Hindawi Publishing Corporation
Advances in Artificial Intelligence
Volume 2012, Article ID 512159, 12 pages
doi:10.1155/2012/512159

Research Article

Soccer Ball Detection by Comparing Different Feature Extraction Methodologies

Pier Luigi Mazzeo, Marco Leo, Paolo Spagnolo, and Massimiliano Nitti

Istituto di Studi sui Sistemi Intelligenti per l'Automazione, CNR, Via G. Amendola 122/D, 70126 Bari, Italy

Correspondence should be addressed to Pier Luigi Mazzeo, mazzeo@ba.issia.cnr.it

Received 30 May 2012; Accepted 26 August 2012

Academic Editor: Djamel Bouchaffra

This paper presents a comparison of different feature extraction methods for automatically recognizing soccer ball patterns through a probabilistic analysis. It contributes to investigate different well-known feature extraction approaches applied in a soccer environment, in order to measure robustness accuracy and detection performances. This work, evaluating different methodologies, permits to select the one which achieves best performances in terms of detection rate and CPU processing time. The effectiveness of the different methodologies is demonstrated by a huge number of experiments on real ball examples under challenging conditions.

1. Introduction

Automatic sport video analysis has become one of the most attractive research fields in the areas of computer vision and multimedia technologies [1]. This has led to opportunities to develop applications dealing with the analysis of different sports such as tennis, golf, American football, baseball, basketball, and hockey. However, due to its worldwide viewership and tremendous commercial value, there has been an explosive growth in the research area of soccer video analysis [2, 3] and a wide spectrum of possible applications have been considered [4–6]. Some applications (automatic highlight identification, video annotation and browsing, content-based video compression, automatic summarization of play, customized advertisement insertion) require only the extraction of low-level visual features (dominant color, camera motion, image texture, playing field line orientation, change of camera view, text recognition) and for this reason they have reached some maturity [7, 8]. In other applications (such as verification of referee decision, tactics analysis, and player and team statistic evaluations), instead, more complex evaluations and high-level domain analysis are required. Unfortunately, the currently available methods have not yet reached a satisfactory accuracy level and then new solutions have to be investigated [9].

In particular, the detection and localization of the ball in each frame is an issue that still requires more investigation. The ball is invariably the focus of attention during the game, but, unfortunately, its automatic detection and localization in images is challenging as a great number of problems have to be managed: occlusions, shadowing, presence of very similar objects near the field lines and regions of player's bodies, appearance modifications (e.g., when the ball is inside the goal, it is faded by the net and it also experiences a significant amount of deformation during collisions), and unpredictable motion (e.g., when the ball is shot by players).

In the last decade, different approaches have been proposed to face the ball detection problem in soccer images. The most successful approaches in literature consist of two separate processing phases: the first one aims at selecting, in each image, the regions which most probably contain the ball (*ball candidates extraction*). These candidates regions are then deeply analyzed in order to recognize which of them really contains the ball (*ball candidate validation*).

Candidate ball extraction can be performed using global information as size, color, and shape or a combination of them. In particular, the circular Hough transform (CHT) and several modified versions have long been recognized as robust techniques for curve detection and have been largely applied by the scientific community for candidate ball detection purposes [10].

The choice of the best methodology for the validation of ball candidates is more controversial. In [11], a candidate verification procedure based on Kalman filter is presented. In [12], size, color, velocity, and longevity features are used to discriminate the ball from other objects. These approaches experience difficulties in ball candidate validation when many moving entities are simultaneously in the scene (the ball is not isolated) or when the ball abruptly changes its trajectory (e.g., in the case of rebounds or shots). To overcome this drawback, other approaches focus on ball pattern extraction and recognition: for example, in [13], the wavelet coefficients are proposed to discriminate between ball candidates and nonball candidates. More recently, the possibility to use scale-invariant feature transform (SIFT) to encode local information of the ball and to match it between ball instances has been also explored [14].

This paper presents a comparison of different feature extraction approaches in order to recognize soccer ball patterns. This kind of detecting problem is related to the flat object recognition problem from 2D intensity images that has been largely studied by the scientific community. The comparison is carried out by using a framework in which candidate ball regions are extracted by a directional circular Hough transform (CHT), then the image patterns are preprocessed by one of the comparing approaches and finally a probabilistic classifier is used to label each pattern as ball or no-ball. In particular, wavelet transform (WT), principal component analysis (PCA), scale-invariant feature transform (SIFT), and Histogram have been applied to the patterns in order to get a more suitable representation, possibly making use of a reduced number of coefficients. These techniques have been compared in order to choose the best one for our application domain. The technique producing the highest detection rate combined with the lowest CPU processing time has been then used in the final ball detection system.

The considered approaches were tested on a huge number of real ball images acquired in presence of translation, scaling, rotation, illumination changes, local geometric distortion, clutter, and partial and heavy occlusion.

In the rest of the paper, Section 2 gives system overview whereas Sections 3 and 4 detail its fundamental steps. Section 5 presents the setup used for the experiments, while in Section 6, the experimental results, a comparison with implemented approaches, and an extensive discussion are presented. Finally, in Section 7, conclusions are drawn.

2. Ball Detection System Overview

As introduced in Section 1 we have implemented a vision system which automatically detects the ball in an acquired soccer video sequence. The proposed system is composed of three main blocks: in the first one, a background subtraction technique is combined with a circle detection approach to extract ball candidate regions. Then a feature extraction scheme is used to represent image patterns, and finally data classification is performed by using a supervised learning scheme. Figure 1 schematizes the proposed approach, whereas, in the following sections, a detailed description of the involved algorithmic procedures is given.

3. Candidate Ball Regions Detection

Ball candidates are identified in two phases: 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, the algorithm evaluates

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

$$\overline{\sigma^t(x,y)} = \alpha\left|\mu^t(x,y) - \overline{\mu^t(x,y)}\right| + (1-\alpha)\overline{\sigma^{t-1}(x,y)}. \tag{2}$$

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 which 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

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

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 in the background model 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). To obtain these two opposite requirements, we have chosen to use two different values for α in the updating equations (1) and (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 α_S and α_D be the two updating values for static and dynamic points, respectively,

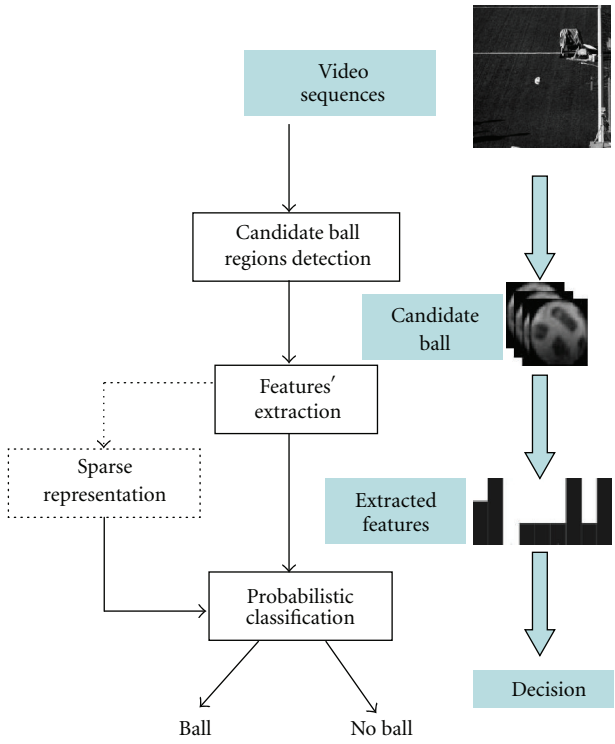


Figure 1: Graphical overview of the proposed approach.

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

In our experiments, we used $\alpha_S = 0.02$ and $\alpha_D = 0.5$. The choice of a small value for α_S is owed to the consideration that very sudden changes in light conditions can produce artifacts in the binary mask: in such cases, these artifacts will be slowly absorbed into the background, while they would remain permanent if we had used $\alpha_S = 0$.

The binary image of moving points, the output of the background subtraction phase, is the input of the following circle detection algorithm.

3.1. Circle Detection. The circle hough transform (CHT) aims to find circular patterns of a given radius R within an image. Each edge point contributes a circle of radius R to an output accumulator space. The peak in the output accumulator space is detected where these contributed circles overlap at the center of the original circle. In order to reduce the computational burden and the number of false positives typical of the CHT, a number of modifications have been widely implemented in the last decade. The use of edge orientation information limits the possible positions of the center for each edge point. In this way, only an arc perpendicular to the edge orientation at a distance R from the edge point needs to be plotted. The CHT and also its modifications can be formulated as convolutions applied to an edge magnitude image (after a suitable edge detection) [13]. We have defined a circle detection operator that is applied over all the image pixels and produces a maximal value when a circle is detected with a radius in the range $[R_{\min}, R_{\max}]$:

$$u(x,y) = \frac{\iint_{D(x,y)} \vec{e}(\alpha,\beta) \cdot \vec{O}(\alpha - x, \beta - y)\, d\alpha d\beta}{2\pi(R_{\max} - R_{\min})}, \tag{5}$$

where the domain $D(x,y)$ is defined as

$$D(x,y) = \left\{ (\alpha,\beta) \in \Re^2 \mid R_{\min}^2 \le (\alpha - x)^2 + (\beta - y)^2 \le R_{\max}^2 \right\}, \tag{6}$$

$\vec{e}$ is the normalized gradient vector:

$$\vec{e}(x,y) = \left[\frac{E_x(x,y)}{|E|}, \frac{E_y(x,y)}{|E|} \right]^T, \tag{7}$$

and $\vec{O}$ is the kernel vector

$$\vec{O}(x,y) = \left[\frac{\cos(\tan^{-1}(y/x))}{\sqrt{x^2 + y^2}}, \frac{\sin(\tan^{-1}(y/x))}{\sqrt{x^2 + y^2}} \right]^T. \tag{8}$$

The use of the normalized gradient vector in (9) is necessary in order to have an operator whose results are independent from the intensity of the gradient in each point: we want to be sure that the circle detected in the image is the most complete in terms of contours and not the most contrasted in the image. Indeed it could be possible that a circle that is not well contrasted in the image gives a convolution result lower than another object that is not exactly circular but has a greater gradient. The kernel vector

contains a normalization factor (the division by the distance of each point from the center of the kernel) which is fundamental for ensuring we have the same values in the accumulation space when circles with different radii in the admissible range are found. Moreover, the normalization ensures that the peak in the convolution result is obtained for the most complete circle and not for the greatest in the annulus. As a last consideration, in (5), the division by $(2\Pi \cdot (R_{\max} - R_{\min}))$ guarantees the final result of our operator in the range $[-1, 1]$ regardless of the radius value considered in the procedure. The masks implementing the kernel vector have a dimension of $(2 \cdot R_{\max} + 1) \times (2 \cdot R_{\max} + 1)$, and they represent in each point the direction of the radial vector scaled by the distance from the center. The convolution between the gradient versor images and these masks evaluates how many points in the image have the gradient direction concordant with the gradient direction of a range of circles. Then, the peak in the accumulator array gives the candidate center of the circle in the image.

4. Feature Extraction Methodologies

In this step, the selected subimages are processed by different feature extraction methodologies in order to represent them only by coefficients containing the most discriminant information. A secondary aim is also to characterize the images with a small number of features in order to gain in computational time. Object recognition by using a learning-from-examples technique is in fact related to computational issues. In order to achieve real-time performances, the computational time to classify patterns should be small. The main parameter connected to high computational complexity is certainly the input space dimension. A reduction of the input size is the first step to successfully speed up the classification process. This requirement can be satisfied by using a feature extraction algorithm able to store all the important information about input patterns in a small set of coefficients.

Wavelet transform (WT), principal component analysis (PCA), scale-invariant feature transform (SIFT), and histogram representation (HR) are different approaches allowing to reduce the dimension of the input space, because they capture the significant variations of input patterns in a smaller number of coefficients. In the following four subsections, we briefly review WT, PCA, SIFT, and HR approaches.

4.1. Wavelet Transform. The WT is an extension of the Fourier transform that contains both frequency and spatial information [15]. The WT operator $F : L^2(\Re) \to L^2(\Re)$ can be defined as follows:

$$F(f(s)) = \hat{f}(s) = \int_{-\infty}^{+\infty} f(u)\Psi_{s,t}(u)du \tag{9}$$

with

$$\Psi_{s,t}(u) = \frac{1}{|s|^p}\Psi\left(\frac{u-t}{s}\right), \tag{10}$$

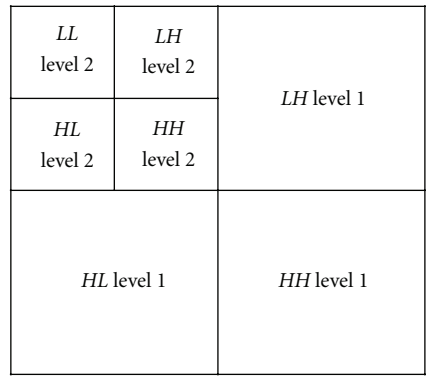


Figure 2: The decomposition of the image with a 2-level wavelet transform.

when s changes, the frequencies which the function Ψ operates are changed, and when t changes, the function Ψ is moved on all the support of the function f. In this paper, we have used a discrete wavelet transform supplying a hierarchical representation of the image implemented with the iterative application of two filters: a low-pass filter (approximation filter) and its complementary one in frequency (detail filter). A bidimensional WT breaks an image down into four subsampled or decimated images. In Figure 2, the final result of a 2-level WT is shown. In each subimage, the capital letters refer to the filters applied on the image of the previous level: H stands for a high-pass filter, and L stands for a low-pass filter. The first letter is the filter that has been applied in the horizontal direction, while the second letter is the filter that has been applied in the vertical direction. The bands LL is a coarser approximation of the original image. The band LH and HL record the changes of the image along horizontal and vertical directions. The band HH shows the high-frequency components of the image. Decomposition is iterated on the LL subband that contains the low-frequency information of the previous stage. For example, after applying a 2-level WT, an image is subdivided into subbands of different frequency components (Figure 3). Numerous filters can be used to implement WT: we have chosen Haar and Daubechies filters for their simplicity and orthogonality.

4.2. Principal Component Analysis (PCA). Principal component analysis (PCA) provides an efficient method to reduce the number of features to work with [16]. It transforms the original set of (possibly) correlated features into a small set of uncorrelated ones. In particular, PCA determines an orthogonal basis for a set of images involving an eigenanalysis of their covariance matrix. Such a basis is given by the eigenvectors (principal components) of that matrix. They are obtained by solving the following eigenvalue problem:

$$Su = \lambda u, \tag{11}$$

where S is the covariance matrix of the original set of images, u is the vector of eigenvectors, and λ is the vector of eigenvalues. We have used the SVD technique to

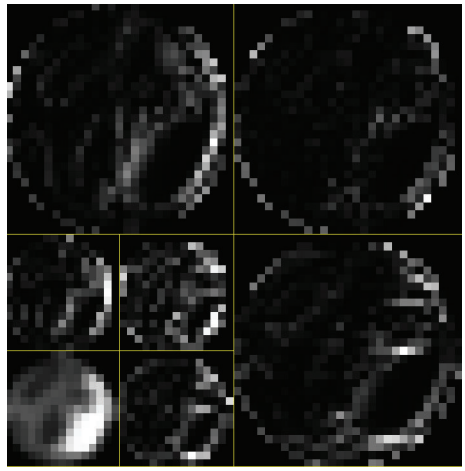


Figure 3: The 2-level wavelet transform on a subimage containing the soccer ball.

solve eigenstructure decomposition problem. The covariance matrix has been evaluated on the entire set of training images. The new set of uncorrelated features is obtained projecting the images (old features) into that basis both in the training and in the testing phases. Many algorithms for PCA return u and λ ranked from highest to lowest. The first eigenvectors capture the largest variability of images, and each succeeding one accounts for the remaining variability. Therefore, the new set of uncorrelated features is obtained as a linear combination of the old ones considering that the first eigenvectors contain the highest information level. The significant variations in images are captured by few vectors (less dimensionality) with respect to the input space.

4.3. Scale-Invariant Feature Transform (SIFT). The scale-invariant feature transform is a method for extracting distinctive invariant features from the images that can be used to perform reliable matching between different views of an object or a scene [17]. The features are invariant to image scale and rotation, and they provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from many different images. The algorithm consists of four main steps:

(1) scale-space extrema detection;

(2) keypoints localization;

(3) orientation assignment;

(4) keypoint description.

The first stage identifies locations and scales that can be assigned under differing views of the same object. Detecting locations that are invariant to scale change of the image can be accomplished by searching for stable features across all possible scales, using a continuous function of scale known as a scale space. Under a variety of reasonable assumptions, the only possible scale-space kernel is the Gaussian function. Therefore, the scale space of an image is defined as a function $L(x, y, \sigma)$, that is produced from the convolution of a variable-scale Gaussian, $G(x, y, \sigma)$, with an input image, $I(x, y)$, that is,

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y), \tag{12}$$

where $*$ is the convolution operator and $G(x, y, \sigma)$ is defined as

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}. \tag{13}$$

The keypoints are detected using scale-space extrema in the difference of Gaussian (DoG) function D convolved with the image $I(x, y)$:

$$\begin{aligned} D(x, y, \sigma) &= \left[G(x, y, k\sigma) - G(x, y, \sigma)\right] * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma), \end{aligned} \tag{14}$$

where k is the multiplicative constant factor which separates two nearby scales. In order to detect the local maxima and minima of $D(x, y, \sigma)$, each sample point is compared to its eight neighbors in the current image and to its nine neighbors in the scale above and below. It is selected only if it is larger than all of these neighbors or smaller than all of them. Once a keypoint candidate has been found by comparing a pixel to its neighbors, the next step is to perform a detailed fit to the nearby data for location, scale, and ratio of principal curvatures. This information allows points to be rejected if they have low contrast (and are therefore sensitive to noise) or are poorly localized along an edge. A 3D quadratic function is fitted to the local sample points. The approach starts with the Taylor expansion (up to the quadratic terms) with sample point as the origin

$$D(X) = D + \frac{\partial D^T}{\partial X} X + \frac{1}{2} X^T \frac{\partial^2 D}{\partial X^2} X, \tag{15}$$

where D and its derivatives are evaluated a the sample point $X = (x, y, \sigma)^T$. The location of the extremum is obtained taking the derivative with respect to X and setting it to 0, giving

$$\hat{X} = -\frac{\partial^2 D^{-1}}{\partial X^2} \frac{\partial D}{\partial X}, \tag{16}$$

that is, a 3×3 linear system easily solvable. The function value a the extremum

$$D\left(\hat{X}\right) = D + \frac{1}{2} \frac{\partial D^T}{\partial X} \hat{X} \tag{17}$$

is useful for rejecting unstable extrema with low contrast. A this point, the algorithm rejects also keypoints with poorly defined peaks, that is, those points having, in the difference of Gaussian function, a large principal curvature across the edge but a small one in the perpendicular direction. By assigning a consistent orientation, based on local image properties, the keypoint descriptor can be represented relative to

this orientation and therefore achieve invariance to image rotation.

After the localization of the interest points in the candidate ball images, the region around each of them has to be accurately described in order to encode local information in a representation that can assure robustness for matching purposes. A consistent orientation is firstly assigned to each detected point: in this way, further information representation can be done relative to this orientation achieving invariance to image rotation that is fundamental to getting an effective representation especially when ball images are handled.

Instead of directly using pixel differences, the orientation is associated to each point after computing a corresponding edge map in the surrounding circular area with radii R. The edge map E is computed as suggested in [18]. Then, for each edge point $E(x, y)$, corresponding magnitude and theta values are computed:

$$\begin{aligned} G_x(x,y) &= \sum_{i=-1}^{1} I(x+i, y+1) \\ &\quad - \sum_{i=-1}^{1} I(x+i, y-1) + I(x, y+1) - I(x, y-1), \\ G_y(x,y) &= \sum_{i=-1}^{1} I(x+1, y+i) \\ &\quad - \sum_{i=-1}^{1} I(x-1, y-i) + I(x+1, y) - I(x-1, y), \\ \theta(x,y) &= \tan^{-1}\left(\frac{G_y(x,y)}{G_x(x,y)}\right). \end{aligned} \tag{18}$$

A 36-bin orientation histogram is then formed from the gradient orientation of sample points within a region of radii R around the interest point.

Peaks in the orientation histogram correspond to dominant directions of the local gradient. The highest peak in the histogram and any other local peak that is within 80% of the highest peak are then detected.

The previous operations have assigned an image location, scale, and orientation to each interest point. These parameters impose a repeatable local 2D coordinate system to describe the local image region and therefore provide invariance to these parameters. The next step is to compute a descriptor for the local image region that is highly distinctive and is as invariant as possible to remaining variations.

To do that, the first step is to smooth the image I to the appropriate level considering the associated radii R of the interest point.

Then, a descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around the interest point location. These samples are then accumulated into orientation histograms summarizing the contents over 4×4 subregions. This results in a feature vector containing 128 elements.

Unfortunately the number of interest points can differ from a candidate ball image to another: this makes it very complex to compare two different images until an intermediate processing level is introduced.

Figure 4 shows some keypoints localized on two different ball images, and Figure 5 illustrates how two keypoints match among two ball images.

In this paper, the fixed-length feature vectors are obtained in this way: the descriptors of the regions around the detected interest points in a set of manually labeled ball images are, at first, quantized into visual words with the k-means algorithm. A candidate ball image is then represented by the frequency histogram of visual words obtained by assigning each descriptor of the image to the closest visual word. In Figure 6 is an example of how this processing pipeline works: starting from the ball image (Figure 6(a)), each region around any of the detected interest point is described by a descriptor having 128 elements, (Figure 6(b)). Then, the descriptor is associated to a cluster between the N clusters (code-book elements) built by vector quantization based on k-means algorithm. Finally, all the clusters associate to the descriptors in the same image are counted and a fixed-length feature vector V is associated to the resulting histogram representation, Figure 6(c).

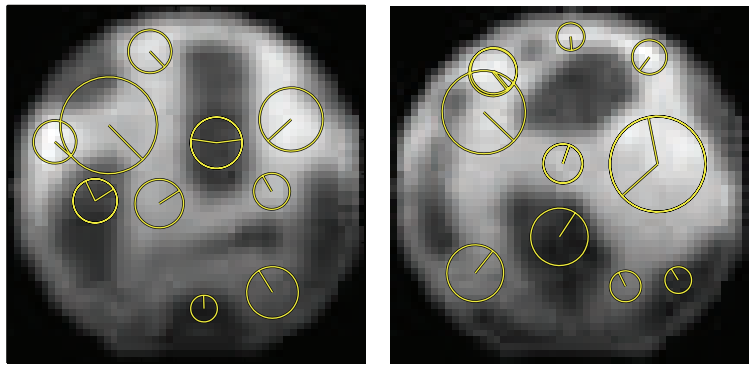
4.4. Image Histogram. The distributions of colors or gray levels of images are commonly used in the image analysis and classification. The gray level distribution can be presented as a gray level histogram:

$$H(G) = \frac{n_G}{n} \quad G = 0, 1, \ldots, NG - 1, \tag{19}$$

where n_G is the number of pixels having gray level G, n is the total number of pixels, and NG is the total number of gray levels.

4.5. Probabilistic Classification. The following step in the proposed framework aims at introducing an automatic method to distinguish between ball and no-ball instances on the basis of the feature vector extracted by one of the previously mentioned preprocessing strategies. To accomplish this task, a probabilistic approach has been used. Probabilistic methods for pattern classification are very common in literature as reported by [19]. The so-called naive Bayesian classification is the optimal method of supervised learning if the values of the attributes of an example are independent given the class of the example. Although this assumption is almost always violated in practice, recent works have shown that naive Bayesian learning is remarkably effective in practice and difficult to improve upon systematically. On many real-world example datasets naive Bayesian learning gives better test set accuracy than any other known method [20]. In general, a Naïve Bayes classifier is also preferable for its computational efficiency.

Probabilistic approaches to classification typically involve modelling the conditional probability distribution $P(C \mid D)$, where C ranges over classes and D over descriptions, in some language, of objects to be classified. Given a description d



(a) (b)

Figure 4: The keypoints localized on two different ball images.

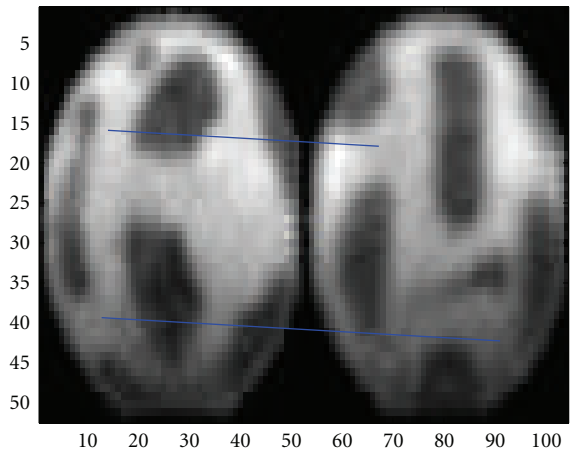


Figure 5: Two keypoints matching among two ball images.

of a particular object, the class $\text{argmax}_c P(C = c \mid D = d)$ is assigned. A Bayesian approach splits this posterior distribution into a prior distribution $P(C)$ and a likelihood $P(D \mid C)$:

$$\begin{aligned} &\text{argmax}_c P(C = c \mid D = d) \\ &\quad = \text{argmax}_c \frac{(P(D = d \mid C = c)P(C = c))}{P(D = d)}. \end{aligned} \tag{20}$$

The key term in (21) is $P(D = d \mid C = c)$, the likelihood of the given description given the class (often abbreviated to $P(d \mid c)$). A Bayesian classifier estimates these likelihoods from training data. If the assumption that all attributes are independent given the class:

$$P(A_1 = a_1, \ldots, A_n = a_n \mid C = c) = \prod_{i=1}^{n} P(A_i = a_i \mid C = c), \tag{21}$$

then a naive Bayesian classifier (often abbreviated to *naive Bayes*) is introduced. This means that a naive Bayes classifier ignores interactions between attributes within individuals of the same class.

Further details and discussions about the practical consequences of this assumption can be found in [21].

5. Experimental Setup

Experiments were carried out on image sequences acquired in a real soccer stadium by a Mikroton EOSens MC1362 CCD camera, equipped with a 135 mm focal length. The camera has the area around the goal in its field of view. Using this experimental setup, the whole image size was 1280x1024 pixels whereas the ball corresponded to a circular region with radii in the range ($R_{\text{MIN}} = 24, R_{\text{MAX}} = 28$) depending on the distance of the ball with respect to the optical center of the camera. The camera frame rate was 200 fps with an exposure time of 1 msec in order to avoid blurring effect in the case of high ball speed. A Nike T90 Strike Hi-Vis Soccer Ball (FIFA approved) was used during the experiments.

Images were continuously acquired in a sunny day from 1 PM (when the sun shined and the ball was bright) to 7 PM (just before sunset when the acquired images have become dark and a part of the ball was almost not visible in the scene): in this way, a wide range of different ball appearances was collected providing the possibility to verify the effectiveness of the proposed approach in very different lighting conditions.

During the data acquisition session, a number of shots on goal were performed: the shots differed in ball velocity and direction with respect to the goalmouth. In this way, differently scaled (depending on the distance between the ball and the camera), occluded (by the goal posts), and faded (by the net) ball appearances were experienced. Moreover, some shots hit a rigid wall placed on purpose inside the goal in order to acquire some instances of deformed ball and then to check the sensitiveness of the ball recognition system in the case of ball deformation.

During the acquisition session, about 4,5 million images were collected. Each image was processed to detect candidate ball regions by finding circular arrangement in the edge magnitude map: edge point contributes a circle to an output accumulator space, and the peak in the output accumulator

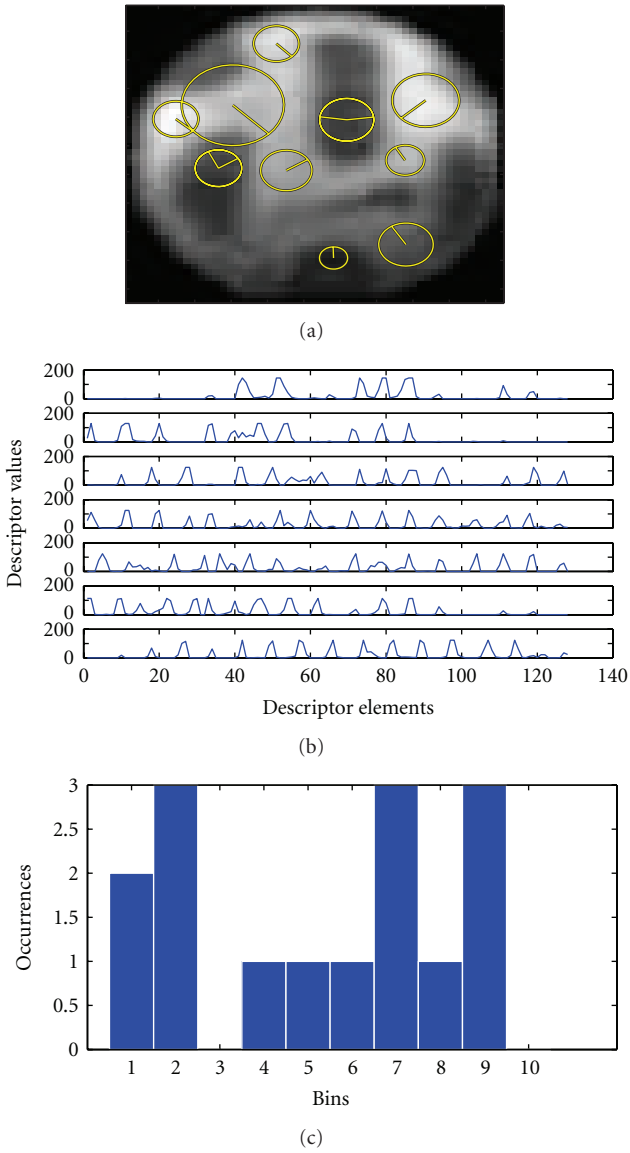


FIGURE 6: (a) Ball image and the detected interest points, (b) relative descriptors (128 elements in each) of the surrounding regions, and (c) the 10-bin histogram representing the ball image using a fixed-length vector.

space is detected where these contributed circles overlap at the center of the original circle.

The main benefits of this method are the low computational cost and the high detection rate when the ball is visible in the image. The main drawback lies in the impossibility of detecting occlusions and ball absence in the image. The circle detection algorithm determines in every situation the highest peaks in the accumulator space. Then, it is difficult to differentiate between images containing and not containing the ball.

In Figure 7(a), one image containing the ball is shown, whereas Figure 7(b) displays the corresponding Hough space (highest values are white, lowest ones are black). Figure 7(c) shows instead two candidate ball regions extracted in correspondence with two peaks in the Hough accumulation space: the region on the left is relative to a ball, whereas the region on the right contains a nonball object.

For each image, a candidate ball region (of size $(2 * R\_MAX+1)x(2*R\_MAX+1)$, i.e., 52×52) was then extracted around the position corresponding to the highest value in the accumulator space.

After that for each candidate ball region, all pixels outside the circular area with radii R_MAX were set to 0 in order to discard information certainly belonging to the background and not to the ball region (see Figure 7(c)).

After this preliminary step, 2574 candidate ball regions were selected and manually divided on the basis of ball presence/absence, lighting conditions, and ball appearance: these sets of images formed the experimental ground truth. The ground truth dataset contained ball acquired in both good lighting conditions (patches acquired from 1 pm to 5 pm) (see Figure 8(a)) and poor lighting conditions (patches acquired during the last 2 hours of acquisition) (see Figure 8(b)). The ground truth data contained a subset (**O**) of examples in which the ball was partially and heavily occluded. In particular, the ball was considered partially occluded if at least half of the ball surface is visible (see Figure 8(d)), otherwise it is heavily occluded (see Figure 8(e)) (contained in the subset (**O**)). In the acquired images, partially or heavily occluded balls occurred either while the ball crosses the goal mouth appearing behind the goal posts or when a person interposes himself/herself between the ball and the camera.

Other special cases in the dataset were the faded ball occurrences, that is, patches relative to balls positioned inside the goal when a net is inserted between it and the camera. Two kinds of faded balls occurred: undeformed ball (see Figure 8(c)) and deformed ball acquired while hitting the rigid wall placed inside the goal (see Figure 8(f)). Figure 8 shows some examples of the ball appearances observed during the experimental phase depending on their different aspects.

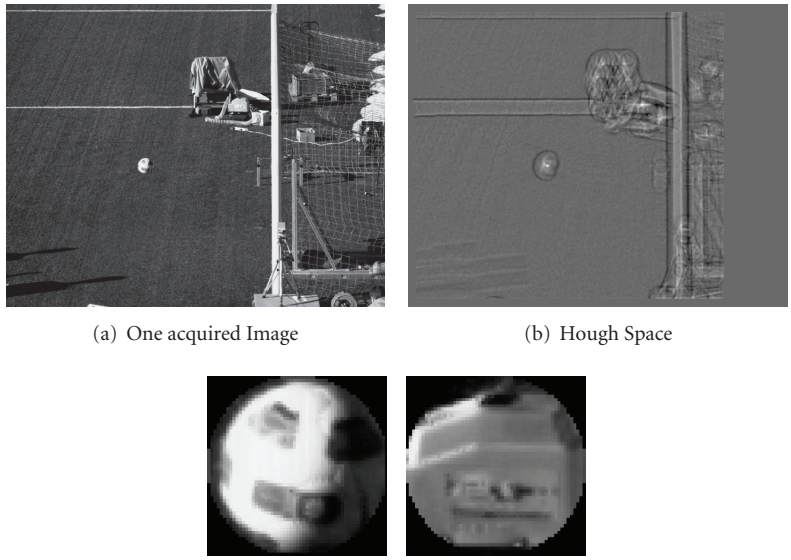
6. Experimental Results

The experimental section aims at evaluating the ball candidate validation performance in presence of different preprocessing techniques.

For the SIFT methodology, the codebook of visual words was built by quantizing (using k-means algorithm [22]) the 128 long feature vectors relative to the detected points of interest in 50 patches containing fully visible balls under good light conditions.

For all the preprocessing methodologies, the naive Bayes classifier was built by considering the 50 above-considered patches and 50 new patches (extracted at the beginning of the acquisition session then with good lighting conditions) which did not contain the ball.

The 100 resulting feature vector elements were finally fed into the naive Bayes classifier that estimated the conditional probability distribution to have a ball instance given a feature configuration. The experimental tests were then carried out



(a) One acquired Image (b) Hough Space

(c) two candidate ball regions

Figure 7: (a) One acquired image, (b) the corresponding Hough space, and two candidate ball regions. The region on the left has been properly centered on the ball, whereas the one on the right is relative to a nonball object in the scene corresponding to a high value in the Hough accumulation space.

considering all the patches that were not used in the learning phase. These patches representations were supplied as input to the learned naive Bayes classifier that associates to each representation the probabilities $P(\text{Ball} \mid V)$ and $P(\text{NoBall} \mid V)$, that is the probabilities that the corresponding patch contained a ball or not. Finally, the patches in input were then classified on the basis of the argmax$[P(\text{Ball} \mid V), P(\text{NoBall} \mid V)]$.

In Table 1, ball classification detection rates using SIFT and different number of clusters in the k-means algorithm, are shown.

Table 2 summarizes the ball recognition results obtained by comparing different feature extraction algorithms on the whole dataset. In the first column of Table 2 are itemized the used feature extraction algorithms, and in the second one are shown the number of coefficients extracted from the relative technique that are input of the probabilistic classifier. From the third to sixth columns are presented, respectively, the values of correct detection of the ball (TP: true positive), the values of the errors in the ball detection (FN: false negative), the values of correct detection of no ball (TN: true negative), and finally the values of errors in detection of ball in the candidate regions in which it does not exist (FP: false positive). In the last column of Table 2 the overall detection performance of each methodology is shown. For all of the compared feature extraction methodologies, we have used the same training set composed of 50 examples of fully visible balls and 50 examples which did not contain the ball. All the training patches are extracted at the beginning of the acquisition session, with good lighting conditions. Except for the SIFT algorithm that is detailed in the first part of this section, we have directly supplied, as input, the extracted coefficient, in the learned probabilistic classifier. Table 2 gives very encouraging results: even if all training examples consisted only of patches extracted from images acquired under good light conditions, the tests demonstrated a good capability of quite all proposed approaches to rightly recognize the ball instance also under very different and challenging conditions (poor light condition, ball occluded or deformed). At the same time, no-ball patches were well classified, avoiding a huge number of false ball validations. Six patches extracted from the test set are reported in Figure 9: in the first row, three patches extracted in correspondence with no-ball objects in the scene (a bench, the net of the goal, and the head of a person); in the second row, three patches having similar appearance but containing the ball.

However, if Table 2 is deeply observed, it should be noted that the best performance, in terms of ball detection rate, is obtained by using wavelet decomposition (in particular Daubechies family slightly better than Haar family). The used approximation coefficients are obtained applying low-pass filter in both directions (vertical and horizontal) till the third level of decomposition. The worst method, in this application context, is the SIFT combined with the bag-of-words (BoW) representation.

However, SIFT-based approach outperforms the others in the classification of partially/heavy occluded ball instances or in case of very bad lighting conditions.

This is proved in Table 3 where the different preprocessing methodologies are compared on a test subset consisting of 116 examples (only partially/heavy occluded or poor lighted balls).

In this case, SIFT representation allowed the probabilistic classifier to obtain the best result in terms of detection rate.

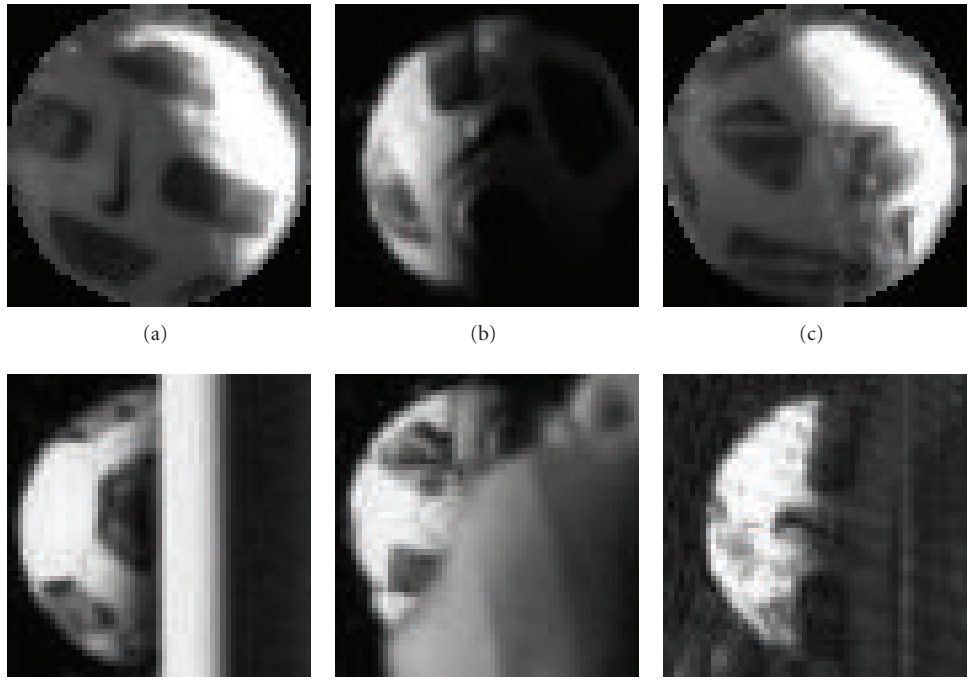


Figure 8: Different ball appearances observed during the experimental phase: (a) ball acquired with good lighting conditions; (b) ball acquired with poor lighting conditions; (c) ball faded by the net; (d) ball partially occluded by the goal post subset; (e) ball heavy occluded by a person in the scene; (f) ball deformed during impact.

Table 1: Ball classification performance using SIFT and different number of clusters in the k-means algorithm on whole dataset.

SIFT descriptors length	TP	FN	TN	FP	Detection rate (%)
10	579	658	988	249	63.33%
20	621	616	1059	178	67.90%
30	558	579	1056	181	65.23%
40	519	718	1091	146	65.07%

Table 2: Ball recognition results using different features on the whole dataset.

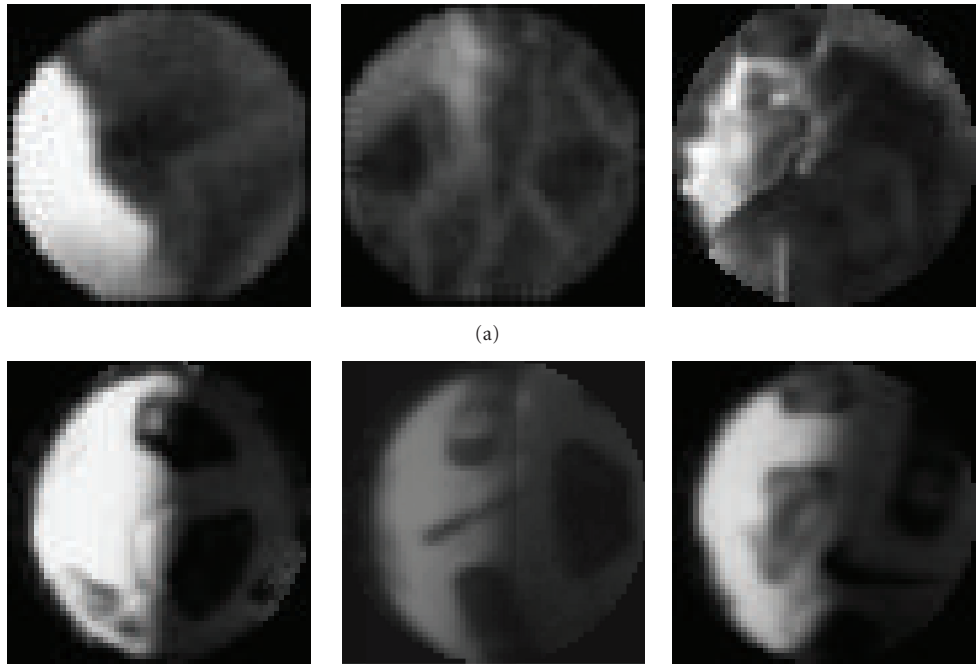
Ball features	Number of input	TP	FN	TN	FP	Detection rate (%)
Wavelet HAAR	64 (8×8)	1179	58	1028	209	89.20%
Wavelet DB3	121 (11×11)	1166	71	1060	177	89.97%
Histogram	256	1174	63	958	279	86.17%
SIFT (BoW)	20	621	616	1059	178	67.90%
PCA	30	999	238	890	347	76.35%

Also histogram representation gave satisfying results, whereas wavelet and PCA representation leaded to very inadequate classification rate.

This leads to a consideration: in a soccer match, the ball, in addition to be often occluded or faded, could be under different lighting conditions depending on the weather conditions or the position into the playing field and then it is better to choose a preprocessing technique that keeps the classifier under acceptable classification rate also when the testing patches differ from those used in the learning phase.

Moreover, computation aspects should be considered in choosing the appropriate preprocessing methodology: from this point of view, wavelet and SIFT methodologies take much more time than histogram calculation and PCA.

Summing up, the histogram extraction methodology is the most suited for the considered issue.



(a)

(b)

Figure 9: Some examples of ball (a) and no-ball (b) patches: many objects in the scene can appear, in particular conditions, very similar to the ball.

Table 3: Ball recognition results using different Features on the dataset (III) containing partially and heavily occluded balls.

Ball features	TP	FN	Detection rate (%)
Wavelet HAAR	66	50	66/116 (56.89%)
Wavelet DB3	52	64	52/116 (44.82%)
Histogram	76	40	76/116 (65.51%)
SIFT (BoW)	86	30	86/116 (74.13%)
PCA	40	76	40/116 (34.48%)

It allows the system to recognize the ball instances in real time with a good classification rate even under challenging conditions (occlusions, poor lighting conditions, etc.).

6.1. Computational Remarks. The proposed algorithm was run on personal computer with an Intel i3 CPU 3.20 GHz and 4 GB RAM, the available Microsoft Visual C++ 2005 implementation; it takes about 2 msec to detect the ball in each high-resolution input image (i.e., max 500 frames can be processed in each second). This is a very desirable result considering that visual technologies aimed at assisting the referee team in deciding if the ball really has crossed the goal line (the algorithm described in this work is part of that system) require the analysis of up to 500 frames per second, in order to cope with the high ball speed (that could reach 130 km/h).

7. Conclusions

This paper presented different approaches to recognize soccer ball patterns through a probabilistic analysis. It investigated the different feature extraction methodologies applied in ball soccer detection.

Experimental results on real ball examples under challenging conditions and a comparison with some of the more popular feature extraction algorithms from the literature demonstrated the suitability of the classical histogram approach in this application context in which real-time performances are a constraint.

Future works are addressed to study new algorithms which cope with occluded balls, taking into account overall ball detection performances and computational time constraints.

Moreover, even if some images have been acquired in rainy days, more intensive tests concerning effects of inclement weather (rain, snow, fog) on the algorithms will be performed.

Acknowledgments

The authors thank Liborio Capozzo and Arturo Argentieri for technical support in the setup of the devices used for data acquisition.

References

[1] M. H. Hung, C. H. Hsieh, C. M. Kuo, and J. S. Pan, "Generalized playfield segmentation of sport videos using color features," *Pattern Recognition Letters*, vol. 32, no. 7, pp. 987–1000, 2011.

[2] K. Choi and Y. Seo, "Automatic initialization for 3D soccer player tracking," *Pattern Recognition Letters*, vol. 32, no. 9, pp. 1274–1282, 2011.

[3] X. Gao, Z. Niu, D. Tao, and X. Li, "Non-goal scene analysis for soccer video," *Neurocomputing*, vol. 74, no. 4, pp. 540–548, 2011.

[4] A. Watve and S. Sural, "Soccer video processing for the detection of advertisement billboards," *Pattern Recognition Letters*, vol. 29, no. 7, pp. 994–1006, 2008.

[5] J. Liu, X. Tong, W. Li, T. Wang, Y. Zhang, and H. Wang, "Automatic player detection, labeling and tracking in broadcast soccer video," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 103–113, 2009.

[6] M. Leo, N. Mosca, P. Spagnolo, P. L. Mazzeo, T. D'Orazio, and A. Distante, "A visual framework for interaction detection in soccer matches," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 24, no. 4, pp. 499–530, 2010.

[7] A. Ekin, A. M. Tekalp, and R. Mehrotra, "Automatic soccer video analysis and summarization," *IEEE Transactions on Image Processing*, vol. 12, no. 7, pp. 796–807, 2003.

[8] Z. Theodosiou, A. Kounoudes, N. Tsapatsoulis, and M. Milis, "MuLVAT: a video annotation tool based on XML-dictionaries and shot clustering," *Lecture Notes in Computer Science (ICANN)*, vol. 5769, no. 2, pp. 913–922, 2009.

[9] T. D'Orazio and M. Leo, "A review of vision-based systems for soccer video analysis," *Pattern Recognition*, vol. 43, no. 8, pp. 2911–2926, 2010.

[10] V. Pallavi, J. Mukherjee, A. K. Majumdar, and S. Sural, "Ball detection from broadcast soccer videos using static and dynamic features," *Journal of Visual Communication and Image Representation*, vol. 19, no. 7, pp. 426–436, 2008.

[11] X. Yu, H. W. Leong, C. Xu, and Q. Tian, "Trajectory-based ball detection and tracking in broadcast soccer video," *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1164–1178, 2006.

[12] J. Ren, J. Orwell, G. A. Jones, and M. Xu, "Tracking the soccer ball using multiple fixed cameras," *Computer Vision and Image Understanding*, vol. 113, no. 5, pp. 633–642, 2009.

[13] T. D'Orazio, C. Guaragnella, M. Leo, and A. Distante, "A new algorithm for ball recognition using circle Hough transform and neural classifier," *Pattern Recognition*, vol. 37, no. 3, pp. 393–408, 2004.

[14] M. Leo, T. D'Orazio, P. Spagnolo, P. L. Mazzeo, and A. Distante, "Sift based ball recognition in soccer images," in *Proceedings of the 3rd international conference on Image and Signal Processing (ICISP '08)*, pp. 263–272, 2008.

[15] S. Mallat, *A Wavelet Tour of Signal Processing*, AP Professional, London, UK, 1997.

[16] I. T. Jolliffe, *Principal Component Analysis*, Springer, New York, NY, USA, 2002.

[17] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[18] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.

[19] I. Tošić and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, no. 2, pp. 27–38, 2011.

[20] F. Colas and P. Brazdil, "Comparison of SVM and some older classification algorithms in text classification tasks," *IFIP International Federation for Information Processing*, vol. 217, pp. 169–178, 2006.

[21] P. A. Flach and N. Lachiche, "Naive Bayesian classification of structured data," *Machine Learning*, vol. 57, no. 3, pp. 233–269, 2004.

[22] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, vol. 1, pp. 281–297, 1967.