ARTICLE IN PRESS

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Vision and Image Understanding xxx (2005) xxx–xxx

Computer Vision and Image Understanding

www.elsevier.com/locate/cviu

# Tracking soccer players aiming their kinematical motion analysis

Pascual J. Figueroa [a], Neucimar J. Leite [a,*], Ricardo M.L. Barros [b]

[a] Instituto de Computação, Universidade Estadual de Campinas, Avenida Albert Einstein, 1251, Caixa Postal 6176, 13084-971 Campinas, SP, Brazil
[b] Faculdade de Educação Física, Laboratório de Instrumentação para Biomecânica, Universidade Estadual de Campinas, Avenida Érico Veríssimo 701, Caixa Postal 6134, 13083-851 Campinas, SP, Brazil

Received 19 February 2004; accepted 6 July 2005

**Abstract**

In this work, we consider the problem of tracking players, during a soccer game, through the use of multiple cameras. The main goal here consists in finding the position of the players on the pitch at each instance of time. The tracking is performed through a graph representation in which the nodes correspond to the blobs obtained by image segmentation and the edges, weighted using the blobs information and trajectory in the image sequence, represent the distance between nodes. We present a new way of trating occlusions by splitting segmented blobs based on morphological operators and a backward and forward graph representation which allows an increasing in the number of frames automatically tracked. Unlike other works in which the analysis of short video sequences is presented, this paper illustrates the tracking results for all the players during a whole game.


*Keywords:* Tracking; Soccer players; Motion analysis

## 1. Introduction

Soccer is a very popular sport in the world and there is a great interest in better understanding its important fundaments if one wants to increase the performance of a team during a game, and better adapt the planning of the trainings. The movement of the players on the field, as a function of time, is a useful information that can contribute for improving the performance of the players at different positions [1]. For tactical variations that a team can assume during a game, for example, the measured values may be associated to physiological variables as well as to technical and tactical information [3,18].

The first studies concerned with the players movement during the game were made by Reilly and Thomas [18] which employed audio recorders to register the estimated location of the players. Withers et al. [23] used a camera to analyze the movement of a unique soccer player. Mayhew and Wenger [15] also used a camera to track two players, each one filmed alternately for 7 min. They computed the time spent for each activity of these tracked players, such as walking, running, jogging, staying, as well as the frequency of the corresponding activities.

Aiming to better quantify the players movements, Erdmann [7] filmed a soccer game with one stationary TV camera (using wide-angle lens of 130°) and analyzed the displacement of one player, by replaying frame by frame, using a videotape player and a transparent squared sheet adapted to the monitor of the screen. The player position was annotated each 1 s, during 5 min, and the kinematic quantities were calculated.

Henning and Briehle [10] analyze the soccer players movement by using a global positioning system (GPS). This kind of system locates the global position of the object by satellites which receive the signal emitted by a transmitter located on the earth surface. This methodology demands a device of 250 g to be carried by the tracked object from which the data are collected at a frequency of 1 Hz.

D'Ottavio and Tranquilii [6] presents a method based on a potentiometer and two cameras for tracking one player during 90 min. The operator focus on the player of interest

* Corresponding author. Fax: +55 19 3788 5847.
*E-mail address:* neucimar@ic.unicamp.br (N.J. Leite).

1077-3142/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2005.07.006

and follows his movement. The information stored in the potentiometer allows to calculate the corresponding player position as a function of time.

Recent advances in video technology and computer processing performance have motivated the interest of researchers in using computer vision and image processing techniques for the automatic analysis of the sport games by videogrammetry.

One of the first works treating the problem of automatic tracking sport players was made by Intille and Bobick [11] which developed a technique called *closed-worlds* applied in the tracking of American football players. A *closed-world* is defined as a region of time and space in which the specific context is adequate to determine all possible objects present in that region, such as players, yard-lines, hash-marks, grass and so forth. The authors report that this method works better than the common template matching for tracking isolated players and that their main difficulty is to treat the case in which a *closed-world* representation contains more than one player.

Taki et al. [21] present a method for a quantitative evaluation of a team work in soccer games. For this purpose, they considered only static cameras and isolated players which are tracked using template matching. Kim et al. [24] use the TV broadcasted images and a tracking method based on template matchings and on the histogram back-projection concept to solve the occlusion problems. Needham and Boyle [17] proposed a method named condensation which uses the Kalman filter for multi-object tracking. The method was applied for sequences of futsal video images. In [22], different systems of color representation are used in the segmentation of soccer players. Matsui et al. [14] also present a work for animation of soccer scenes using TV broadcasted images. Kang et al. [13] use color information and joint probability data association filter with a prior knowledge of the scene for a tracking approach of soccer players.

As it is the case for other tracking systems based on video cameras, the tracking methods need to solve different problems concerned with extraction, identification, and correct trajectory definition of the players. Furthermore, the outdoor unconstrained environment and the very common occlusion problems make the correct tracking of all the players more difficult.

Indeed, one of the most challenging problems related to the tracking of soccer players concerns the occlusion and the players congestions which occur, especially, in cases of free kicks and corners. Iwase and Saito in [12] uses eight cameras covering the whole region of the goal (the penalty box) to treat this problem. However, this kind of solution is expensive and the occlusion problems are not totally solved.

In this work, we present all the main procedures of a tracking system, for analyzing a whole soccer game, based on at least four static cameras which together should cover the whole playing field. As we will see elsewhere, by considering a model of the players and some morphological operations, we treat occlusion and congestion problems by splitting segmented blobs representing set of two or more partially occluded players. The splitting process is done using a graph representation in the forward and backward directions and constitutes one of the main contributions of our approach.

The overlapped regions related to the four cameras are used for synchronization and to solve some cases of occlusions. This kind of configuration, based on the placement of a set of cameras on the pitch, provides enough reference points necessary to the calibration of the cameras, and guarantees that the size of the players in the image is big enough to discriminate noise and the interested components of the scene.

The first step in our tracking algorithm concerns the segmentation of the players while a second one detects their correct trajectories. The segmentation step consists in extracting the blobs representing these players. Further, we construct a graph having the blobs as nodes and find the players trajectory by a minimal path searching on this graph. The edges of the graph are defined by considering information such as distance between blobs, colors (grey level intensity), and the movement direction of the tracked players. Here, we aim at isolating the players by splitting their corresponding blobs and correctly treating some cases of occlusions.

This papers is organized as follows: the next section introduces the segmentation algorithm. Section 3 describes the tracking procedure in details. Section 4 illustrates and discusses the results of the method when applied to real sequences of soccer images and, finally, some conclusions are drawn in Section 5.

## 2. Segmentation of the players

Background substraction is a very common method used for segmenting moving objects which consists of the difference between a set of images and its background model. To consider problems in outdoor scenes such as changes of illumination, shadows, background objects, etc., these methods need to frequently update the background representation model. For this purpose, some statistical adaptive methods [4,8,16,20] have been discussed in the literature. These methods, which work well for relatively simple scenes with slow changes of illumination, update the background model for each frame by assuming its background as a Gaussian distribution. This approach can easily incorporate to the model objects that stop moving for a certain time. In some applications, however, it is desirable to keep these still or slow moving objects correctly tracked. In soccer games, for example, it is common to have players that stand still for many video frames.

In cases of sudden changes of illumination and background objects, we need to consider more complex procedures taking into account some additional information of the analyzed sequence. For example, in [9], the background is modeled using several consecutive frames and the Gauss-

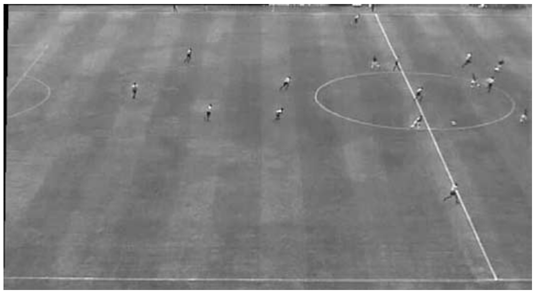

Fig. 1. A frame of a soccer video.

Fig. 2. The background image.

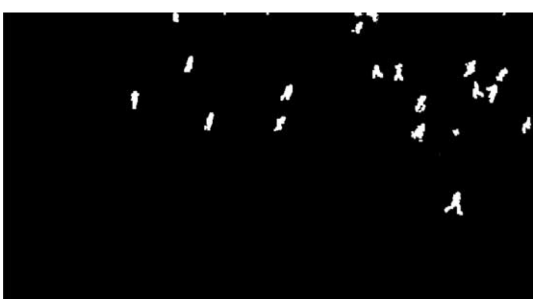

Fig. 3. The blobs in the segmented image.

ian bimodal distribution, where each pixel is represented by the minimum and maximum intensity values and the maximum difference between frames.

In this work, we extract the background image by applying a median filter along the pixels of some consecutive frames, thus, regularly updating the background image of a certain number of the video frames. The intensity of each background pixel is calculated as the mean of the most likely repeated intensities along the considered set of frames. This method correctly updates the background image for the case of slow changes of illumination and keep most of the still objects detected.

Our complete segmentation algorithm consists of the following steps:

- Background extraction.
- Difference between the current frame and the image corresponding to the extracted background.
- Image binarization by thresholding.
- Morphological filtering (opening and closing) to eliminate noise.
- Labeling of the connected pixels and definition of the corresponding regions as *blobs*.

Fig. 1 shows a frame of a soccer video sequence and Fig. 2 shows the corresponding updated background considering a set of 20 frames. Fig. 3 shows the final result of the segmentation step.

## 3. Tracking players

In this section, we consider the specific problem of tracking the blobs representing the players and, finally, the definition of their location on the playing field. As mentioned early, one of the main difficulties of the tracking process concerns the partial or total temporal occlusions of the objects. Hence, the splitting of the blobs aiming at separating or isolating the players is also considered in this section. This splitting takes into account the spatial–temporal information of the image sequence. The spatial information is explored by considering the size, shape, and color of the blobs, while the temporal information explores the relation between blobs in different frames. In this work, we use a graph representation to define this temporal dependence.

The graph is constructed from the set of blobs obtained during the segmentation step in such a way that nodes represent blobs and edges, the distance between these blobs. Therefore, each node of the graph stores the spatial information of a blob while an edges conveys the temporal information related to the dependence between blobs. This representation model allows us to better approach the correspondence problem of the objects which can help not only in the splitting of the blobs but also in correctly tracking them along the sequence. The tracking of each player is performed by a minimal path searching in the graph. Besides the distance between blobs, the edges of the graph are weighted by considering information such as velocity, orientation, and color of the blobs. This kind of structure was also used in [5] for the tracking of objects in a video stream obtained from a moving airbone platform.

Since the graph constitutes the main data structure of the proposed method, the next section discusses its construction in details.

### 3.1. The graph construction

Let $G$ be an oriented graph of a video sequence and $n_i(t)$ are the nodes at frame $t$. Our data structure can be defined according to the following steps:

1. Creation of a node $n_i(t)$ for each blob $i$ in the first frame, $t = 1$, and insertion of this node into graph $G$.
2. Creation of a node $n_j(t+1)$ for each blob $j$ in frame $t+1$ and insertion into $G$.
3. Computation of the distance $d_{i,j}$ between nodes $n_i(t)$ and $n_j(t+1)$.

4. Creation of an edge $e_{i,j}$ satisfying condition $d_{i,j} < d_{\max}$, where $d_{\max}$ represents a given maximal distance.
5. Repetition of steps 2–4 for the whole video sequence.

The distance between blobs, used for including nodes into the graph, is computed as the minimal distance between two blobs contour pixels. A natural way to compute this distance is from the centroid of the corresponding blobs, nevertheless, since a blob can represent more than one player, better results are obtained by taking into account its contour information.

Fig. 4 shows some nodes and edges of the graph for four consecutive frames.

Each node stores information about the blobs features defined during the graph construction. Examples of these information are:

- *width* and *height*: the size of the bounding rectangle of the blobs.
- *area*: the number of pixels of a blob.
- *perimeter*: the number of pixels in the contour of a blob.
- *x,y*: the coordinates of the center of a blob in the image.
- *color*: the color associated to a blob (Section 3.3).

After the graph construction, the following parameters can be defined:

- *num_comp*: the number of components relative to the number of players in a blob (Sec. 3.2).
- *dist*: the distance between two linked nodes computed as the Euclidean distance between the center points of the blobs.
- *direction*: the direction of the players trajectory.
- *velocity*: the velocity of the objects.

The main function of the edges information is to define the path or the possible paths of a player on the graph, during the tracking process. Thus, edges linking very distant blobs, according to the $d_{\max}$ value, are not included in the graph. This value may be defined from the maximal displacement a component is supposed to achieve between two consecutive frames.

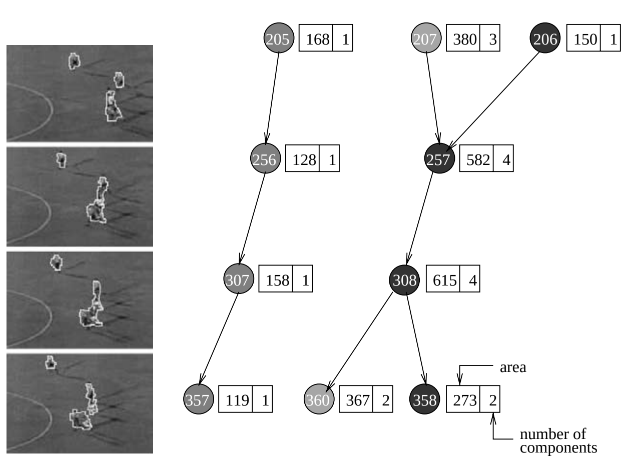


Fig. 4. Example of graph construction. The numbers inside the circles represent the labels of the blobs, the two numbers near the circles are the area and the number of components of the corresponding blobs.

### 3.2. Number of components definition

The number of components information represent the number of players in a blob. The correct determination of the number of components is important if we need to split this blob (representing more than one player), and make correct decisions about trajectories during the tracking. This number is difficult to determine when, for example, a blob containing more than one player, in frame $t$, is split or connected to another blob in frame $t + 1$.

To solve this problem, we group blobs by considering the edges between them, as shown in Fig. 5. Two nodes $v_1$ and $v_2$, in frame $t$, belong to the same group if there is any node $u$, in frame $t + 1$, so that there exist edges $(v_1,u)$ and $(v_2,u)$ in the graph. A simple way to define a group is to use a deep-first search of a new undirected graph defined by nodes and edges in frame $t$ and $t + 1$. For each new searching on the graph, starting from nodes in frame $t$, a new group number is defined. Each visited node, during this searching, receives the corresponding defined group number. For example, in Fig. 5, the searching starting from nodes $v_1$ and $v_3$ defines the groups number 1 and 2, respectively.

The area of the blobs considered here is a parameter associated with the number of components of each node. This area (in terms of number of pixels) is inversely proportional to the distance of the players with respect to the cameras location.

For each group of nodes in frame $t$, belonging to the same group, we consider a subdivision process of the objects in frame $t + 1$. First, the number of objects belonging to the same group is defined as the sum of the components number of its nodes. For each node $u$ in frame $t + 1$, belonging to a certain group, the area of the corresponding blobs in the actual position (based on the location of the players on the field) is estimated and subtracted from the actual area indicated in the blob. At this point, the total number of components of the group in frame $t$ is decreased by one, and the number of components of the nodes in frame $t + 1$ is increased in the same proportion. This step guarantees that each node is associated at least with one

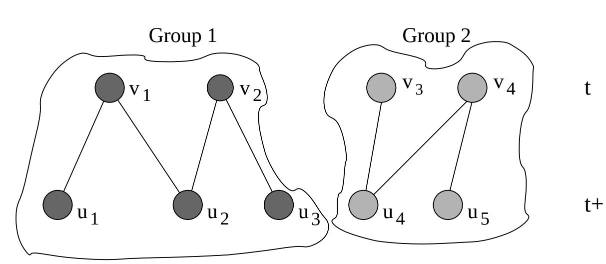


Fig. 5. Nodes grouped by common edges.

object of the scene. The above procedure is executed while the number of components in the region is greater than zero. At each step, we start always from the node having the highest area. The following algorithm summarizes the main steps of this procedure.

**Algorithm**

1. Determine the number of components $\eta$ belonging to group $R$ by adding the number of components, $\eta_v$, of each node at frame $t$.
   $\eta_R \leftarrow \sum_{v \in R}(\eta_v)$.
2. For each node $u$ belonging to group $R$ in frame $t+1$:
   Estimate the area $A_p$ of the player proportional to its blob position in the image.
   $A_u \leftarrow A_u - A_p$ {update the current area of node $u$}
   $\eta_R \leftarrow \eta_R - 1$

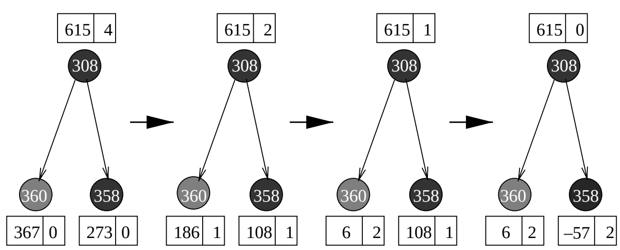


Fig. 6. Example of number of components definition.

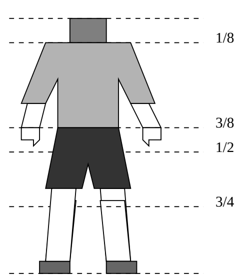


Fig. 7. A player model.

   $\eta_u = 1$ {initialize the number of components of node $u$}
3. While $\eta_R > 0$ do
   Find a node $u$ with the highest area $A_u$
   $A_u \leftarrow A_u - A_p$
   $\eta_u \leftarrow \eta_u + 1$
   $\eta_R \leftarrow \eta_R - 1$

Fig. 6 illustrates these steps for the number of components definition.

### 3.3. Blobs color definition

In soccer, as in many other group sports, the color of the uniform of the two teams must be different from each other. This information can be used to discriminate the teams as well as to solve occlusion problems. In such a case, some applications consider the histogram back-projection method [24,16] which can work well for full color uniforms. With our camera positioning setup, it is difficult to discriminate colors and, thus, we work only with the intensity or gray-level information of the components. Also, each part of the uniform may have different colors, as we can see in Fig. 8A. Generally, a player can be modeled as a group of many regions, each one having some predominant colors (Fig. 7). In [24], for example, the vertical RGB distribution of the blobs (Fig. 8B) is compared with the distribution of the corresponding players model.

In our work, since the size of a blob may change, depending on the position of the players on the pitch, we try to divide the model of the players into two or more regions, so that each region represents a part of the team uniform, i.e., t-shirt, short, socks, and so on. For each region, we consider a filtering by threshold based on the vertical intensity distribution of the blobs. The two vertical lines in Fig. 8B represent two threshold values, $T_1$ and $T_2$, defined as the minimal and maximal mean values of the vertical distribution. These thresholds constitute the limits

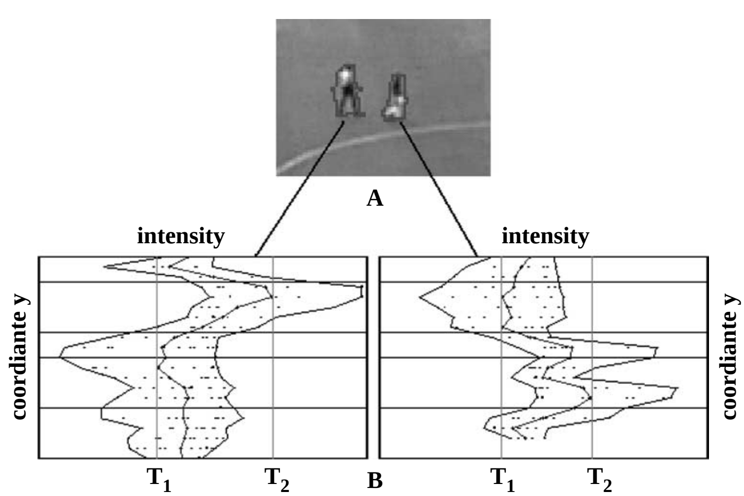


Fig. 8. Vertical intensity distribution. (A) Two players and (B) their corresponding intensity distribution.

expressing the significant intensity values of each interest region. Let, for example, $R_1$ and $R_2$ be two of these regions related to the t-shirt and short of a player, respectively. These regions can be spatially defined, based on the size of the players in the image and on the model shown in Fig. 7, as follows:

$$\frac{1}{8} * psize < R_1 < \frac{3}{8} * psize,$$

$$\frac{4}{8} * psize < R_2 < \frac{3}{4} * psize,$$

where *psize* is the estimated player size according to its position on the pitch.

For each region $R_i$, we count the number of pixels, $p$, of a blob lower than $T_1$ and greater than $T_2$, yielding the following values:

$$S1_{R_i} = \#\{p\} \quad \forall p \in R_i \wedge R_i(p) < T_1,$$

$$S2_{R_i} = \#\{p\} \quad \forall p \in R_i \wedge R_i(p) > T_2.$$

These values are associated with the most discriminant intensity values of each region, based on the considered vertical intensity distribution of the blobs. Thus, the color or intensity of each region, $R_i$, can be defined as:

$$C_{R_i} = \begin{cases} 1 & \text{if } \frac{S1_{R_i}}{S1_{R_i}+S2_{R_i}} > 0.6, \\ 2 & \text{if } \frac{S1_{R_i}}{S1_{R_i}+S2_{R_i}} < 0.4, \\ 0 & \text{otherwise.} \end{cases}$$

Finally, the color of a blob, representing the team identification, can be determined from the color information of a region or from a combination of colors of different regions. For example, for the video sequence considered here, the blobs color is defined as:

$$C_{\text{blob}} = \begin{cases} 1 & \text{if } C_{R_1} = 1 \text{ or } C_{R_2} = 2, \\ 2 & \text{if } C_{R_1} = 2 \text{ or } C_{R_2} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

### 3.4. Splitting the blobs

As we have seen in Section 3.2, we define the number of objects in each blob by considering spatial and temporal information of the segmented objects. In this section, we consider the problem of splitting the blobs containing more than one tracked object with the aim of having just one blob associated with one player. Naturally, this splitting can be difficult to implement, mainly in cases when two or more players are very close to each other and the occlusion is almost total.

During the segmentation step, some objects can be grouped into one blob due to small distances between them or effects such as shadows and noise. Our first approach to the splitting problems tries to isolate players linked by short connections, as shown in Fig. 12A. This procedure may be seen as a postprocessing of the segmentation step. Further, a blob can be split by considering the constructed graph and the model of the blobs, as we will discuss later.

To illustrate this splitting process, we selected regions of a video sequence (Fig. 9) containing some occlusions. Fig. 10 shows the blobs obtained after the segmentation step. Fig. 11 shows the corresponding graph and the number of components defined for each node.

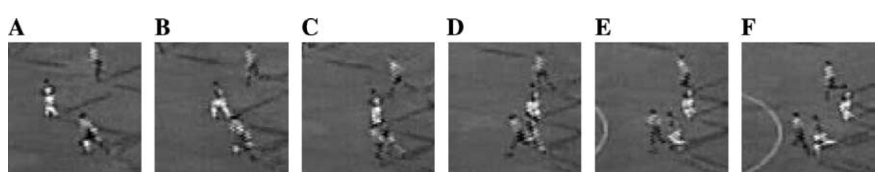


Fig. 9. Some cases of occlusions.

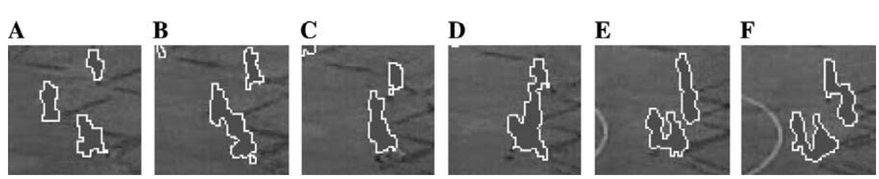


Fig. 10. The blobs related to the sequence in Fig. 9.

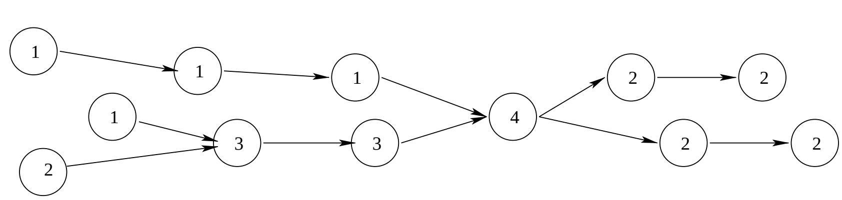

Fig. 11. The graph representation of the blobs in Fig. 10.

*3.4.1. Splitting using morphological operators*

To eliminate short connections between blobs, we consider these blobs as regions of the original images and apply a set of morphological operations [19]. First, the original image (Fig. 12A) is eroded and then dilated $n \times 3$ times ($n$ depends on the players position on the field). The result of this operation is shown in Fig. 12B (the shaded regions). The next step consists of a conditional homotopic thickening of these shaded regions, limited to the corresponding blob contours (Fig. 12C).

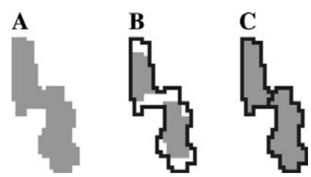


Fig. 12. Simple splitting by morphological operations. (A) original connected blobs, (B) erosion operation, and (C) conditional homotopic thickening.

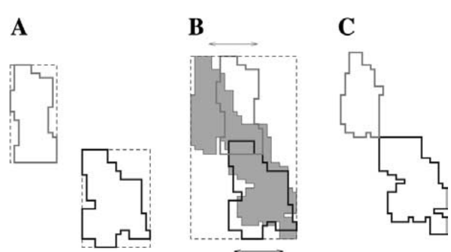


Fig. 13. Vertically blob splitting using the blobs model. (A) Blobs before connection, (B) connected blobs, and (C) blobs after a vertical splitting.


Fig. 14. Horizontally blob splitting using the blobs model. (A) Blobs before connection, (B) connected blobs, and (C) blobs after a horizontal splitting.

Fig. 18 shows some examples of the splitting by morphological operations of the components in Fig. 10.

*3.4.2. Splitting using the blobs model*

In this section, we consider the model of the blobs and the information conveyed by the graph to split a blob, containing more than one player, which cannot be split by simple morphological operations.

A blob in the current frame can be split into two or more blobs, depending on the number of components of the corresponding node in the graph and on its configuration in the previous frame. Hence, the model extracted from the blobs before an occlusion is considered in the connected version of the current blob, during a matching operation which further yields a splitting of this blob. This matching is performed by placing and shifting the model in the current blob and searching for the best fit, the region yielding more intersection pixels is selected. For example, in Fig. 13, the blobs in the current frame (Fig. 13C) were obtained from the shaded blob in Fig. 13B and from the model of the isolated blobs in the previous frame, after the above matching operation (Fig. 13A).

Depending on their location in the previous frames and on the number of components, the blobs can be split vertically or horizontally. Figs. 13 and 14 show an example of vertically and horizontally blob splitting, respectively. Since this splitting into more than three blobs becomes more complex, we consider only two or three components splitting at the same time. There are situations in which the size of a blob is not big enough to be split, horizontally or vertically. In such a case, this blob is not split and we consider that some of its corresponding objects are in total occlusion and, therefore, their location in the image is the same.

As stated before, the graph representation considered here provides useful information for the correct splitting of the blobs. This graph can be constructed in a direct order in which the normal sequence of the events is taken into account, or in a inverse order, where we start creating the graph from the last frame. These two ways of constructing the graph allow us to consider the splitting situations before and after a total occlusion. Such a possibility usually yields a reduction of the number of nodes

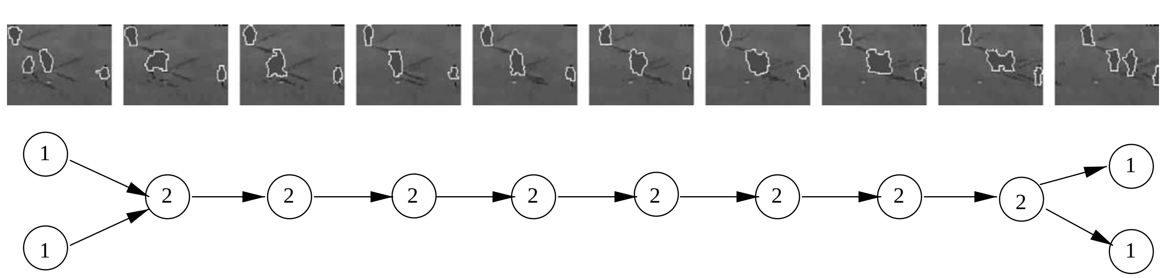


Fig. 15. Graph of a sequence containing occlusions in eight frames.

representing occlusions, as we can see, for example, in Figs. 15–17.

Fig. 19 shows the final configuration of the blobs obtained after a splitting of the segmented regions in Fig. 18.

### 3.5. Tracking each player

#### 3.5.1. Calibration of the cameras

In order to get the actual position of the players on the field, we need to calibrate the cameras, i.e, to find their intrinsic (e.g., focal length, lens distortion) and extrinsic (e.g., translation, rotation) parameters allowing the object-image transformation of the players coordinates. In this work, to estimate these parameters we apply the perspective projection matrix which relates the images coordinate to the world coordinate system [2].

The image coordinate system $o - xyz$ and the world coordinate system $O - XYZ$ can be expressed using the homogeneous coordinate as follows:

$$\begin{bmatrix} x' \\ y' \\ h \end{bmatrix} = C \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}, \tag{1}$$

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{h} \begin{bmatrix} x' \\ y' \end{bmatrix}, \tag{2}$$

where $C$ is a $3 \times 4$ matrix that performs rotation and perspective projection transformations.

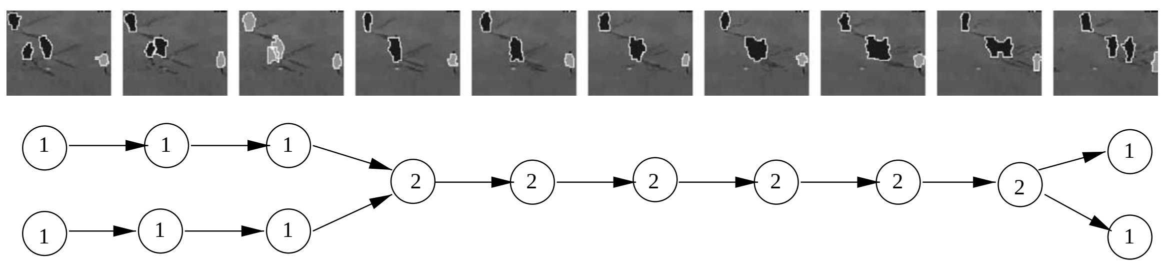

Fig. 16. Result of the blobs splitting using a graph in direct order in which the number of occlusions is reduced to six frames with respect to the number of occlusions in Fig. 15.

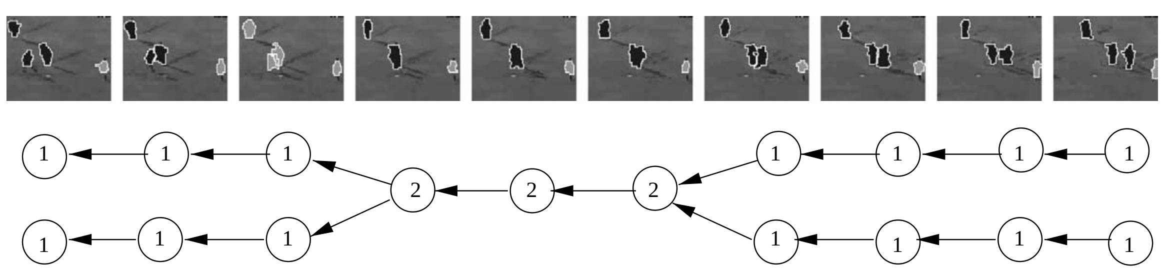

Fig. 17. Result of the blobs splitting using a graph in inverse order in which the number of occlusions is reduced to only three frames.

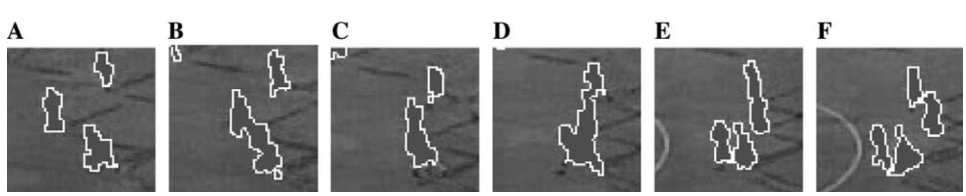


Fig. 18. Splitting by segmentation.

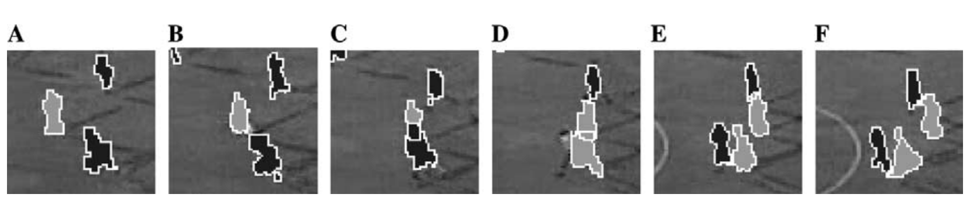


Fig. 19. Splitting by using the blobs model.

After expanding Eqs. (1) and (2), we get the following two equations:

$$XC_{11} + YC_{12} + ZC_{13} + C_{14} - XxC_{31} - YxC_{32} - ZxC_{33} - xC_{34} = 0, \tag{3}$$

$$XC_{21} + YC_{22} + ZC_{23} + C_{24} - XyC_{31} - YyC_{32} - ZyC_{33} - yC_{34} = 0. \tag{4}$$

For the 2D case, with the $Z$ coordinate in the world coordinate system equal to zero (assuming that the soccer field is flat), and $C_{34} = 1$ in the homogeneous coordinate system, we have the following equation:

$$\mathbf{Ax} = \mathbf{b}, \tag{5}$$

$$A = \begin{bmatrix} X_1 & Y_1 & 1 & 0 & 0 & 0 & -x_1X_1 & -x_1Y_1 \\ 0 & 0 & 0 & X_1 & Y_1 & 1 & -y_1X_1 & -y_1Y_1 \\ X_2 & Y_2 & 1 & 0 & 0 & 0 & -x_2X_2 & -x_2Y_2 \\ 0 & 0 & 0 & X_2 & Y_2 & 1 & -y_2X_2 & -y_2Y_2 \\ \vdots & \vdots & & & & & & \vdots \\ X_n & Y_n & 1 & 0 & 0 & 0 & -x_nX_n & -x_nY_n \\ 0 & 0 & 0 & X_n & Y_n & 1 & -y_nX_n & -y_nY_n \end{bmatrix},$$

$$x^{\mathrm{T}} = [C_{11} \quad C_{12} \quad C_{14} \quad C_{21} \quad C_{22} \quad C_{24} \quad C_{31} \quad C_{32}],$$

$$b^T = [x_1 \quad y_1 \quad x_2 \quad y_2 \ldots x_n \quad y_n],$$

where $(X_i, Y_i, 0)$ represents the world coordinates of the reference point $i$ and $(x_i, y_i)$ is the image coordinate of the same point. By solving Eq. 5, we find the eight necessary parameters for calibration. As each reference point generates two equations, we need at least four of such points. In case of more than four points, the system in Eq. 5 can be solved by considering the least-square method. To obtain these reference points, we measure the length of the lines in the playing field and detect their corresponding intersection points.

#### 3.5.2. Determination of the players location in the image

The players location in the image is determined by the position of the players feet in the blob. This can be done by finding the maximum $y$ coordinate of the blobs and its middle point in the $x$ coordinate. However, this method fails when lower parts of the player are lost during the segmentation or when shadows are present in the segmented blob, as shown in Fig. 20. In this work, we consider this situation by computing the players size (height) based on their current location and their maximal (near the camera) and minimal (far from the camera) estimated size in the image. The $y$ coordinate of a player location is obtained by adding the computed size of this player to the coordinate of the uppermost pixel of the corresponding blob. Here, we consider the fact that by taking into account this blob pixel, we avoid including the shadow of the player in the definition of its location, as illustrated in Fig. 20. This consideration is valid for all cases in which the shadows of the players are located below their heads. This configuration depends on the placement of the cameras with respect to the playing field.

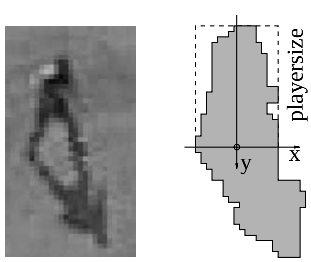


Fig. 20. Definition of the player location based on its estimated size in the image.

Since the maximal and minimal size of the player is defined by considering its feet location in the image, the estimated vertical location of the player matching the expected player size, in the corresponding blob, can be computed as follows. Let $y$ be the vertical position of the feet given initially by

$$y = y_0 + psize_{\min},$$

where $y_0$ is the top vertical pixel of the player blob and $psize_{\min}$ is the minimal estimated player size. At each step, the value $y$ is increased while $psize_y > y - y_0$, $psize_y$ being the player expected size at the $y$ position defined as

$$psize_y = psize_{\min} + (psize_{\max} - psize_{\min}) * y/image_y,$$

where $psize_{\max}$ is the maximum estimated player size and $image_y$ is the vertical dimension of the image.

From the image coordinate $(x, y)$, we calculate the player coordinate $(X, Y)$, in the world coordinate system, by substituting the value of $(x, y)$ in Eqs. (1) and (2), and assuming always that the $Z$ coordinate of the players is equal to zero.

### 3.6. The minimal path searching

The tracking of a player starts by its blob definition and the corresponding node identification in the constructed graph. The color of this blob is used as the team reference for the rest of the tracking. At each step, we traverse the graph by considering a minimal path using the distance information between the blobs. This method represents an easy way to track isolated players since there is only one edge to be considered at each step. If an edge connecting two nodes does not exist, we assume that the player was lost, probably because of an error during the segmentation step. In such a case, we can extrapolate the location of the player in the image, using a prediction method, and we try to find a nearest blob in the next frames. A common example of non-detection of a player occurs when it falls down and its size becomes smaller. The tracking of the players in case of contacts or occlusions by other players is more difficult to consider and needs additional information to be implemented. Although the splitting of the blobs (Section 3.4) is supposed to eliminate this problem, some situations remain in which one node may have more than one player.

This work pays special attention to the case of two player occlusions since it is one of the most common situation in soccer (e.g., two players disputing the control of the ball), together with the tracking of the isolated players. Occlusions of more than two players are considered only in case of short temporal contacts.

Soccer, as many other sports, is a tactical game and the success of a team depends also on the way its players mark the opponent ones. This is why a defender is always very close to the offensive player of the other team. Besides the cases of crossing or passing by, all these events generate occlusions or contacts between two players. The tracking, in such a situation, is performed by taking into account the location of the blob containing these two players and by considering that the trajectory is the same for both of them. For some short occlusions, if the current computed position of a player is out of the expected distance, with respect to the previous frame, then this player location can be defined by prediction.

To maintain the right trajectory of the players after an occlusion, it is important to correctly identify their corresponding nodes in the graph. For this purpose, we consider the color of the blobs (Section 3.3) as well as the distance information conveyed by the graph. As stated before, one blob color is defined for each team, according to the algorithm described in Section 3.3. This information is used as a weight $w$ in the determination of a minimal value of the graph edges. Experimentally, we set $w = 1$ when the color of the blob is the same as the one defined in the beginning of the tracking, and $w = 4$ when these colors are different. If the blob color cannot be identified ($C_{\text{blob}} = 0$, as stated in Section 3.3), we set $w = 2$.

Fig. 21 shows examples of two players from different teams running together (the lines indicate the corresponding paths in the figure). In such a case, it is difficult to correctly split the blobs and assign only one trajectory for both players. After a certain number of frames, these connected components are separated and have their own trajectory. This trajectory definition is possible by considering the color of the blobs associated with the corresponding players.

Finally, we need to take into account situations in which this blob color information does not allow a correct tracking, as it happens, for example, when the players in contact belong to the same team (Fig. 22). In this situation, we can use the trajectories direction ($D$) and the the mean velocities ($V_{\text{m}}$) of the players before and after the occlusion (Fig. 23). If this contact is short, in terms of number of frames, we maintain the same direction for each player as before the occlusion. Further, if the contact is long and without much motion then, heuristically, we can consider that each player moves in opposite directions after an occlusion event.

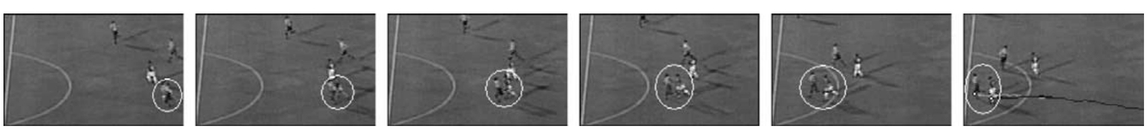

Fig. 21. Tracking players. Situation in which two players from different teams are running together.

Fig. 22. Tracking players. A complex situation representing occlusions of players from the same team.

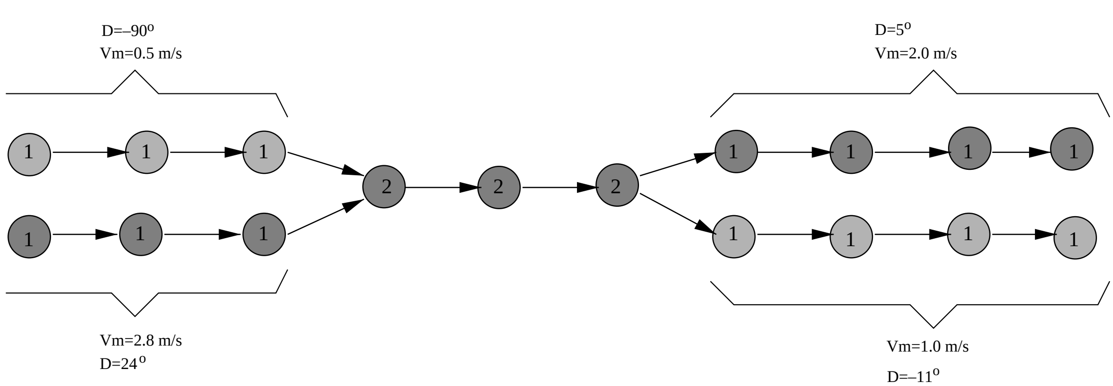


Fig. 23. Correct path definition by using the velocity and the direction information of two selected players in Fig. 22.

At each step of the tracking method, the real 2D coordinates of the players on the field is reconstructed, using the calibration parameters and the image coordinates representing their location in the image. Initially, it is also defined the field of view of each camera. Based on this information, we can determine exactly the cameras a tracked player is visible from and choose the one that better focus on this player. We select the best view depending on the distance of the blob to the center of the field of camera view. The overlapped region, i.e, the region visible by more than one camera, is also considered here in the solution of occlusion problems. For example, if a tracked player in this overlapped region appears occluded in the current camera, then we can verify if this player becomes isolated in another view. This kind of solution is also explored in [12] which considers eight cameras covering the whole goal area.

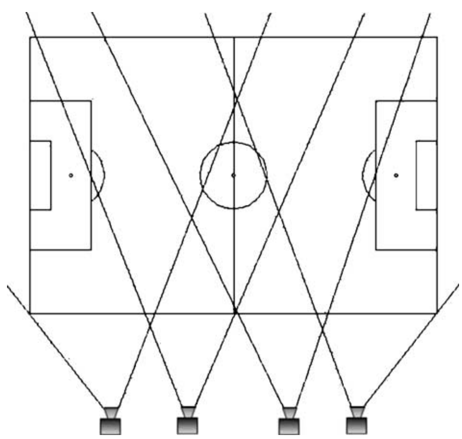

Fig. 24. Example of cameras placement.

## 4. Applications and some results

To illustrate the proposed method, we consider a game of the Brazilian championship filmed by four digital cameras JVC GR-DVL9500, with image dimension of $480 \times 720$, placed at one side of the pitch and at the highest location of the stadium (at a height of 20 m and a distance of 40 m, approximately, from the field line). Position and distance between cameras were chosen arbitrarily so that one camera covered at least one-fourth of the playing field and had extra overlapping regions. Fig. 24 shows the placement of these cameras which defines the size of the payers in the images varying from $5 \times 10$ to $15 \times 30$ pixels.

By considering four views, we ensure that we have enough information to calibrate the cameras, while preserving good features (e.g., size and intensity) of the players. Here, for example, we can easily remove noise from the processed images based on the size of the segmented players. Also, as we have seen before, enough color information from the segmented objects can be very useful in the tracking process, mainly when two or more players are in contact.

The above mentioned game was filmed in the afternoon of a sunny day with a partially clouded sky. The results discussed here concern a 10 min game processed by the segmentation algorithm described in Section 2. Further, the tracking method was performed for the 22 players from

Table 1
Evaluation of the tracking algorithm for a 10 min sequence

| Player | Solved occlusions or contacts | Non-solved occlusions of 2 or 3 players | Non-solved occlusions (more than 3 players) | Lost players | Stops during the tracking | Manually tracked frames | Frames with correctly solved occlusions |
|---|---|---|---|---|---|---|---|
| 1 | 8 | 1 | 0 | 0 | 1 | 1 | 107 |
| 2 | 38 | 8 | 4 | 0 | 12 | 418 | 446 |
| 3 | 38 | 3 | 3 | 0 | 6 | 254 | 640 |
| 4 | 51 | 7 | 4 | 0 | 11 | 111 | 664 |
| 5 | 52 | 8 | 2 | 0 | 10 | 318 | 672 |
| 6 | 52 | 8 | 2 | 4 | 14 | 104 | 721 |
| 7 | 36 | 2 | 1 | 2 | 5 | 48 | 422 |
| 8 | 47 | 7 | 5 | 1 | 13 | 399 | 516 |
| 9 | 55 | 9 | 2 | 1 | 12 | 345 | 1229 |
| 10 | 57 | 5 | 4 | 2 | 11 | 353 | 585 |
| 11 | 49 | 5 | 6 | 0 | 11 | 492 | 1196 |
| 12 | 2 | 1 | 2 | 0 | 3 | 45 | 27 |
| 13 | 47 | 2 | 4 | 4 | 10 | 368 | 1028 |
| 14 | 44 | 3 | 4 | 0 | 7 | 428 | 822 |
| 15 | 26 | 4 | 4 | 0 | 8 | 267 | 655 |
| 16 | 51 | 12 | 4 | 0 | 16 | 469 | 538 |
| 17 | 23 | 4 | 3 | 1 | 8 | 79 | 315 |
| 18 | 52 | 3 | 3 | 4 | 10 | 187 | 718 |
| 19 | 52 | 6 | 2 | 1 | 9 | 67 | 544 |
| 20 | 60 | 4 | 2 | 4 | 10 | 391 | 796 |
| 21 | 43 | 4 | 3 | 2 | 9 | 179 | 634 |
| 22 | 59 | 9 | 1 | 0 | 10 | 111 | 830 |
| Mean | 46.6 | 5.7 | 3.1 | 1.3 | 10.1 | 270 | 699 |

both teams. Table 1 illustrates some numerical results of the test. The second column shows, for each player in column 1, the number of occlusions or contacts correctly processed by the method, i.e, the cases in which the blobs were correctly tracked during the occlusion events. The third column shows the situation in which the tracking failed because of two or three player occlusions. The fourth and fifth columns show the case of fails caused by occlusions of more than three players and by problems such as segmentation errors or by players out of the playing field. The sixth column shows the total number of stops during the tracking caused by all the above mentioned events.

Finally, the seventh column shows the total number of frames whose components tracking was done manually due to errors in the automatic tracking. This manual process consists in following the players position on the screen by a displacement of the mouse cursor whose position on the successive frames is automatically detected. Normally, in case of two or three player occlusions, the number of manual corrections is low (only 10% of the total number of manual corrections). In case of occlusions of more than three players caused, for example, by corners or free kicks, the number of manual corrections is higher (more than 90% of these corrections). During the 10 min of the analyzed sequence, there were two corners and three free kicks (Fig. 25). The final column in Table 1 shows the number of frames having occlusion events correctly solved. The last line of this table shows the mean value for each considered case, by taking into account all the players, except the goalkeepers (players number 1 and 12). These players were excluded from the mean because of the few number of occlusions concerning them (as we can see in the second, third, and fourth columns in Table 1), and the success of the tracking method, in such a case (99% ), with only two stops on average.

Fig. 25. Examples of corner and free kick in the video. In such situations, some players were manually tracked.

Table 2
Number of stops reduced by the method

| | Number of solved occlusions or contacts | Number of stops during the tracking |
|---|---|---|
| Mean | 46.6 | 10.1 |
| Percentage | 82 | 18 |

If we consider that the number of solved occlusions is represented mostly by the cases of two or three player in contact, then we correctly solved 89% of these cases for the analyzed video sequence. The mean number of stops (last line in Table 1) shows that during the tracking of each player we stopped one time each 1 min, i.e., each 450 frames. By considering the total number of stops and the number of solved occlusions, we have a reduction of 82% of the number of stops in the tracking process, as we can see in Table 2. The mean number of manually tracked frames represents only 6% of the total number of frames (Table 3), and from 94% of the automatically tracked frames, 15% correspond to cases of occlusions.

The quality of the images is very important if we want to deal with the occlusion problems, the size of the represented players being proportional to their distance from the camera. For example, in Table 1, we can see that we stopped 14 and 5 times for the players 6 and 7, respectively. By analyzing these players location on the field, we remark that they are wing players: one on the left side (the upper region of the images) and the other on the right side of the field (the lower region of the images), defining smaller and bigger representations of these players, respectively. Hence, a straightforward way to improve the whole tracking method proposed here is by the use of multiple cameras positioned on both sides of the playing field.

The data obtained from the method of tracking can be used to determine kinematical variables of the soccer players movements such as covered distance, velocity, and acceleration. These parameters can be useful in the classification of the activities realized by the players (walking, jogging, running, and sprinting), as well as in the tactical analysis of their movements. Fig. 26 shows an example of application of our tracking system. It presents the velocity distribution of the eleven players during 45 min of a game. The value showed in each bar represents the permanence time in minutes of the corresponding player at predefined velocity ranges. These ranges are, respectively, [0–2 m/s], [2–4 m/s] [4–6 m/s], and above 6 m/s for each bar, from left to right, in the sets

Table 3
Number of automatically tracked frames

| | Number of manually tracked frames | Number of automatically tracked frames (cases of occlusions) | Number of automatically tracked frames (cases of isolated players) |
|---|---|---|---|
| Mean | 270 | 699 | 3531 |
| Percentage | 6 | 15.5 | 78.5 |

of four bars shown in Fig. 26. Fig. 27 gives the total distance covered by the set of players and Fig. 28 shows a visual representation of a player displacement during 45 min.

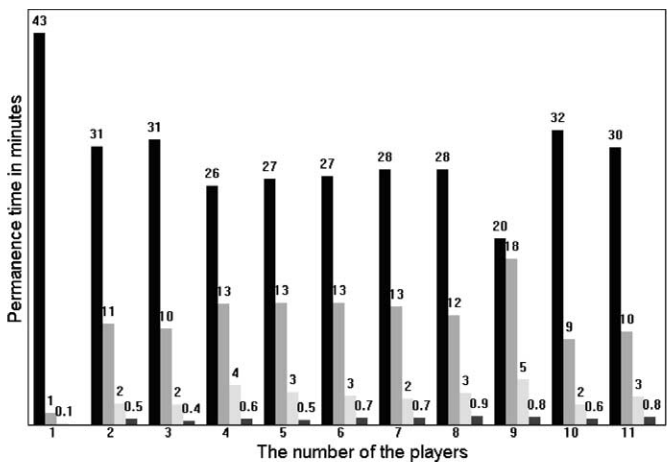


Fig. 26. Example of application: the velocity distribution of the players.

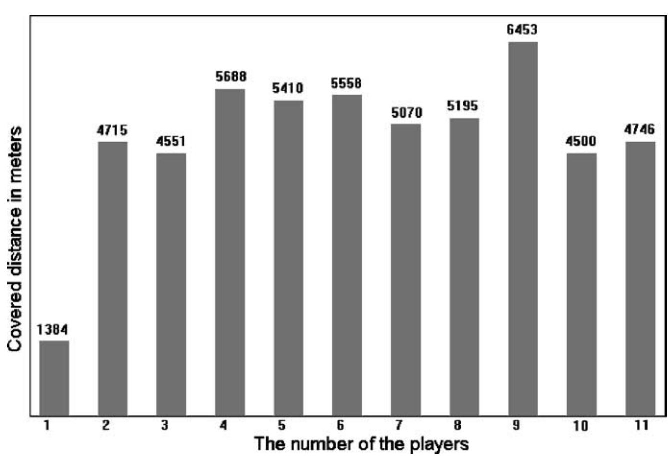


Fig. 27. Example of application: the total distance in meters covered by the players.

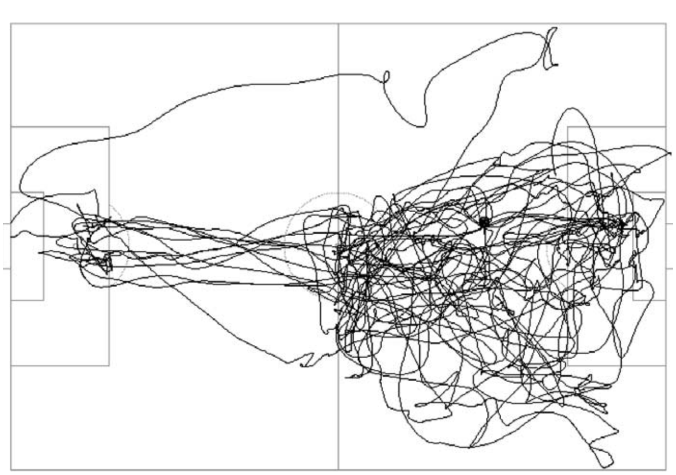

Fig. 28. The visual representation of a player displacement on the field.

## 5. Conclusions

In this work, we proposed a method for tracking soccer players which considers at least four static cameras. The tracking algorithm is based on the search of paths in a graph defined by blobs representing segmented players. The different cases of occlusions or contact of these players are treated by splitting the corresponding blobs and taking into account features such as number of components, area of the blobs, players trajectory, and so forth. This algorithm has been used in a large number of soccer video images and its performance was illustrated here through a 10 min processing whose statistical results show the effectiveness of the method. One of the main contributions of this work is the proposition of a methodology that makes possible the analysis of players movement through a more effective tracking procedure based on the improvement and combination of some techniques involving segmentation, cameras calibration, components identification, determination of the players location, and so on.

In this work, the blob splitting method based on a graph representation, which allows a forward and backward tracing, is a novelty in the solution of occlusion problems. Indeed, in our experiments, this solution increased the number of frames automatically processed by the tracking operations mainly in cases of 2 or 3 player occlusions. More complex situations of occlusions gathering many players is still an open problem faced by all the image processing-based approaches in which the segmentation of the components is an important step to be considered. The proposed methodology enables the development of systems for quantitative analysis of soccer players movements during a game. Although it is not possible to have a total automatic system, one of the objective here is to reduce the number of manual interventions by a user.

Unlike other works in which the analysis of short sequences of a game is presented, this paper illustrates the tracking results and some description of the performance, such as covered distance and mean velocity, of all the players during a whole game. The time needed for this analysis is approximately 90 min per player and includes the visual verification of the players trajectory. In this work, we were constrained to deal with gray-level information only due to the image quality defined by a small number of cameras covering the whole soccer field. With a narrower field of view that can be obtained by taking into account more cameras than the four ones considered here, one can incorporate to the tracking procedure the color information which can be useful, for example, in the solution of problems concerning occlusions.

## References

[1] J. Allen, R. Butterly, M. Welch, R. Wood, The physical and physiological value of 5-a-side soccer training to 11-a-side match play, J. Hum. Move. Stud. 34 (1998) 1–11.

[2] D.H. Ballard, C.M. Brown, Computer Vision, Prentice-Hall, Englewood Cliffs, NJ, 1982.
[3] J. Bangsbo, L. Norregaard, F. Thorso, Activity profile of competitions soccer, Can. J. Sport Sci. 16 (2) (1991) 110–116.
[4] A. Cavallaro, T. Ebrahimi, Video object extraction based on adaptive background and statistical change detection, in: Proc. SPIE Electonic Imaging—Visual Communication and Image Processing, 2001, pp. 465–475.
[5] I. Cohen, G. Medioni, Detecting and tracking moving objects for video surveillance, in: Proc. Computer Vision and Pattern Recognition, 1999, pp. 319–325.
[6] S. D'Ottavio, C. Tranquilii, rindimento di jogador di futbole, Sd5 - Rivista di Cultura Sportiva (1993) 24.
[7] W. Erdmann, Quantification of games—preliminary kinematic investigation in soccer, in: Science and Football II, 1991, pp. 165–174.
[8] A. François, G. Medioni, Adaptive color background modeling for real-time segmentation of video streams, in: Proc. Internat. Conf. on Image Science, System and Technology, 1999, pp. 227–232.
[9] I. Haritaoglu, D. Harwood, L. Davis, $\omega^4$: real-time surveillance of people and their activities, IEEE Trans. Pattern Anal. Mach. Intell. 22 (8) (2000) 809–830.
[10] E. Hennig, R. Briehle, Game analysis by gps satellite tracking of soccer players, in: Proc. XI Congress of Canadian Society of Biomechanics, 2000, p. 44.
[11] S.S. Intille, A.F. Bobick, Visual tracking using closed-worlds, Technical report, M.I.T Media Laboratory Perceptual Computing Section Technical Report No. 294, 1994.
[12] S. Iwase, H. Saito, Tracking soccer players based on homography among multiple views, Proc. SPIE (2003) 283–292.
[13] Jinman Kang, Isaac Cohen, Gerard Medioni, Tracking people in crowded scenes across multiple cameras, in: Asian Conf. on Computer Vision, 2004.
[14] K. Matsui, M. Iwase, M. Agata, T. Tanaka, N. Ohnishi, Soccer image sequence computed by virtual camera, in: Proc. IEEE Conf. on Computer Vision and Pattern, 1998, pp. 23–25.
[15] S.R. Mayhew, H.A. Wenger, Time-motion analysis of professional soccer, J. Human Move. Stud. 11 (1985) 49–52.
[16] S.J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, H. Wechsler, Tracking groups of people, Comput. Vision Image Understand. 80 (2000) 42–56.
[17] C.J. Needman, R.D. Boyle, Tracking multiple sport players through occlusion, congestion and scale, in: British Machine Vision Conference, 2001, pp. 93–102.
[18] T. Reilly, V. Thomas, A motion analysis of work-rate in different positional roles in professional football match play, J. Human Move. Stud. 2 (1976) 87–97.
[19] J. Serra, Image Analysis and Mathematical Morphology, Academic Press, London, 1992 (second printing).
[20] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, in: Proc. Computer Vision and Pattern Recognition, 1999, pp. 246–252.
[21] T. Taki, J. Hasegawa, T. Fukumura, Development of motion analysis system for quantitative evaluation of team work in soccer game, in: Proc. Internat. Conf. on Image Processing, 1996, pp. 815–818.
[22] N. Vandenbroucke, L. Macaire, J.G. Postaire, Color pixel classification in an color hybrid color space, in: Proc. Internat. Conf. on Image Processing, 1998, pp. 176–180.
[23] R.T. Withers, Z. Maricic, S. Wasilewski, L. Kelly, Match analyses of australian professional soccer players, J. Human Move. Stud. 8 (1982) 159–176.
[24] H. Kim, Y. Seo, S. Choi, and K.S. Hong, Where are the ball and players? Soccer game analysis with color-based tracking and image mosaick, in: Proc. Internat. Conf. on Image Analysis and Processing, 1997, pp. 196–203.