Image and Vision Computing 27 (2009) 469–479

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Automatic soccer players tracking in goal scenes by camera motion elimination

Seyed Hossein Khatoonabadi, Mohammad Rahmati *

Computer Engineering Department, Amirkabir University of Technology, Hafez Street, Tehran, Iran

ARTICLE INFO

Article history:
Received 7 August 2006
Received in revised form 16 February 2008
Accepted 29 June 2008

Keywords:
Grass field extraction
Lines tracking
Membership identification
Occlusion reasoning
Players tracking
Soccer goal scenes

ABSTRACT

In this paper, we propose a novel and effective algorithm for tracking soccer players in goal scenes, by eliminating fast camera motions effect through the correspondence between line marks in soccer field model and image sequences. The proposed algorithm comprises four steps. At the first step, we introduce an automatic grass field extraction algorithm that is tested for various soccer video types and conditions. The field line marks, which are used to locate the players, are detected at the second step using Hough Transform and tracked in all frames by predicting their positions using Kalman Filter. At the third step, we introduce a novel approach for estimating the players' positions during the course of tracking players. We estimate a player's position in the current frame by observing his position in the old frame within the soccer field model, using Perspective Transformation of the old frame to the real world coordinate system, and then by projecting back the obtained player's position into the current frame. At the final step, the players are tracked by applying the region-based detection algorithm, the Histogram Back-Projection algorithm or a combination of the Merge-Split approach and the Template-Matching algorithm around the estimated positions, depending on whether or not the occlusion has occurred and if yes, how it has occurred. Then their memberships are identified by an algorithm that employs both appearance and spatial information of the players. Image sequences of different soccer games were captured from different sources, and all experiments were performed off-line. The results of our experimentations show that our algorithm is highly robust to occlusion, different soccer field colours, different lights such as sunlight and spotlight, shadows of players and fading the whole screen due to fast camera movements.

1. Introduction

In the field of machine vision, tracking moving objects in image sequences is an interesting and a difficult problem. Tracking moving objects is applied in different application areas including human body tracking, traffic monitoring, sport match analyzing and medical image processing.

In most multiplayer games like soccer, interesting and valuable events occur only in a small portion of the game period especially at goal scenes. Furthermore, coaches and spectators are interested in the goal scenes for different reasons. Due to the importance of these events in a soccer game, we address the problem of soccer players tracking only in goal scenes in this paper.

The research on various aspects of soccer video analysis has been growing in the recent years. Gong et al. [1] introduced a system to classify a sequence of soccer frames into various categories such as shot at left goal, top-left corner, right penalty area, midfield, etc. The foregone introduced system is based on soccer field model, ball, players and motion vectors. Soccer shots are classified into three kinds of views including long shots, in-field medium shots and out-of-field or close-up shots [2]. Xu et al. [3] detected that the game is whether in play status or in break. Highlighted soccer scenes like goal scenes are extracted by object features in [4], by cinematic features in [5,6] and by audio and video information in [7,8]. Utsumi et al. [9] detected and tracked players based on their colour rarity and local edge properties. Seo et al. [10] and Yoon et al. [11] proposed systems for mosaicing a sequence of soccer images and tracking players based on Template-Matching algorithm. The players' positions at the next frame are predicted by Kalman Filter in [10] and by nearest neighbourhood in [11]. Also, the next players' positions are predicted by assuming that a player moves at a constant velocity during a short period of time and then tracked by extracting players' colourful uniforms in [12]. Vandenbroucke et al. [13] and Lefevre et al. [14] used active contour concept to track soccer players. At the first step of these researches, the players are separated into two teams based on their colour characteristics. Then, players of each team are extracted by corresponding deformable snakes. Both [13] and [14], however, are unable to separate some players where they are in close distance of each other (in [13] this happens when players belong to the same team). Some tracking systems such as [15,16] proposed to use multiple cameras in their projects.

* Corresponding author. Tel.: +98 21 6454 2741; fax: +98 21 649 5521.
E-mail addresses: hossein.khatoonabadi@gmail.com (S.H. Khatoonabadi), rahmati@aut.ac.ir, mohammad.rahmati@gmail.com (M. Rahmati).

0262-8856/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2008.06.015

Similarly, some researches are reported in other multiplayer games such as squash [17], handball [18,19] and football [20,21]. Intille and Bobick [20,21] tracked players by defining closed-world as a space-time region of an image sequence where contextual information like the number and the type of objects within the region is assumed to be known.

To the best of our knowledge, no previous work on goal scenes of a soccer game has been reported that considers fast camera movement, complex and dynamic shot and rapid moving players. Applying conventional algorithms for tracking players in soccer games that contain small and non-rigid objects in low resolution video image sequences usually fails. The failure is due to the fact that, they have only used low-level information and have not exploited the global information such as field model.

In our proposed algorithm, for the first frame of a shot, the grass field region is extracted by some low level image processing using the dominant colour components of the soccer images. Then, the players are detected by a region-based detection algorithm in the field region. Soccer field line marks are detected and tracked by employing an approach, which is a combination of Hough Transform and Kalman Filter. Our proposed algorithm for tracking players in the subsequent frames consists of the following global steps:

(1) Find players' positions in the old frame on a pre-specified model (i.e. soccer field coordinate system).
(2) Find the players' positions in the current frame by projecting back the obtained players' positions in the previous step into the current frame.
(3) Detect players by a region-based detection algorithm in cases in which there are no occlusions. Otherwise, Template-Matching, Merge-Split or Histogram Back-Projection algorithm is used to resolve occlusion, depending on how occlusion occurs.

The novelty of our proposed approach relies mainly on prevailing over the large changes between two consecutive frames in the image space, created due to fast camera movement, as the cameraman tries to follow the position of the ball. To achieve this goal, we calculate these changes in the field model, which results in small changes. As a result, by eliminating camera motions effect, estimation of players' positions will be more accurate.

The rest of this paper is organized as follows. In the next section, the grass field extraction algorithm is introduced. Detecting and tracking soccer field line marks using combination of Hough Transform and Kalman Filter is presented in Section 3. The detailed explanations about procedures used for finding players including players detecting, memberships identifying, players' positions estimating and players tracking are given in Section 4. Finally, the experimental results and conclusions are summarized in Sections 5 and 6, respectively.

2. Grass field extraction

2.1. Background

Since both player detection algorithm and field line marks detection algorithm rely on accurate field detection, we have to use a fully automatic and robust grass field extraction algorithm. In most previous work, it is assumed that the soccer field has one distinct dominant colour of green and it compasses a large area of an image sequence, especially in long and medium shots, which comprises the major parts of the whole video. For example, hue values in HSV (Hue-Saturation-Value) space for the grass field are set from 54 to 90 in [3], from 60 to 150 in [9] and from 65 to 85 in [2] for their soccer images. Furthermore, Ekin et al. [5] proposed a dominant field colour detection algorithm by the mean value of each colour component, which are computed around their respective histogram peaks in HSV space.

Some researches have used RGB (Red–Green–Blue) space [10,11]. Yoon et al. [11] suggested the following rules to extract the field.

$$O(x,y)=\begin{cases}1, & \text{if}\begin{cases}I_G(x,y)>I_R(x,y),\\ I_G(x,y)>I_B(x,y),\\ \left|I_R(x,y)-R_{\text{peak}}\right|<R_t,\\ \left|I_G(x,y)-G_{\text{peak}}\right|<G_t,\\ \left|I_B(x,y)-B_{\text{peak}}\right|<B_t,\\ GL(x,y)<GL_t\end{cases}\\ 0, & \text{otherwise}\end{cases} \tag{1}$$

where $O(x, y)$ is the binary output image, I_R, I_G and I_B indicate the R, G and B values at each pixel with their peak values represented by R_{peak}, G_{peak} and B_{peak}, andR_t, G_t and B_t are the threshold values for R, G and B, respectively. The $GL(x, y)$ is the gray level image and GL_t refers to the threshold value for $GL(x, y)$.

The above conditions consider dominant green colour (1st and 2nd rules), ground colour (3rd to 5th rules) and discrimination between a line and the ground (6th rule). The authors pre-set threshold values of R_t, G_t and B_t to 10, 15 and 10, respectively, and then control them according to the deviation variance from peak values. They also set threshold value of GL_t to 150 for all soccer video types. They did not introduce any complete and exact algorithm to control the threshold values.

2.2. Our field extraction algorithm

In our experiments with various image sequences, we realized that the proper threshold values chosen by Yoon's algorithm vary from one image sequence to another, depending on weather conditions, lighting and different soccer field colours. We improved the Yoon's algorithm in such a manner that it extracts the grass field precisely and automatically for each image sequence, no matter what the conditions are.

Practically verified, as shown in Fig. 1, the histogram distribution of the grass-colour for each colour component is not symmetric with respect to the peak values; hence, the selection of peak value is not an appropriate criterion to be used in finding the desired interval in Eq. (1). Therefore, similar to Ekin's system [3], we select the colour mean value in the vicinity of the peak value in each colour component ($A = \{R, G, B\}$) defined by

$$A'_{\text{peak}}=\frac{\sum_{H(i)\geqslant\alpha\cdot H(A_{\text{peak}})} i\cdot H(i)}{\sum_{H(i)\geqslant\alpha\cdot H(A_{\text{peak}})} H(i)} \tag{2}$$

where A_{peak} refers to the peak value, $H(i)$ is the value of the ith index of colour histogram and constant coefficient, α, indicates which indices must be selected with respect to peak count. We proposed the following equations for selecting the threshold values:

$$A_t=\text{std}(I_A(x,y))\ \text{for}\ (x,y)\in H_A(I_A(x,y))\geqslant\alpha\cdot A_{\text{peak}} \tag{3}$$

$$GL_t=GL_{\text{peak}}+\beta.\text{std}(GL(x,y)) \tag{4}$$

where β is a pre-defined constant coefficient and std$(\cdot)$ indicates standard deviation from A'_{peak}. We assume α and β, 0.1 and 0.75, respectively. In Fig. 1, comparison between our proposed algorithm and Yoon's algorithm is illustrated. Histograms of different colour components along with the gray level histogram of the input frame are shown in this figure.

Fig. 2 illustrates the detected grass field area for the three sample images selected from different types of soccer videos. It should be noted that the players and line marks on the soccer field are pre-

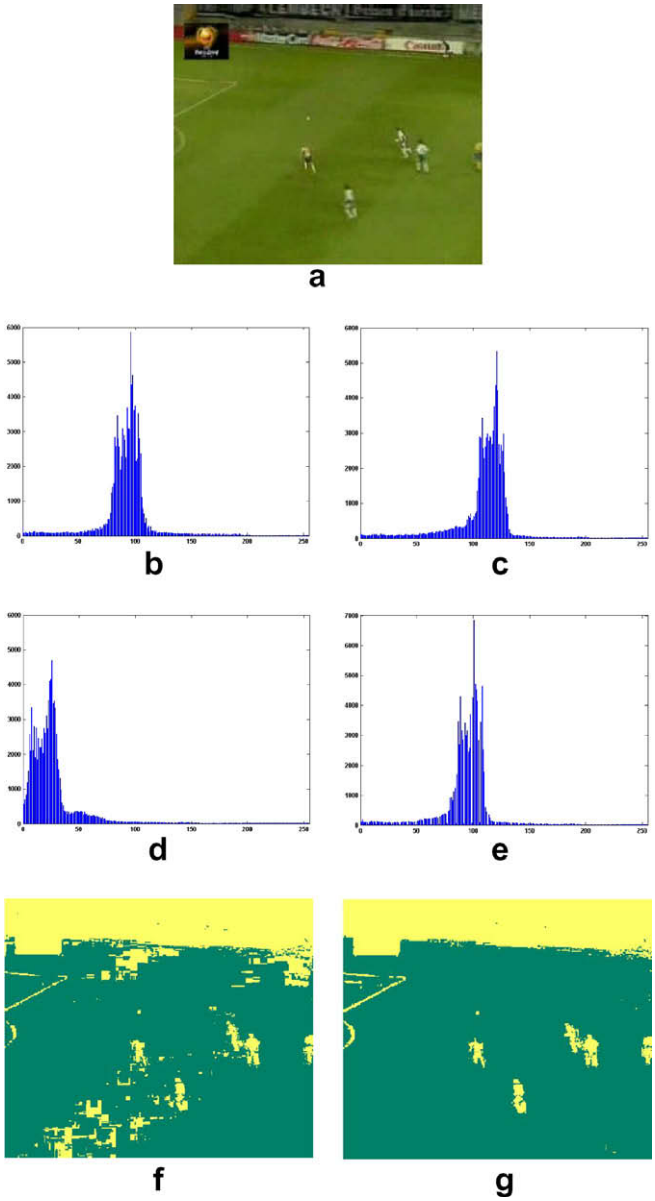

Fig. 1. Comparison of grass field extraction between our algorithm and Yoon's algorithm, (a) original frame, (b–e) histograms of red, green and blue components and gray level histogram of the input frame, respectively, (f) Yoon's algorithm, (g) our algorithm.

served, no matter what kind of field is considered. Our experimental results show that it's not necessary to change α and β for various soccer video types and conditions.

3. Field line marks detection

As mentioned before, our players tracking algorithm is based on field line marks detection, so at this step it is important to select candidate line pixels, accurately. In most cases, line marks are white in colour, but sometimes there are other white objects in the images, which might confuse the system. Other objects such as TV logos, advertisement billboards, the spectators, or even the players themselves can have white coloured parts. Moreover, the parts of line marks may not have pure white colour due to wear, tear and low resolution.

In order to detect line pixels of the field, we first apply a two-dimensional filter, h, defined by Eq. (5) on the gray level image,

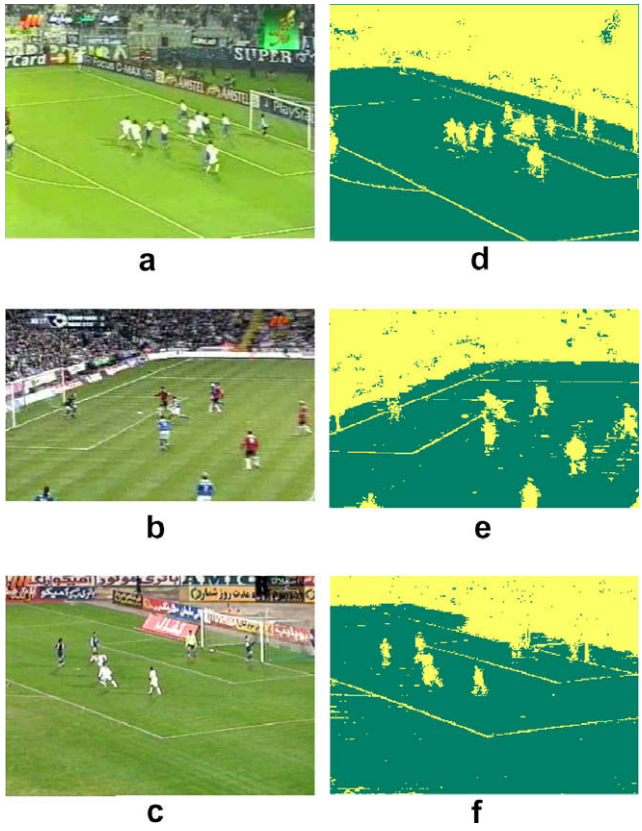

Fig. 2. Grass field detection results, (a–c) three sample images from different types of soccer videos, (d–f) corresponding grass field detection using our algorithm.

which is obtained from the colour image. To remove regions belonging to other objects, those pixels containing more than 75% of non-grass pixels in a 10 by 10 neighbourhood are removed.

$$h = \begin{pmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{pmatrix}. \tag{5}$$

Furthermore, to eliminate the pixels with low RGB values or low gradients, the candidate pixels must satisfy both conditions, given below:

$$\begin{cases} I_A(x,y) > A_{\text{peak}} + \frac{A_t}{2} \text{ for all } A \in \{R,G,B\}, \\ GL(x,y) > GL_t \end{cases} \tag{6}$$

This gives rise to more accurate results. Then, we apply thinning morphology in order to eliminate extra pixels. The candidate line pixels detected in a sample image are shown in Fig. 3. Finally Hough Transform is applied on the selected candidate line pixels to detect the line marks.

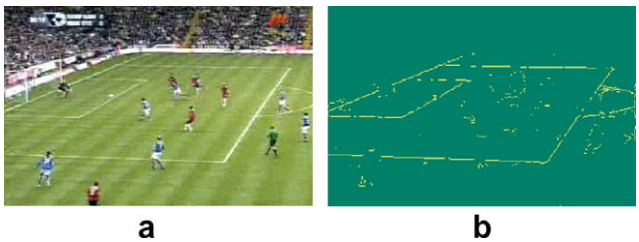

Fig. 3. (a) Original image, (b) candidate line pixels.

In order to reduce the line detection error, we track the detected lines in the subsequent frames. Since detection of line marks in each frame by Hough Transform is computationally expensive, we use Extended Kalman Filter (EKF) in a Hough space, developed by Mills et al. [22], to track those lines that are detected during the first frame. By this algorithm, searching for Hough parameters is restricted to a small part of accumulator array; hence, the efficiency of the Hough Transform computations is improved.

In Mill's algorithm, the positions of a set of n lines in each frame are estimated by Kalman Filter. These lines are assumed to be moving with constant translational and rotational velocity in the image plane. The state vector consists of the coordinates of each line (ρ_i, θ_i for $1 \leqslant i \leqslant n$) and their motion parameters (centre of rotation, (x, y), the rotational velocity, ω, and the translational velocity (u, v)) defined by

$$s_{t+1} = \begin{bmatrix} x_{t+1} \\ y_{t+1} \\ u_{t+1} \\ v_{t+1} \\ \omega_{t+1} \\ \rho_{1,t+1} \\ \theta_{1,t+1} \\ \vdots \\ \rho_{n,t+1} \\ \theta_{n,t+1} \end{bmatrix} = \begin{bmatrix} x_t + u_t \\ y_t + v_t \\ u_t \\ v_t \\ \omega_t \\ r_{1,t} \\ \theta_{1,t} + \omega_t \\ \vdots \\ r_{n,t} \\ \theta_{n,t} + \omega_t \end{bmatrix} = f(s_t) \tag{7}$$

where subscript t refers to frame number and $r_{i,\,t}$ is given by

$$r_{i,t} = \rho_{i,t} - x_t \cos(\theta_{i,t}) - y_t \sin(\theta_{i,t}) + (x_t + u_t)\cos(\theta_{i,t} + \omega_t) + (y_t + v_t)\sin(\theta_{i,t} + \omega_t) \tag{8}$$

with a measurement by a Hough Transform that is

$$m_t = [\rho_{1,t}, \theta_{1,t}, \ldots, \rho_{n,t}, \theta_{n,t}]^T \tag{9}$$

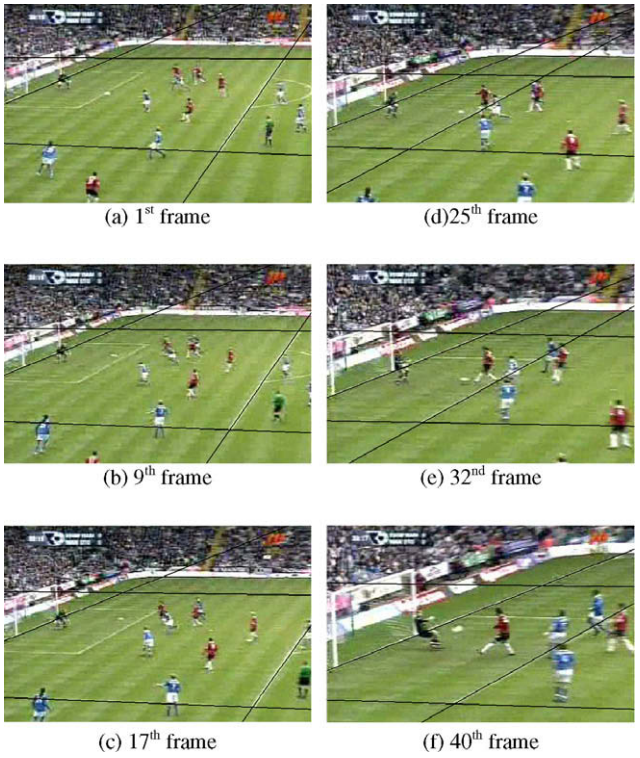

Fig. 4. The successive results of applying the line tracking algorithm on a sample image sequence with fast operation of camera.

Two error vectors associated with the EKF are given by

$$w_t = [\varepsilon_u, \varepsilon_v, \varepsilon_\omega, \varepsilon_{\rho_1}, \varepsilon_{\theta_1}, \ldots, \varepsilon_{\rho_n}, \varepsilon_{\theta_n}]^T \tag{10}$$

$$v_t = [e_{\rho_1}, e_{\theta_1}, \ldots, e_{\rho_n}, e_{\theta_n}]^T \tag{11}$$

Parameters ε_u, ε_v and ε_ω are related to errors caused by supposing u, v and ω as constant values. Parameters ε_{ρ_i} and ε_{θ_i} are errors of independent deviation from the motion model for the ith line. The measurements of ρ and θ for the ith line in Eq. (9) introduce errors e_{ρ_i} and e_{θ_1} due to quantization of the Hough accumulator. The successive results of applying the explained line tracking algorithm on a sample image sequence with fast operation of camera can be seen in Fig. 4.

4. Players detection and tracking

In the first frame, on the non-field pixels of the image, the players are detected using a region-based detection algorithm. The team membership of each player is identified using Ratio Histograms and their spatial content information. For other frames, the initial position of each player is estimated by camera motions effect elimination, and then players are tracked by searching the region of players around the estimated positions. Tracking algorithm must be robust to changes in view, player's posture, occlusion and fading. The flowchart shown in Fig. 5, displays the complete procedure for our proposed system. Details of our algorithm will be discussed in the following section.

4.1. Players detection

Detection of the players is performed based on the region-based detection algorithm by applying a 3×3 Median Filter and Connected Component Labelling algorithm on non-grass region (grass region is extracted in Section 2). This eliminates thin lines and noises and also changes the shape of players' pixels into blobs. However, when two players are located within the vicinity of each other, the Median Filter causes the players to be connected to each other through a narrow gulf. Hence, we used Opening Morphology [23] using the structuring element illustrated in Fig. 6, in order to separate objects that are joined to each other by a narrow gulf. The Opening Operation is an important step for removing these connected pixels, as shown in Fig. 7. Moreover, in order to achieve more accurate results, other restrictions such as area (at least 50) and ratio of major length to minor length (up to 6) are applied. Also, regions allocated to TV logos and advertisement billboards must be removed. The TV logos are usually located in a fixed position on the image sequence, so they can be detected by low temporal illumination variance. For detection of advertisement billboards, we check their positions in the field model as well.

Results of our proposed players detection algorithm for sample images of Fig. 2 are shown in Fig. 8. It should be noted that we also created a template for each extracted player's region that will be used for membership identification and players tracking. Each template is made of two matrices. One matrix is a binary matrix in which the ON values are assigned to players and OFF values are assigned to non-player objects. The other matrix includes colour values of players' pixels.

In some cases, such as fast camera operations, players could appear faded in the scene. This may cause the player's region detection to fail. Thus, in cases where the players detection fails, the grass field extraction step may be performed with new parameters so that the probability of players detection is improved by decreasing threshold valuesR_t, G_t, B_t and GL_t. Then, the region-based detection algorithm is applied. This process is repeated until the region of player is detected. Besides, if any occlusion is detected in the

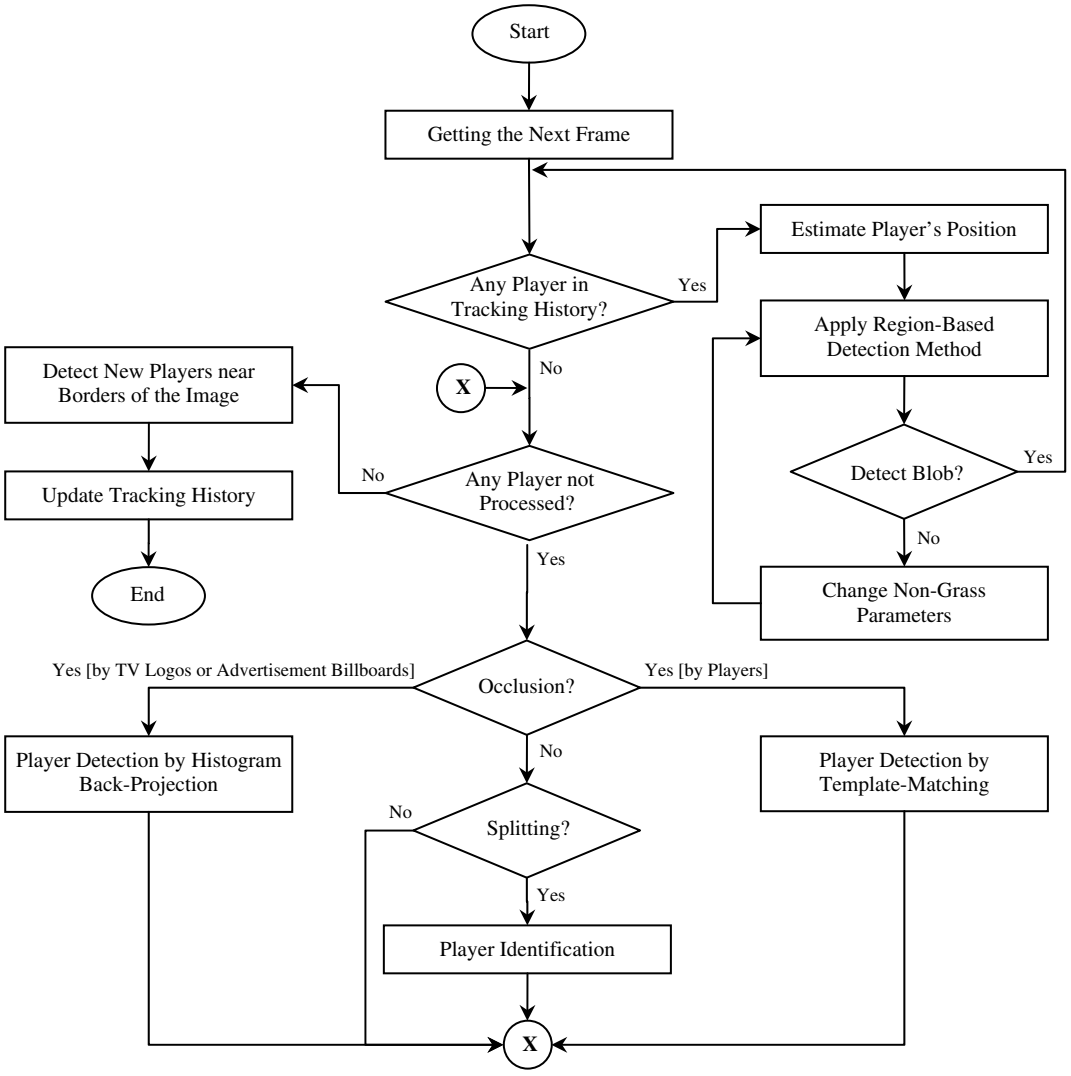

Fig. 5. The flowchart of our proposed players tracking procedure for each frame after initialization process.

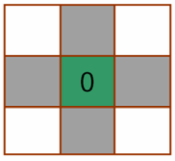

Fig. 6. Structuring element used in our opening morphology.

extracted region, we regard this region as not belonging to the player. Since in the state where occlusion or fading occurs it's difficult to extract all of the player's pixels, the player's template will be updated by its player's region only when there's neither occlusion nor fading.

4.2. Membership identification

In the grass field area of the goal scenes, there are players of competing teams, referees and a goal keeper whose colours of uniforms are different; hence, a player team membership could be determined by Ratio Histograms proposed by Swan et al. [24]. Given a pair of histograms, I (input image) and M (membership model), each containing n bins, the Ratio Histogram R_i is defined as

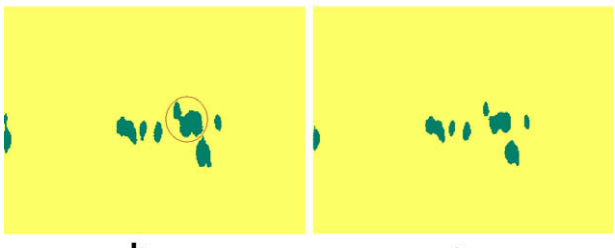

Fig. 7. Players detection result, (a) original image same as Fig. 2a, (b) applying median filtering and connected component labelling algorithm without using opening operation, (c) applying opening operation separates the overlapped players. Separation of other players is not possible at this step.

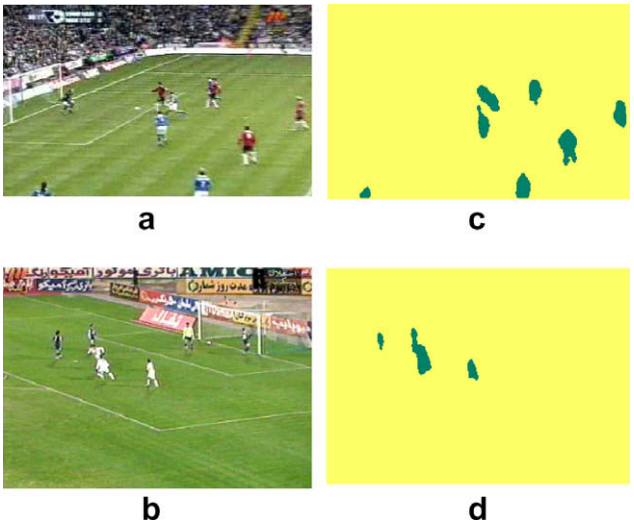

Fig. 8. Players detection results, (a–b) two sample images same as Fig. 2b and c, (c–d) corresponding players detection using region-based detection algorithm.

$$R_i = \min\left[\frac{M_i}{I_i}, 1\right] \tag{12}$$

where i refers to the index of bins in the histogram. Higher ratio value shows better fit to the model. Although in most cases, the players' memberships could be determined using the histograms alone; nevertheless, in some cases due to colour of shirts, shorts and socks and similarity between them, using the Ratio Histograms may not be a good choice. Hence, a different approach, as explained below, is used.

First, the templates of input player and each membership are partitioned into m blocks, and the Ratio Histograms for the corresponding blocks are determined. Then, the players' memberships are determined based on voting performed on m blocks. In our experiments, the blocks are partitioned into 5 equal levels. In this way, not only the appearance information is used but also the information of spatial content is incorporated for membership identification.

4.3. Players' positions estimation

Usually in a goal scene of a soccer video, the TV producers have to perform quick panning and zooming in order to follow players and the ball. This results in changing of view and position of objects with respect to the image. In other words, although the position of objects does not change considerably with respect to the field model, due to limitation of objects speed, their positions have large changes with respect to the image coordinate system in the state where fast camera operation is carried out. Furthermore, since players movements in goal scenes are erratic and change of trajectories are sudden, we can't always model the players' paths.

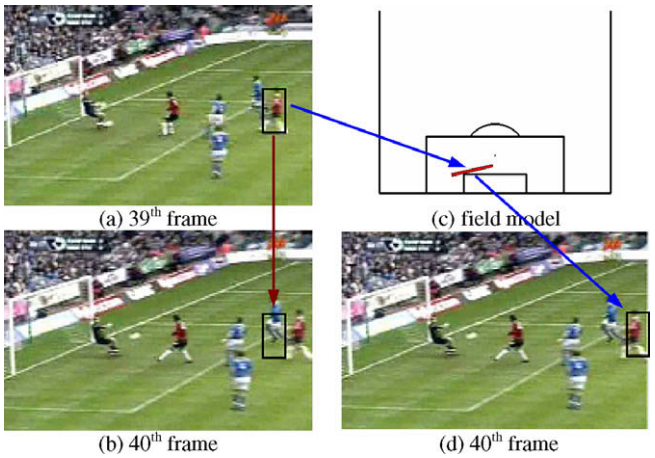

Fig. 9. Detecting player's position, (a) old frame with only one of the players highlighted, (b) corresponding position of the highlighted box shown in the current frame, (c) highlighted box projected and shown in the field model, (d) player's position following the projective mapping step.

In the literatures, estimation of players' positions in the subsequent frame is based on their positions in the preceding frames with respect to the current image. In contrast, we first locate players in the world coordinate system (i.e. the field model) for the old frame, and then estimate players' positions by computing the corresponding positions in the current frame.

In the first step, player's position is projected on the field model employing a Perspective Transformation using four control points, selected from the old frame. Then, player's position in the field model is projected back on the current frame using a similar transformation. By applying Perspective Transformation on each pixel of image, corresponding point in the field model is obtained [25].[1] The Perspective Transformation is defined as

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad or \quad x' = \mathbf{H}x \tag{13}$$

The matrix H is called a homogeneous Matrix (only ratios of terms are important). There are eight independent ratios in Perspective Transformation. It follows that projectivity has eight degrees of freedom. Hence, Perspective Transformation parameters, h_{ij}, are calculated by selecting the four corresponding points between image and field model, and then solving the following equations

$$x' = \frac{x'_1}{x'_3} = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}} = \frac{h'_{11}x + h'_{12}y + 1}{h'_{31}x + h'_{32}y + 1} \tag{14}$$

$$y' = \frac{x'_2}{x'_3} = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} = \frac{h'_{21}x + h'_{22}y + 1}{h'_{31}x + h'_{32}y + 1} \tag{15}$$

$$h'_{ij} = \frac{h_{ij}}{h_{33}} \tag{16}$$

Note that the matrix H can be multiplied by an arbitrary non-zero number without altering the projective transformation.

Since, we are only interested in scenes leading to a goal in a soccer match, for candidate control points we consider line marks in the penalty area and also the intersections of two horizontal lines and two vertical lines. In reality, in most goal scene sequences, there are at least two horizontal lines and two vertical lines available. Note that it is not required that the intersection points themselves lie inside the image area, since their positions can be calculated by the line parameters.

In Fig. 9, a box is drawn to indicate player's position in the old frame (Fig. 9a) and its current position following a projective mapping (Fig. 9d) and without mapping (Fig. 9b). As shown in Fig. 9d, the bounded box that is located on the player is relatively exact and only few mismatches are noticeable due to the player movement. This illustrates that even when a player moves fast towards the goalpost, the player's position change is very small in the field model. Therefore, by mapping the player's position to the field model, player's position estimation will be robust.

Considering the above notes, in order to map the pixels belonging to the image plane into the standard soccer field model and

[1] **Projectivity Definition:** A *projectivity* is an invertible mapping h from P^2 to itself such that three points x_1, x_2 and x_3 lie on the same line if and only if $h(x_1)$, $h(x_2)$ and $h(x_3)$ do [23]. **Projectivity Theorem:** A mapping $h{:}P^2 \to P^2$ is a projectivity if and only if there exists a non-singular 3×3 matrix **H** such that for any point in P^2, represented by a vector x, $h(x)$=**H**x [23].

vice versa, it is required to find two horizontal lines and two vertical lines in the image plane and their corresponding lines in the model. It should be noted that in a soccer field, there are three horizontal lines and six vertical lines, which could be observed near the goalpost. Therefore, the combinatorial number of possible ways that may be considered under the mapping is equal to 45 ($C(3,2) \times C(6,2)$), which $C(n,r)$ is defined as

$$C(n,r) = \frac{n!}{(n-r)!r!} \tag{17}$$

For each combination, the Perspective Transformation parameters are obtained and they are back projected into the candidate line pixels of the image (refer to Section 3 for extracting candidate line pixels). For those line pixels that are back projected on the image (the line pixels that are outside the image borders are not considered) a reward (positive score) is assigned if any candidate line pixel exists. Otherwise, punishment (negative score) is assigned. The combination with the highest score is selected as the final result. Also, in order to consider error in our mapping, the candidate line pixels are dilated by the structure element shown in Fig. 6, before any Back-Projection.

It should be noted that when the distance between the detected lines is further, the actual estimated players' positions will be more accurate. For this reason, whenever possible, those detected lines that are farthest apart from each other are used for tracking, i.e. lines such as those highlighted in Fig. 4a. During the tracking step, when any lines are disappeared from the tracking list, the missing lines are replaced with a new set of lines, which are detected in the new frame, e.g. the new line shown in Fig. 4d.

4.4. Players tracking

Player's position in the world coordinate system is estimated using a soccer field model for the current frame with respect to the player's position in the preceding frame. This is the main approach for our tracking algorithm. This causes the camera motions effect to be eliminated, while a player around its position is searched. In the first step, the position of each player is detected in the current frame (refer to Section 4.3). In the second step, a region-based detection algorithm is performed to extract players' regions. Since the position of each player changes with respect to the field, due to the player movement and/or projection error, we apply the region-based detection algorithm in an extended area of estimated player's region.

In tracking step, the major problem is occlusion. Gabriel et al. [26] divided the Occlusion Reasoning algorithms into two major classes, the Merge-Split approach and Straight-Through approach. In Merge-Split approach, when several blobs are predicated as being occluded, they are merged and encapsulated into a new single blob. In this case, the new blob is an active blob and is tracked using its new feature characteristic. On the other hand, upon separation of a blob from encapsulation, the new generated blob is detected and identified using the features before encapsulation. In this approach, the Occlusion Reasoning consists of three parts: (1) occlusion predication, (2) splitting predication and (3) blob identification. In Straight-Through approach, the tracking is based on individual blobs, even in the presence of occlusion. In this case, the blobs are not merged, but they are continuously tracked individually; therefore, it is necessary that the pixels belonging to an object are classified.

In the image sequences of the soccer games we used in our experiments, the player images have low resolutions and are deformable. Employing Straight-Through approach requires the features to be robust and be able to be extracted in all cases. However, in some cases (especially for the beginning frames of occlusion) it is possible to use the most updated template of the players and employing the Template-Matching algorithm to discriminate occluded objects. Thus, for small occlusion scenario, the Template-Matching algorithm is used. In other cases, the Merge-Split approach is used. In this regard, by referring to the tracking list of players, the following steps are repeated until the list is reduced to one player within a blob.

First, for those players who are either separated or detected by Template-Matching algorithm in the preceding frames, their templates are convolved with the input image, on the search region, and the region with highest similarity index is found. The similarity index used here is based on Euclidean distance between RGB pixels of the input image and RGB pixels of the template. It should be noted that only active pixels on the template (ON values in corresponding binary matrix) are used for similarity index computation. If the computed similarity index is high enough then the detected region is assigned to the player, and player's pixels are eliminated from non-grass region for further processing. Following the Template-Matching algorithm step and elimination of some non-grass region, the largest region is assigned to other occluded players. If the remaining region is not large enough, we label the players as hidden.

In the tracking process, if there are more blobs on the search regions, it is necessary that the separated players are identified. Assume that object A is identified as a blob before occlusion. Following any separation from an occlusion, there exist two cases for identification of object A:

(1) The membership of object A is easily differentiable from other encapsulated objects.
(2) There are other objects on the encapsulated objects similar to object A.

The problem, mentioned for the first case, is easily resolved using the information of colour (refer to Section 4.2). In the second case, the information of appearance for object A before occlusion is not enough, because there are other objects with the same membership as object A in the encapsulation. In this case, assuming constant speed and direction for objects in occlusion with respect to the field model, the estimated nearest distance of objects is used.

The aforementioned operations are executed when players are occluded by each other. In the cases where the occlusions occur between players and the TV logos or advertisement billboards, we use the Histogram Back-Projection algorithm [24]. In this algorithm, the Ratio Histogram (refer to Section 4.2) of a player's template is back projected into the searched area and the region with the most similarity index is assigned to player's region. As mentioned earlier, in case of occlusion, the player's template is not updated. Also, upon arrival of any new players into the scene, the search is performed in the border area of the image. When a new player is inserted to the tracking list, player's region is assigned to its template.

5. Experimental results

The proposed algorithm has been tested on real image sequences of goal scenes that were totally eight minutes long and had been captured by different sources, i.e. VHS tapes, TV cards and commercial compact-disks. The image sequences were obtained from different games, played under sunlight and spotlight conditions. The database was composed of 126 clips of different resolutions at 25 or 30 frames per second, which were transformed to DIVXMPG4 format. This algorithm was implemented in MATLAB on Windows XP using a PC ATHLON XP 2000 with 1.67 GHz processor. The computation period for tracking players depended on many factors, such as the number of players, occlusion, image size,

noise, etc. In average, the time required to process each frame was about 1.5 s.

5.1. Players detection

Our players detection algorithm is evaluated using different image sequences and our results are compared with other algorithms. We used our modified threshold value as defined in Eq. (2) and compared it to the algorithms proposed by Yoon and Ekin. Without loss of generality, due to the large amount of data, we only used one fourth of all frames in the 126 clips. All three algorithms are evaluated under the following conditions:

- When two or more players are occluded, they are considered as one player
- When there is any player near the border lines of a frame where it's not detected easily, we do not consider this player in our counting
- Players that are occluded by the TV Logos or advertisement billboards are not counted in our experiments

This decision could be made to be independent of user, when these conditions are applied to all three algorithms. In any case, under small occlusion, if any of the three algorithms identifies a player as a separate player, whereas the other two algorithms would not do so, then an error is counted for these two algorithms. As an example, for the player shown in Fig. 7, which has been separated from other occluded players, as a result of applying Opening operator, if any of the three algorithms detects it as one separate player then it should be detected by other algorithms as well; otherwise, an error is assigned to these algorithms. Also, for the players who are in the vicinity of image borders, TV logos or advertisement billboards, if at least one of the three algorithms is able to detect them, then counting is correct. On the other hand, if all algorithms fail to detect this player, we ignore it and do not consider it in our counting. For example, the player on the left side of Fig. 13b is not considered in our players counting because it has not been detected by any of the algorithms.

The performance of all three mentioned algorithms, under similar conditions, are compared and shown in Table 1. The following quantitative criteria, i.e. True Detection (TD), False Positive Detection (FPD), False Negative Detection (FND), Positive Predictivity and Sensitivity are used for comparison. The Positive Predictivity and Sensitivity are defined by

$$+\text{Predictivity} = \frac{\text{TD}}{\text{TD} + \text{FPD}} \tag{18}$$

$$\text{Sensitivity} = \frac{\text{TD}}{\text{TD} + \text{FND}} \tag{19}$$

5.2. Line detection and tracking

In all of our experiments, line detection for the first frame is performed either automatically or manually. Among 126 video clips used in our experiments, there were 97 clips in which we experienced no difficulty finding the four required lines automatically for the first frame. Those clips, in which the line detection did not work well for the first frame, were selected manually. In all cases, the remaining lines were detected and tracked automatically.

Among 13,340 video frames, there were 831 frames (about 6.23%) in which our algorithm did not perform correctly during the line tracking step. In Fig. 10, the two frames, in which our system did not find the proper lines, are shown. In most of these cases, the lines were not detected because they were hard to be detected, even by a human observer. Also, in most of the cases, the missing lines were remained undetected for several consecutive frames.

Fig. 10. Some of the samples in which the proper lines could not be detected by our system. The lines are blurring due to fast camera movement and all of them are difficult to be detected (In these samples, some horizontal lines are not detected by our algorithm).

Table 1
Players detection comparison using 15411 players of the 126 clips

Criterion	Our algorithm	Yoon's algorithm	Ekin's algorithm
TD	13 403	11 532	12 495
FPD	1524	4681	1833
FND	2008	3879	2916
+Predictivity	89.79	75.32	87.20
Sensitivity	86.97	74.82	81.07

5.3. Membership identification

The experiments for players' membership identification are performed on the results of players detection step (refer to Section 5.1) where the detected region includes only one player (among 13,403 players who are detected correctly, there are 9384 regions that are occupied by one player). During the system learning step, for each member, at most 10 samples from the beginning frames of the clips are manually selected. Our experiments show that among the 9384 players detected, there were 68 cases whose memberships were determined incorrectly. Among these 68 cases, there are 36 cases in which the players are detected near the borders of the images and the players regions are not completely observable.

5.4. Players tracking

In a similar manner, using one forth of all frames, the errors in our players tracking were evaluated. At the players tracking step, we say an error has occurred when either of the following conditions is met:

- Tracking list points to a position in which there is actually no player present (FPD).
- Tracking list missed to point a player (FND).

Table 2 list the results obtained from our tracking algorithm.

Table 2
Players tracking result using the 126 clips

Criterion	Our algorithm
TD	18 124
FPD	1909
FND	771
+Predictivity	90.47
Sensitivity	95.92

Furthermore, to evaluate the proposed Occlusion Reasoning algorithm, the following items have been mentioned:

- Capability of the algorithm in detecting a player's position when occluded by TV logos or advertisement billboards: for this purpose, after the system detects a blob as occluded by TV logos or advertisement billboards, we consider how long the algorithm is capable to detect the blob position correctly. The system has detected the beginning of occlusion between a blob and TV logos or advertisement billboards, 261 times among all of the frames. There were 2698 cases in which at least 50% of intersection existed between the real minimum bounding rectangle and the minimum bounding rectangle detected by Histogram Back-Projection method.
- Capability of the algorithm in detecting players using the Template Matching algorithm in the beginning of occlusion with other players: Among all of the frames, the system detected the beginning of occlusion between two blobs, 409 times. The number of cases in which the system was able to detect the player using Template Matching algorithm was 3003.
- Capability of the algorithm in identifying players in split step: The system needed to identify players after the Merge step 604 times, among which 85 separated blobs belonged to more than one player. Among the remaining players, 324 players were identified using the Histogram Back-Projection method and 195 players were identified by speed and direction information. Only 6 players in the first case and 36 players in the second case were identified incorrectly.

5.5. Test cases

In Fig. 4, several frames were shown to illustrate the performance of our line tracking algorithm. The same frames are shown in Fig. 11 and illustrate the performance of our players tracking algorithm. In the first frame, players marked 1 and 3 are occluded by players 2 and 4, respectively. Hence, the system detected the overlapping players as one player in region-based detection algorithm. In the first frame, players' memberships are manually determined. The occluded players are also identified. In Fig. 11, the occluded players are identified with white boxes. At this step, the template for each team membership is constructed and updated for each consequent frame.

In the 25th frame, the blob of player 5 is merged with the blobs of players 3 and 4 that have been detected as occluded players before. Therefore, at this step, player 5 is searched, using Template-Matching algorithm, for a region with the most similarity index. The detected region is similar enough to the player's template, thus this region is assigned to player 5 and pixels assigned to this player are deleted from the non-grass regions. The blob formed from the remaining non-grass pixels is assigned to players 3 and 4. The same situation is repeated for the 32nd frame and player 5 is detected by Template-Matching algorithm. Furthermore, in this frame, players 3 and 4 are detected separately. Therefore, the player's membership identification step is performed and their memberships are assigned. Similar situation for players 1 and 2 in frame 40 has happened and membership is determined for players 1 and 2 after separating.

As seen in Fig. 11f, players 3 and 5 are merged because the region with the most similarity index for the template of player 5 is not similar enough, and therefore the tracking is continued using Merge-Split approach. Also, since these two players are merged, their positions in the field model are determined by player 5 which causes the position of player 3 to be determined incorrectly. Furthermore, there is an occlusion between the goalie (indicated as player 6) and the advertisement billboards from frame 27. Therefore, Histogram Back-Projection is used for detecting the goalie. The result of applying our tracking algorithm on this image sequence, for obtaining player 4 trajectory in the field model, is shown in Fig. 12.

In another experiment, a different image sequence was used and a few samples of this sequence are shown in Fig. 13. In this sequence, the system is not able to detect and track the goalie, because the image is blurred when the goalie appears in the image as shown in Fig. 13b. In the first frame, players 1 and 2 overlapped each other; hence, the system detected the overlapping players as one player in region-based detection algorithm. These players are tracked as one region until separation has occurred. At this step, the players are identified by Histogram Ratio. The tracking results of players 1 and 2 and their trajectories in the field model are shown in Fig. 14. On the other hand, as shown in Fig. 13f, players 3 and 4 were detected as new players in the scene even though they were absent in the previous frames.

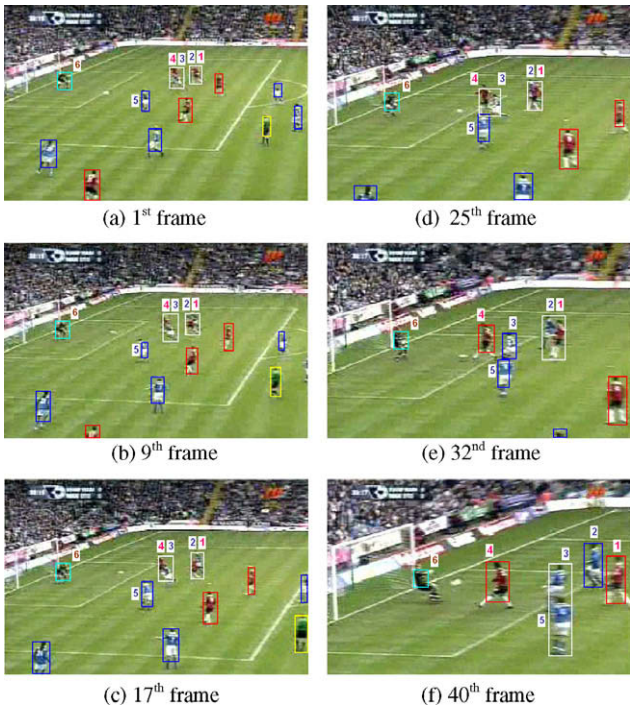

Fig. 11. Results of applying our players tracking algorithm on the sample image sequence shown in Fig. 4. The red and blue boxes are assigned to each competing team, the turquoise colour is assigned to goal keeper, yellow box to referee and the white colour is assigned to cases in which the players are encapsulated. Note that the displayed numbers are manually added so that they can be indexed in the text.

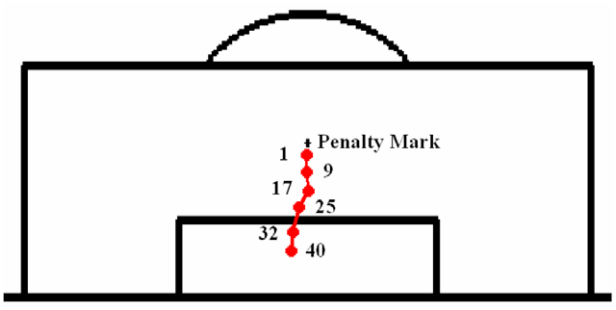

Fig. 12. Obtained trajectory of player 4 in the field model for the image sequence shown in Fig. 11.

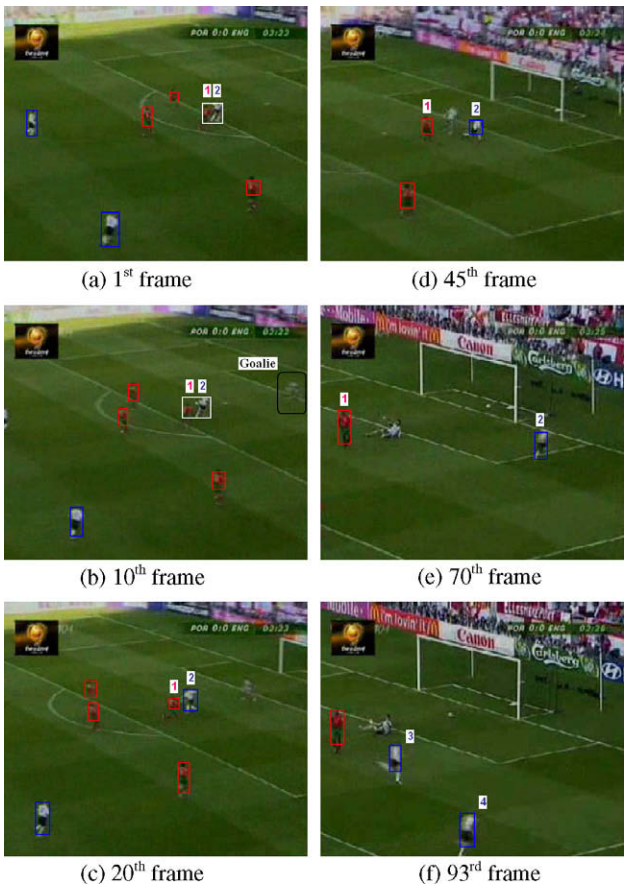

Fig. 13. Results of applying our players tracking algorithm on a sample image sequence. The red and blue boxes are assigned to each competing team, the turquoise colour is assigned to goal keeper, yellow box to referee and the white colour is assigned to cases in which the players are encapsulated. Note that the displayed numbers are manually added so that they can be indexed in the text.

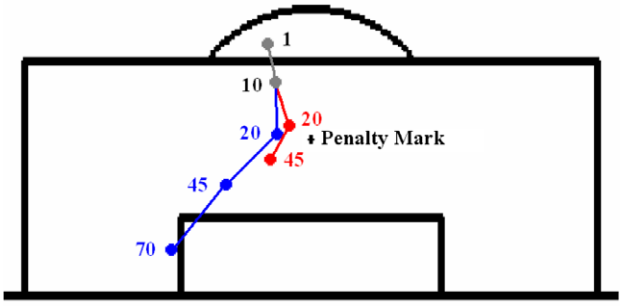

Fig. 14. Obtained trajectory of players 1 and 2 in the field model for the image sequence shown in Fig. 13. (Red and blue colours represent players 1 and 2, respectively, and the gray colour is common between both players.)

6. Conclusion

The main objective of this paper is to track soccer players in goal scenes within a video footage. Goal scenes are important segments of a soccer game for coaches, players and spectators. The assumed constraint in this paper (detection of four lines in the penalty area) is satisfied most of the time. In this paper, players' positions estimation algorithm is based on finding the four white lines in the penalty area. In goal scenes, usually all or parts of these four lines are visible. For close-up shots, one can use other algorithms like using the information of player's position in the previous frames and using Kalman Filter for estimation.

Players are tracked by searching in the output of region-based detection algorithm. Furthermore, for occlusion reasoning between players, either Template-Matching algorithm or Merge-Split approach is used. The Template-Matching algorithm is used when the detected area is very similar to the player's template. An important step in using the Merge-Split approach is identifying players following the separation from the occlusion. For this purpose, when a player does not have the same membership as occluded players, the information such as colour, spatial or other information like player velocity and direction is used.

Experimental results show the capability and robustness of our algorithm for tracking players even in the presence of occlusion, fading and fast camera operation. Despite the high performance of our proposed algorithm, there is a deficiency that is worth being noted. The deficiency is the miss-detection of players' positions in the field model when Merge-Split approach is used. In other words, when occlusion is encountered, the players are merged together and their positions are considered the same for all occluded players and thus an error is highlighted.

Our proposed approach can be applied to other sports video in which a pre-defined field model is introduced. Future work will concentrate on developing an algorithm that can deal with complex cases, image mosaicing of a goal scene and automatic goal scenes detection.

Acknowledgement

The authors thank the reviewers, who provided helpful and constructive comments to improve this manuscript.

References

[1] Y. Gong, T.S. Lim, H.C. Chuan, H.J. Zhang, M. Sakauchi, Automatic parsing of TV soccer programs, in: Proceedings of 2nd IEEE International Conference on Multimedia Computing and Systems, 1995, pp. 167–174.
[2] A. Ekin, A.M. Tekalp, A framework for tracking and analysis of soccer video, in: Symposium Electronics Imaging: Science and Technology: Visual Communication and Image Processing, 2002, pp. 763–774.
[3] P. Xu. L. Xie, S.F. Chang, A. Divakaran, H. Sun, Algorithm and system for segmentation and structure analysis in soccer video, in: Proceedings of International Conference on Multimedia Expo, 2001, pp. 721–724.
[4] D. Yow, B.L. Yeo, M. Yeung, B. Liu, Analysis and presentation of soccer highlights from digital video, in: Proceedings of Asian Conference on Computer Vision, 1995, pp. 167–174.
[5] A. Ekin, A.M. Takalp, R. Mehrotra, Automatic soccer video analysis and summarization, IEEE Trans. Image Process. 12 (7) (2003) 796–807.
[6] Y.Q. Yang, Y.D. Lu, W Chen, A framework for automatic detection of soccer goal event based on cinematic template, in: Proceedings of the International Conference on Machine Learning and Cybernetics, vol. 6, 2004, pp. 3759–3764.
[7] R. Leonardi, P. Migliorati, M. Prandini, Semantic indexing of soccer audio-visual sequences: a multimodal approach based on controlled Markov chains, IEEE Trans. Circ. Syst. Video Technol. 14 (5) (2004) 634–643.
[8] S.C. Chen, M.L. Shyu, M. Chen, C. Zhang, A decision tree-based multimodal data mining framework for soccer goal detection, in: Proceedings of the IEEE International Conference on Multimedia and Expo, vol. 1, 2004, pp. 265–268.
[9] O. Utsumi, K Miura, I. Ide, S. Sakai, H. Tanaka, An object detection method for describing soccer games from video, in: Proceedings of IEEE International Conference on Multimedia and Expo, vol. 1, 2002, pp. 45–48.
[10] Y. Seo, S. Choi, H. Kim, K.S. Hong, Where are the ball and players? soccer game analysis with color-based tracking and image mosaic, in: Proceedings of International Conference on Image Analysis and Processing, 1997, pp. 196–203.
[11] H.-S. Yoon, Y.L.J. Bae, Y.K. Yang, A soccer image sequence mosaicing and analysis method using line and advertisement board detection, ETRI J. 24 (6) (2002) 443–454.
[12] A. Yamada, Y. Shirai, J. Miura, Tracking players and a ball in video image sequence and estimating camera parameters for 3D interpretation of soccer games, in: Proceedings of the 16th International Conference on Pattern Recognition, vol. 1, 2002, pp. 303–306.
[13] N. Vandenbroucke, L. Macaire, J.G. Postair, Contribution of a color classification to soccer players tracking with Snakes, in: Proceedings of the International Conference on System, Man, and Cybernetics, vol. 4, 1997, pp. 3660–3665.

[14] S. Lefevre, C. Fluck, B. Maillard, N. Vincent, A fast snake-based method to track football players, in: IAPR International Workshop on Machine Vision Applications, RFAI publication, 2000, pp. 501–504.
[15] S. Iwase, H. Saito, Parallel tracking of all soccer players by integrating detected positions in multiple view images, in: Proceedings of the 17th International Conference on Pattern Recognition, vol. 4, 2004, pp. 751–754.
[16] P. Figueroa, N. Leite, R.M.L. Barros, I. Cohen, G. Medioni, Tracking soccer players using the graph representation, in: Proceedings of the 17th International Conference on Pattern Recognition, vol. 4, 2004, pp. 787–790.
[17] J. Perš, G. Vuckovic, S. Kovačič, B. Dezman, A low-cost real-time tracker of live sport events, in: Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis in Conjunction with 23rd International Conference on Information Technology Interfaces, 2001, pp. 362–365.
[18] J. Perš, S. Kovačič, Tracking people in sport: making use of partially controlled environment, in: Proceedings of 9th International Conference on Computer Analysis of Images and Patterns, 2001, pp. 374–382.
[19] J. Perš, S. Kovačič, Computer vision system for tracking players in sports games, in: Proceedings of the 1st International Workshop on Image and Signal Processing and Analysis, 2001, pp. 81–86.
[20] S.S. Intille, A.F. Bobick, Real-time closed worlds tracking, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997, pp. 697–703.
[21] S.S. Intille, A.F. Bobick, Closed worlds tracking, in: Proceedings of the 5th International Conference on Computer Vision, 1995, pp. 672–678.
[22] S. Mills, T.P. Pridmore, M. Hills, Tracking in a Hough space with the extended Kalman filter, in: The British Machine Vision Conference, Norwich, 2003, pp. 173–182.
[23] M. Sonka, V. Hlavac, R. Boyle, Image Processing, Analysis and Machine Vision, Chapman & Hall, Computing, London, UK, 1993.
[24] M.J. Swain, D.H. Ballard, Color indexing, Int. J. Comput. Vis. (1991) 11–32.
[25] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2000.
[26] P. Gabriel, J. Verly, J. Piater, A. Genon, The state of the art in multiple object tracking under occlusion in video sequences, Adv. Concepts Intell. Vis. Syst. (2003) 166–173.