Players and Ball Detection in Soccer Videos Based on Color Segmentation and Shape Analysis

Yu Huang, Joan Llach, and Sitaram Bhagavathy

Thomson Corporate Research,
2 Independence Way, Princeton, NJ 08540, USA
{yu.huang2, joan.llach, sitaram.bhagavathy}@thomson.net

Abstract. This paper proposes a scheme to detect and locate the players and the ball on the grass playfield in soccer videos. We put forward a shape analysis-based approach to identify the players and the ball from the roughly extracted foreground, which is obtained by a trained, color histogram-based playfield detector and connected component analysis. We employ Euclidean distance transform to extract skeletons for every foreground blob, and then perform shape analysis to remove false alarms (non-player and non-ball blobs) and cut-off the artifacts (mostly due to playfield lines) based on skeleton pruning and reverse Euclidean distance transform. Results are given to demonstrate the proposed algorithm works well in soccer video clips.

Keywords: Object detection, shape analysis, distance transform, skeleton pruning.

1 Introduction

The players and the ball are the most important objects in soccer videos. Detection and tracking of them are motivated by various applications, such as event detection, tactics analysis, automatic summarization and object-based compression [17].

Methods of locating the ball as well as players in soccer videos can be split in two groups: the first group makes use of fixed cameras (usually calibrated in advance) in a controlled environment [9]; the second group uses only regular broadcasting videos [3]. While the former can provide better performance, the latter is more flexible. In this paper, we focus on those efforts made in the second group.

In [3], a pioneering work in soccer video analysis was reported. The ball is detected by both its chromatic and morphological features. The ball is assumed to be among the white regions and its circularity (an indicator of resemblance of the shape to a circle) bigger than a given threshold. For the players, their uniform colors are recognized based on peaks (other than green) in the color histogram.

In [11], the authors propose a modified Circle Hough Transform to detect the ball on selected frames. Their method, however, cannot handle occlusions and requires the ball to be homogeneous, which greatly limit its application. In [10], there is an application of ball detection for video compression. Color histogram back projection

N. Sebe, Y. Liu, and Y. Zhuang (Eds.): MCAM 2007, LNCS 4577, pp. 416–425, 2007.

and intensity template matching are used to detect the ball. It is claimed that trajectory knowledge is used for prediction, but no details are given in the paper.

Unlike the object-based algorithms, a trajectory-based algorithm was put forward in [18] for detecting and tracking the ball. In this off-line framework, how to obtain the ball candidates, remove the false alarms and verify the candidates becomes critical. Similarly, trajectory constraints are also used in [7], where the difference is that a Viterbi algorithm rather than a Kalman filter is employed to verify the ball's trajectory candidates.

Since soccer is a spectator sport, the play fields, the lines, the ball and the clothing of the players are designed to be visually distinctive in color. Therefore, some approaches tried to alleviate detection difficulties by finding the grass playfield first [1, 8, 15, 16], using color segmentation and post-processing with morphological operations, such as connected component analysis, in order to limit the search area. However, how to remove false alarms (e.g. small areas that look like the ball) and cut-off artifacts, especially when player blobs merge with the field lines, is still a challenging problem.

A simple way to represent the playfield color is using a constant mean color value that is obtained through prior statistics over a large data set. The color distance between a pixel and the mean value of the field is used to determine whether it belongs to the field or not [15]. Since the soccer field is roughly green colored, the hue component defined in Smith's hexagonal cone model can be used to detect green-colored pixels within a certain range [16].

Statistically, single Gaussian or mixture of Gaussian (MoG) can be learned to represent a color model of the playfield and adapted by the Expectation-Maximization (EM) algorithm incrementally [8]. A non-parametric color model for the playfield is the histogram, where the dominant color is typically used for playfield detection assuming the field has a uniform shade of green and occupies the largest area in each frame [1].

More efforts in shape analysis are made for foreground blobs obtained by background subtraction, but most of the techniques rely on silhouette and region features [4]. Recently, skeleton analysis has been employed for object recognition [5, 12], shadow removal [13] and shape filtering [2, 14].

This paper proposes a shape analysis-based approach to identify the players and the ball from the roughly extracted foreground, which is obtained by a trained color histogram-based playfield detector and connected component analysis. We employ Euclidean distance transform to extract skeletons for every foreground blob, and then perform shape analysis to remove false alarms (non-players and non-ball blobs) and cut-off the artifacts (mostly due to playfield lines) based on skeleton pruning and reverse Euclidean distance transform.

2 Playfield Pixel Detector

Soccer is normally played on a grass field. In order to detect the players and the ball, a useful first step is to detect the pixels that form the playfield. In our work, we employ a simple but powerful histogram learning technique to detect the playfield pixels. This framework was proposed by Jones et al. [6] for skin detection. Color models are learned for playfield pixels and non-playfield (background) pixels, using a training set of soccer videos. The color model is an RGB color histogram with *N* bins per channel in the RGB color space.

The playfield and non-playfield models are learned as follows. The pixels in the training set videos are labeled as playfield pixels and non-playfield pixels, either manually or using a semi-supervised method. Based on its RGB color vector, each labeled playfield pixel is placed into the appropriate *rgb* bin of the playfield histogram. A similar process is carried out for the pixels labeled as non-playfield. The histogram counts are converted into a discrete probability distribution:

$$P(rgb|\,\text{playfield}) = \frac{f(rgb)}{T_f}, \quad P(rgb|\,\text{non-playfield}) = \frac{n(rgb)}{T_n}, \tag{1}$$

where $f[rgb]$ is the pixel count in bin *rgb* of the playfield histogram, $n[rgb]$ is the corresponding count from the non-playfield histogram, and $T_{f,}$ and T_n are the total counts contained in the playfield and non-playfield histograms, respectively.

A playfield pixel classifier is derived through the standard likelihood ratio approach [6]. A particular RGB value is labeled playfield if

$$\frac{P(rgb|\,\text{playfield})}{P(rgb|\,\text{non-playfield})} \geq \theta, \tag{2}$$

where $\theta \geq 0$ is a threshold which can be adjusted to trade-off between correct detections and false positives.

The number of bins per channel, *N*, and the detection threshold, θ, can be chosen based on the receiver operating characteristic (ROC) curve. The ROC curve shows the relationship between correct detections and false detections as a function of the detection threshold θ. ROC curves are computed using a test set of soccer videos (usually the labeled data set is separated into a training set and a test set).

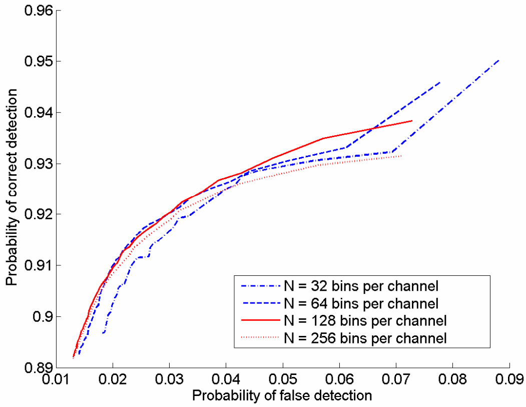

Fig. 1. ROC curves for playfield pixel classification with respect to *N*

Fig. 1 shows a family of ROC curves for different values of *N*. The *y*-axis gives the fraction of pixels labeled as playfield that were classified correctly, and the *x*-axis gives the fraction of non-playfield pixels that were wrongly classified as playfield. The higher the area under an ROC curve, the better the overall detection performance. Accordingly, the color histogram model with $N = 128$ bins is found to give the best results. After choosing *N*, a suitable operating point is selected on the corresponding ROC curve, to give the "best" (application dependent) trade-off between the probability of correct detections and the probability of false positives. The value of θ corresponding to the operating point is then used for playfield pixel classification through (2). For our experiments, we have chosen $\theta = 0.1$ and $N = 128$, and the corresponding probability of correct detection is 0.935 and the probability of false detection is 0.057.

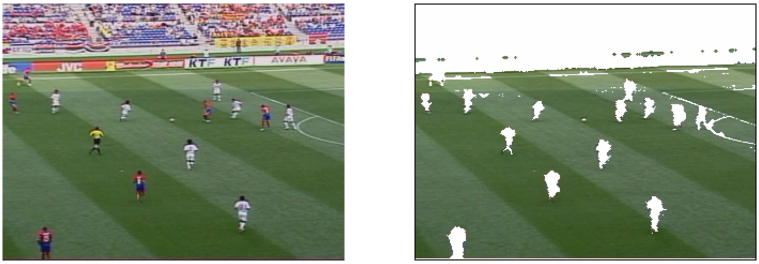

Fig. 2. Playfield detection result: the original frame and the detected playfield pixels

Fig. 2 shows the result of the playfield detection method described above. The original frame is displayed on the left. The detected playfield pixels are shown on the

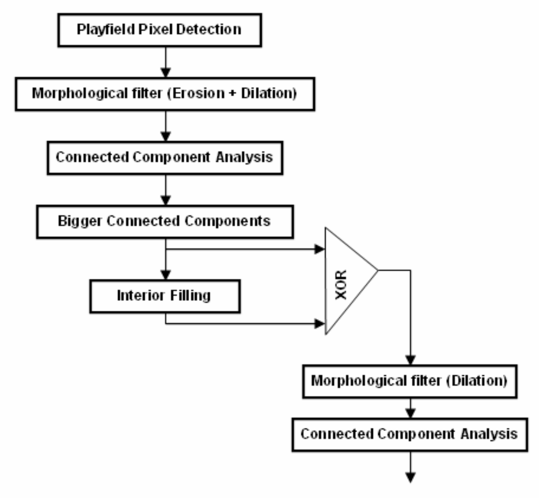
Playfield Pixel Detection
Morphological filter (Erosion + Dilation)
Connected Component Analysis
Bigger Connected Components
Interior Filling
XOR
Morphological filter (Dilation)
Connected Component Analysis

Fig. 3. Foreground blob extraction in soccer video

right. A morphological opening operation was applied after detecting playfield pixels in order to remove small false positive blobs. It can be observed from Fig. 2 that the method is successful in detecting playfield pixels under significant variations in color (dark and light stripes) and illumination (sunlit and shaded areas).

3 Playfield Extraction and Foreground Blob Detection

Fig. 3 illustrates the diagram of foreground extraction. Going through the playfield pixel detector, each frame yields a binary mask. Morphological filtering (erosion and then dilation) is applied to eliminate the noise.

Connected components analysis (CCA) scans the binary mask and groups its pixels into components based on pixel connectivity. The playfield is supposed to consist of several connected components, each of them bigger in size than a given area threshold. Thus, the playfield mask is obtained by filling the interior of each bigger component.

Ideally, the non-playfield pixels inside the extracted playfield areas should be the foreground pixels that can be grouped into different foreground blobs by CCA. Since the ball is typically quite small (less than 30 pixels in size for QVGA content), we cannot use erosion before grouping pixels into different blobs, but we can still use dilation. Therefore, noisy areas have to be removed later by shape analysis and true object appearance model.

4 Shape Analysis for Foreground Cleaning

Region properties for each blob are given by a few basic shape descriptors, such as perimeter P, area A, major/minor axes C_L/C_S, roundness $F = P^2/(4\pi A)$ and eccentricity $E = C_L / C_S$.

Skeletons provide an intuitive, compact representation of a shape. One of its important features is separation of the shape's topological properties from its geometric properties. Here we employ distance transform to extract skeleton and generate additional shape descriptors for use.

4.1 Distance Transform

Given a foreground binary mask W of size *m x n*, $\overline{W}$ denotes the complementary of W, i.e. the set of background pixels. Its *squared distance transform* (SDT) [2] is given by *H = {h(x,y)},* i.e.

$$h(x,y) = \min\{(x-i)^2 + (y-j)^2; 0 \le \mathrm{i} < \mathrm{m}, 0 \le j < n, (i,j) \in \overline{W}\}. \quad (3)$$

Given a set of foreground pixels *V* and a picture *F = {f(x,y)}* of size *m x n*, such that *f(x,y)* is set to the SDT value $h(x,y)$ when the pixel *(x,y)* belongs to *V* and 0 otherwise. Then *reverse Euclidean distance transform* (REDT) [2] of *V* consists in obtaining the set of points *W* such that

$$W = \{(x,y) \mid \max\{f(i,j) - (x-i)^2 - (y-j)^2\} > 0, \quad 0 \le \mathrm{i} < \mathrm{m}, 0 \le j < n, (i,j) \in F\}. \quad (4)$$

Thus, if we compute the map $H^* = \{h^* (x,y)\}$ such that

$$h^*(x,y) = \max\{f(i,j) - (x-i)^2 - (y-j)^2, 0 \le \mathrm{i} < \mathrm{m}, 0 \le j < n, (i,j) \in F\}, \quad (5)$$

then we obtain W by extracting from H^* all pixels of strictly positive values.

4.2 Skeleton Descriptors

Given SDT of the shape $H = \{h(x,y)\}$, its skeleton *Ske* is defined by [2]

$$Ske = \{(x,y) \mid \exists(i,j), (x-i)^2 + (y-j)^2 < h(x,y),$$

$$\text{AND} \max_{(u,v)}\{h(u,v) - (x-u)^2 - (y-v)^2\} = h(x,y) - (i-x)^2 - (j-y)^2\}. \quad (6)$$

Comparing with the definition of REDT, we see that applying REDT to the SDT of the shape, in which only the upper envelope elliptic paraboloids are marked, can yield the *Ske*. In Fig. 4, a skeleton of player blob is shown.

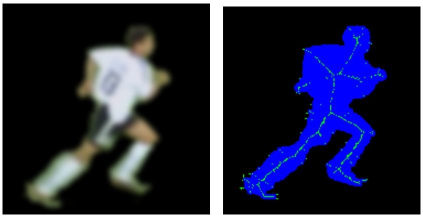

Fig. 4. Skeleton results: the player blob and its skeleton (in green)

As explained by [2], the Power diagram is a special kind of Voronoi diagram and the Power labeling assigns to each point of the plane the index of the cell containing it in the Power diagram (more details are given in [2]). When we apply REDT to get the skeleton, simultaneously the Power labeling for each skeleton point is generated as well.

Eventually, we get those additional descriptors for shape analysis as: length of the skeleton L_s, covering C_{ske}, (i.e. the number of pixels in Power labeling associated with the skeleton), maximal thickness of the skeleton $d_{\max}$, (i. e. radius of the maximal ball which covers the blob, correspondingly the center of the maximal ball is one of the skeleton pixel) elongation G=$A/(2d_{\max})^2$ and aspect ratio T=$L_s / d_{\max}$ etc.

4.3 White Pixel Proportion in Skeleton's Covering

The ball and the field lines in wide view shots are nearly white. For each skeleton point, the white pixel proportion in its covering (Power labeling) is a good indicator for artifact detection. A simple method to identify white pixels in each foreground blob is

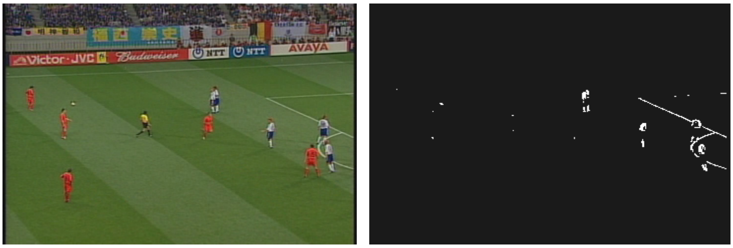

Fig. 5. White pixel detection: the original frame and the detected white pixels

$$X(x,y)=\begin{cases}1, & (r(x,y)-1/3)^2+(b(x,y)-1/3)^2<a \text{ AND } \mathrm{I(x,y)>b} \\ 0, & \textit{otherwise}\end{cases}, \tag{7}$$

where *r(x,y)* and *b(x,y)* are normalized red and blue components for pixel *(x,y), I(x,y)* denotes the intensity value in the range [0,255], and *X(x,y)* is the white pixel mask. Fig. 5 shows results of white pixel detection. Below we use it for ball detection and player blobs' skeleton pruning.

4.4 Ball and Player (Referee) Localization

Apparently, there is a predefined range for the ball blob's area *A* according to the camera configuration. Meanwhile, the proportion of detected white pixels *X(x,y)* by (4) in the blob indicates the possibility of being a ball. Roundness *F* and eccentricity *E* for a blob candidate should be close to 1.0, different from disconnected segments of field lines. Eventually, detection of the ball blob is carried out by

$$B=\begin{cases}1, & p_W > \mathrm{r_w} \text{ AND } \mathrm{ar_{min}} < \mathrm{A} < \mathrm{ar_{max}} \text{ AND } F < f_{\max} \text{ AND } \mathrm{E} < e_{\max} \\ 0, & \textit{otherwise}\end{cases}, \tag{8}$$

where $p_W = C\{(x,y)\,|\,X(x,y)=1\}/A$, C{.} is counting the number of white pixels. (All the thresholds are empirically determined.)

For an isolated player (or referee) blob, its shape also can also be approximated by an ellipse, limiting its elongation and aspect ratio. These constraints can help removing some fragments of field lines. In addition, since field lines look nearly white in color, so can use the proportion of white pixels as an index for non-player blobs. Similarly, an area range for the player (or referee) blob is predefined according to the camera configuration.

4.5 Skeleton Pruning

When the player (referee) blob merges with fragments of field lines, shape analysis becomes complex. We propose a skeleton pruning-based method to cut-off those artifacts.

First, the width of a field line is less than that of a player body in the wide view shot; so at every skeleton point we first check whether its thickness is small compared with the maximum skeleton thickness, i.e. smaller than $c \cdot d_{\max}$, where c is thickness factor. If it is true, then we calculate average values of RGB components for all pixels covered by this skeleton point (Power labeling). The observed result is that the average RGB components are close to those of the white color when this skeleton point corresponds to a fragment of field lines.

To avoid excessive pruning, we add the distance constraints: the pruned skeleton is relatively far away from the human body centroid which corresponds to the skeleton point with the maximal covering value, i.e. its horizontal distance is bigger than $d_h \cdot d_{\max}$ and its vertical distance $d_v \cdot d_{\max}$, where d_v, d_h are distance factors.

5 Experimental Results

We test our proposed algorithm with three video clips (about 300 frames for each clip) and the results are quite encouraging. Some detection results are shown in Fig. 6-8. For each figure, the extracted grass field is given at the bottom left, the initial segmented foreground is shown at bottom right, the cleaned foreground blobs by shape analysis and skeleton pruning (skeleton is in green and the blob is in blue) are displayed at top left and the final detected ball and players is shown at top right, where each detected object is enclosed by an ellipse in yellow.

It is shown most of false alarms from field lines are removed. Artifacts in player blobs due to merging with field lines are cut-off. Our proposed method overperforms previous methods as [1, 16]. The method in [1] can't detect the ball and have to set the ball's location manually; it can't remove artifacts in players' localization and require

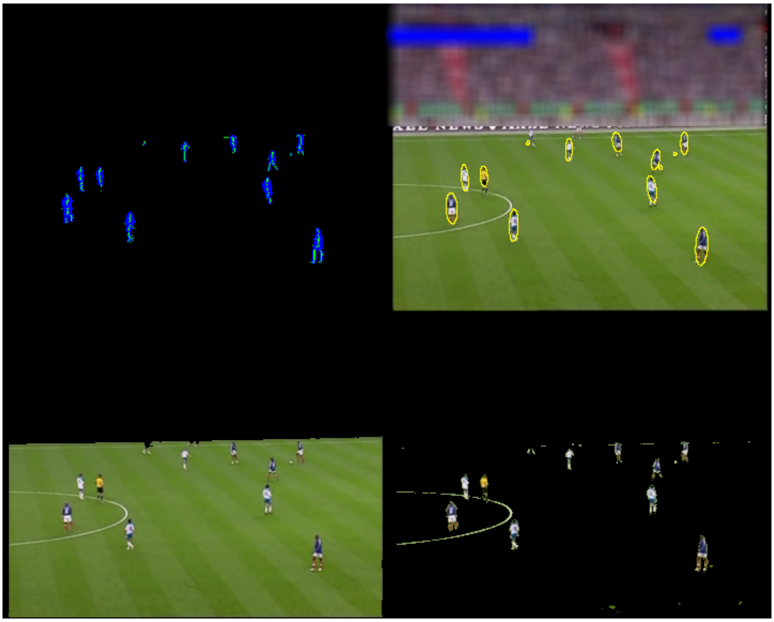

Fig. 6. Detection results in one frame of video 1

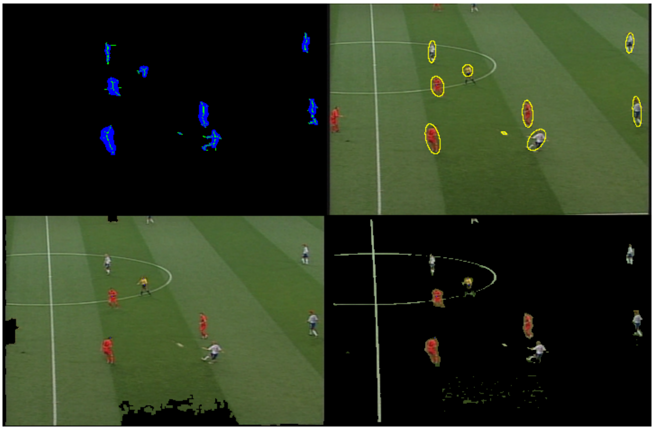

Fig. 7. Detection results in one frame of video 2

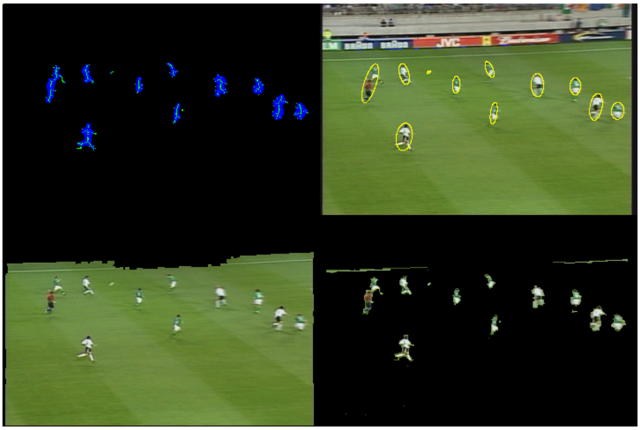

Fig. 8. Detection results in one frame of video 3

color histogram models learned beforehand to detect players' presence. The system in [1] focus on player detection only and it fails in finding players when they merge with field lines, though tracking modules are employed to handle overlapping/occlusions.

However we still find in our results, players standing at the border of the field might be missed since we extract the playfield region first; sometimes the player's feet or legs are not extracted by the playfield detector because some of their pixels look field-green due to the course graininess.

6 Conclusions and Future Work

In this paper, a shape analysis-based soccer ball and the players detection method has been proposed. We propose a learned color histogram model to detect the playfield pixels and group them into a playfield region. Then, the foreground blobs are extracted with morphological processing. Shape analysis and skeleton pruning are performed to remove false alarms (non-players/referees and non-ball) and cut-off the artifacts (mostly due to playfield lines).

In future, we will work on how to separate the players (referees) when they overlap and partially occlude each other. Besides, we will consider incorporating the proposed detector with a ball tracking system and a player/referee tracking system.

References

1. Choi, S., Seo, Y., et al.: Where are the ball and players? Soccer game analysis with color-based tracking and image mosaic. In: Int. Conf. on Image Analysis and Processing (September 1997)
2. Coeurjolly, D., Montanvert, A.: Optimal Separable Algorithms to Compute the Reverse Euclidean Distance Transformation and Discrete Medial Axis in Arbitrary Dimension. IEEE T-PAMI 29(3), 437–448 (2007)
3. Gong, Y., Sin, L.T., Chuan, C.H., Zhang, H., Sakauchi, M.: Automatic parsing of TV soccer programs. In: Proc. Multimedia Computing & Systems, pp. 167–174 (1995)
4. Haritaoglu, I., Harwood, D., Davis, L.S.: Hydra: multiple people detection and tracking using silhouettes. In: 2nd IEEE Workshop on Visual Surveillance (1999)
5. He, L., Han, C.Y., Wee, W.G.: Object recognition and recovery by skeleton graph matching. In: IEEE ICME'06 (2006)
6. Jones, M., Rehg, J.M.: Statistical Color Models with Application to Skin Detection. In: IEEE CVPR'99, June 1999, pp. 274–280 (1999)
7. Liang, D., et al.: A scheme for ball detection and tracking in broadcast soccer video. In: PCM 2005, Korea (2005)
8. Liu, Y., Jiang, S., Ye, Q., Gao, W., Huang, Q.: Playfield Detection Using Adaptive GMM and Its Application. In: IEEE ICASSP '05, March 2005, pp. 421– 424 (2005)
9. Needham, C.J., Boyle, R.D.: Tracking multiple sports players through occlusion, congestion and scale. In: BMVC'01 vol. 1, pp. 93–102 (2001)
10. Nementhova, O., Zahumensky, M., Rupp, M.: Preprocessing of ball game video sequences for robust transmission over mobile network. In: CDMA International Conference (CIC), Seoul (2004)
11. D'Orazio, T., et al.: A Ball Detection Algorithm for Broadcast Soccer Image Sequences. In: IAPR ICPR'02 (2002)
12. Ozcanli, O.C., Tamrakar, A., Kimia, B.B.: Augmenting Shape with Appearance in Vehicle Category Recognition. In: IEEE CVPR'06, June 2006, pp. 935–942 (2006)
13. Renno, J., Orwell, J., Thirde, D., Jones, G.A.: Shadow Classification and Evaluation for Soccer Player Detection. In: BMVC'04, September 2004, Kingston, (2004)
14. Tam, R.C., Heidrich, W.: Feature-Preserving Medial Axis Noise Removal. In: ECCV'02, pp. 672–686 (2002)
15. Tong, X., et al.: An Effective and Fast Soccer Ball Detection and Tracking Method. In: ICPR'04, pp. 795–798 (2004)
16. Utsumi, O., Miura, K., Ide, I., Sakai, S., Tanaka, H.: An object detection method for describing soccer games from video. In: IEEE ICME '02 (2002)
17. Wang, J.R., Parameswaran, N.: Survey of Sports Video Analysis: research issues and applications. In: Pan Sydney Area Workshop on Visual Information Processing (VIP03) (2003)
18. Yu, X., Xu, C., Tian, Q., Leong, H.W.: A ball tracking framework for broadcast soccer video. In: ICME'03 (2003)