 500 frames extracted from the training video

	Randomized solver	Circular Hough
# of detected ball	460	323
Ball detection rate	92%	64%
# of false positives/frame	0.6	3.2

An other series of tests are executed on a set of frames extracted from the *ISSIA-CNR Soccer Dataset* [12]. The dataset contains six synchronized views acquired by six Full-HD *DALSA 25-2M30* cameras, three for each major side of the playing-field, at 25 fps in quasi-stationary playing conditions. Videos are acquired during some matches of the italian "serie A". In Fig. 5 six frames are shown, with superimposed the detection results of the algorithm. In this case we have observed a reduction of the detection rate (however higher than reference solver's one), motivated by the complexity of the scene (tab. 2); in particular

**Table 2.** Results on videos acquired during a real soccer match

	Randomized solver	Circular Hough
Ball detection rate	86%	79%
# of false positives/frame	0.1	1.2

**Fig. 5.** Ball detection on real video

cases of missed detection are relative to frames where the ball is stationary, probably because the background suppression algorithm tends to mainly highlight the ball when it is moving. Number of false positive for frame continues to be less than 1.

### 3.1 Conclusions

In this work a randomized iterative work-flow is presented in order to provide a circle detector able to implement the *ball candidate extraction* stage in a complete ball detector for soccer game. This exploits geometrical properties of the isophote distribution on video frames to cluster a large number of edge pixels into subsets of equal curvature, limiting the successive intensive computation only to the most meaningful ones. The method, coupled with a background suppression algorithm, is applied on real videos and standard database achieving a detection rate higher than standard circular Hough solution, maintaining a lower number of false

positives. This can simplify the successive *ball candidate validation* necessary to discard false positives, and eventually, estimate with tracking techniques the positions of the ball in frames where circle detector fails (e.g., in presence of occlusions).

## References

1. Davies, E.R.: Machine vision: theory, algorithms, practicalities. Morgan Kaufmann (2004)
2. D’Orazio, T., Leo, M.: A review of vision-based systems for soccer video analysis. *Pattern Recognition* 43(8), 2911–2926 (2010)
3. Mazzeo, P.L., Leo, M., Spagnolo, P., Nitti, M.: Soccer ball detection by comparing different feature extraction methodologies. *Advances in Artificial Intelligence* 6 (2012)
4. Chen, T.C., Chung, K.L.: An efficient randomized algorithm for detecting circles. *Computer Vision and Image Understanding* 83(2), 172–191 (2001)
5. Lichtenauer, J., Hendriks, E., Reinders, M.: Isophote properties as features for object detection. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR*, vol. 2, pp. 649–654 (2005)
6. Canny, J.: A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence PAMI* 8(6), 679–698 (1986)
7. Spagnolo, P., D’Orazio, T., Leo, M., Distante, A.: Moving Object Segmentation by Background Subtraction and Temporal Analysis. *Image and Vision Computing* 24, 411–423 (2006)
8. Valenti, R., Gevers, T.: Accurate eye center location through invariant isocentric patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(9), 1785–1798 (2012)
9. Koenderink, J.J., van Doorn, A.J.: Surface shape and curvature scales. *Image and Vision Computing* 10(8), 557–564 (1992)
10. Chung, K.L., Huang, Y.H., Shen, S.M., Krylov, A.S., Yurin, D.V., Semeikina, E.V.: Efficient sampling strategy and refinement strategy for randomized circle detection. *Pattern Recognition* 45(1), 252–263 (2012)
11. Rousseeuw, P.J., Croux, C.: Alternatives to the median absolute deviation. *Journal of the American Statistical Association* 88(424), 1273–1283 (1993)
12. D’Orazio, T., Leo, M., Mosca, N., Spagnolo, P., Mazzeo, P.L.: A Semi-Automatic System for Ground Truth Generation of Soccer Video Sequences. In: *6th IEEE International Conference on Advanced Video and Signal Surveillance*, Genoa, Italy (September 2009)