An Investigation into the Feasibility of Real-Time Soccer Offside Detection From a Multiple Camera System

Tiziana D'Orazio, *Member IEEE,* Marco Leo, Paolo Spagnolo, Pier Luigi Mazzeo, Nicola Mosca, Massimiliano Nitti, and Arcangelo Distante

***Abstract*—In this paper, we investigate on the feasibility of multiple camera system for automatic offside detection. We propose six fixed cameras, properly placed on the two sides of the soccer field (three for each side) to reduce perspective and occlusion errors. The images acquired by the synchronized cameras are processed to detect the players' position and the ball position in real-time; a multiple view analysis is carried out to evaluate the offside event, considering the position of all the players in the field, determining the players who passed the ball, and determining if active offside condition occurred. The whole system has been validated using real-time images acquired during official soccer matches, and quantitative results on the system accuracy were obtained comparing the system responses with the ground truth data generated manually on a number of extracted significant sequences.**

***Index Terms*—3-D trajectory analysis, multiple cameras, player and ball tracking.**

I. Introduction

AUTOMATIC event detection in sport contexts is gaining attention in the computer vision research community due to the needs of decision support tools that could prevent wrong interpretations due to perspective errors, occlusions, and high velocity of events. Despite a lot of research efforts for soccer summarization in broadcast video, algorithms for real-time analysis of soccer events have to be further investigated. The high velocity and complexity of soccer events impose strict real-time constraints that usually make the summarization algorithms inapplicable in this context. In particular offside detection during soccer matches is a very complex task because it is necessary to evaluate simultaneously several events that may occur in a large area of the field: the player who passed the ball, the players position in the field at the exact moment in which the ball is passed, the position of the player who is going to receive the ball and also the application of the active/passive offside rule.

In this paper, we investigate the feasibility of a visual system able both to detect the offside event through real-time processing of the acquired images and to provide an objective analysis showing the image sequence recording the event under consideration. Broadcast images cannot be used for this problem because the position of the cameras cannot guarantee the exact evaluation of the player position in the pitch. For this reason, we used fixed cameras covering the whole pitch with a small overlapping among their fields of view enabling ball and player tracking during all the match. We investigated the problems of detecting the players and the ball positions in the field and of interpreting their trajectories to evaluate the offside occurrence.

A. Related Work

In the recent decades, many papers addressing player segmentation and tracking, player position detection, ball recognition and location on the playfield have been reported. Most of them work on broadcast images with the aim of recognizing players' actions for video summarization [1] and virtual view replay [2]. A few works have been presented for real-time processing of sport images dealing with just one of the above mentioned problems.

In [3], a background recovering algorithm for soccer player segmentation has been presented which takes into account the specific problem of lighting changes and the fact that slow and fast motion in the scene can be considered. The problem of eliminating shadows to obtain a good segmentation for soccer player detection has been addressed in [4]. An unsupervised learning procedure determines the RGB color distributions of the foreground and shadow classes.

The problem of multiview tracking of soccer players to localize the targets on the ground field has been faced with different approaches in literature. In [6], a geometrical module performs the image-to-model homography estimation, while a supervisor collects and fuses the tracking data provided by each view. The local tracking is carried out by extracting interest points described by their position and local appearance. Color histograms within elliptic regions are used in [7] for tracking players on the pitch during a sport game. Since the players are being tracked in their natural environment,

Manuscript received June 12, 2008; revised January 9, 2009. First version published July 7, 2009; current version published December 1, 2009. This work was supported under Grant Federazione Italiana Giuoco Calcio. An international patent was granted for this system in 2007 [24]. This paper was recommended by Associate Editor S. Sull.

The authors are with the Institute of Intelligent Systems for Automation, Italian National Research Council, Bari 70126, Italy (e-mail: dorazio@ba.issia.cnr.it; leo@ba.issia.cnr.it; spagnolo@ba.issia.cnr.it; mazzeo@ba.issia.cnr.it; nmosca@ba.issia.cnr.it; nitti@ba.issia.cnr.it; distante@ba.issia.cnr.it

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2009.2026817

which is constrained by certain rules, the authors use the knowledge of the scene to model the context and improve the reliability of tracking. Reference histograms of players and background are used to evaluate the presence of players in a given state. The main constraint of this approach, limiting its application to other contexts, is that the cameras observe the playground from above with the optical axis perpendicular to the floor. This means that is highly unlikely to observe complete occlusion during the match. In [8], the tracking is performed through a graph representation. Occlusions are treated by splitting segmented blobs using morphological operators and a backward and forward graph representation. Overlapping regions related to the set of four cameras placed on the pitch are used for synchronization and to solve some cases of occlusions. A collaborative multicamera tracking has been presented in [9] that addresses several issues including occlusions and propagation of wrong information. Each target is tracked in each view by a dedicated particle filter-based local tracker. The trackers in different views interact with each other via belief propagation so that a local tracker operating in one view takes advantage of additional information from other views.

The use of color features has been considered in literature not only to improve the tracking abilities but also to separate players belonging to different teams. In [10], color classification and segmentation are used to obtain the blobs corresponding to the strip of the teams and the referees. The authors use a previously learned set of color classes and find the region of interest by mapping the image pixels into the respective color classes, and then grouping the pixels using morphological operators. Hybrid color spaces have been investigated in [11] to detect the space producing the best discrimination between pixels belonging to players of opponent teams. A method for tracking football players with multiple cameras has been presented in [12]. The system comprises two processing stages, operating on data coming from a single data and then multiple cameras. Category classification of the player's uniform is carried out by intersecting the RGB histograms of segmented blobs and five model histograms obtained by a semi-supervised strategy where an operator labels examples of players observation before the match starts. A semi-automatic system is developed in [13] for player ball possession acquisition with broadcast soccer video. Support vector machine on the color histogram are used for teamship recognition. For each type of people the authors manually identify a color differentiating them from other people in advance. For each color, several color bins are built and used to evaluate the distribution of pixels having colors falling into the prebuilt bins. This distribution forms a color histogram used to evaluate his team through a support vector machine. In [14], players and ball positions are estimated from monocular soccer video by the calibration of the camera through homography between the image and the playfield, and self-calibration for rotating and zooming camera.

The problem of ball tracking is more difficult than that of players due to the ball small size in an image and abrupt changes of direction. Previous ball detection methods are not applicable because the ball is not sufficiently large and the ball texture is not appreciable [15]. Some works have been presented based on the ball route analysis. In [16], a trajectory-based detection and tracking algorithm is presented for locating the ball in broadcast soccer video. The ball tracking problem in soccer sequences taken from fixed cameras that produce small images of the ball almost blurred, has been considered in [18]. The ball tracking is batch processed at every mth frame in order to produce an accumulation image that contains pixels only from the ball area. A particle filter is applied to the accumulation image to not only distinguish the ball from noise but also to decide if the ball is visible or not. The spatiotemporal relationships between players, lines and the ball have been considered in [19] to estimate the ball route also during overlaps with players and lines. Fixed cameras have been used in [5] to detect the 3-D ball trajectory and also to interpret four different motion phases as rolling, flying, in possession and out of play. In [17], 16 PAL cameras, with a very narrow field of view, are proposed to track players, to extract 3-D ball trajectories and to calculate their world coordinates and positions with respect to the offside line. Unfortunately, there is no evidence neither of the application of the complex International Football Federation (FIFA) law of the game regarding the offside detection neither of extended tests in multiple matches.

Most of the related works provide good solutions for many problems of player and ball tracking but either they were applied to broadcast images and are not suitable for images with large camera views or did not consider the processing constraint of real-time applications. Their use into automatic systems for offside detection is not immediate since it is required the actual functioning of algorithms in very challenging situations such as 90 min of continuous processing with varying lighting conditions, none initializations by the user, large camera views and so on.

B. Our Contribution

The offside official rules from the FIFA laws of the game [20] are very difficult both to be understood by spectators and to be put into practice by experienced people such as referees. In this paper, we investigate the feasibility of a real-time visual system for offside detection during soccer matches. This paper, requested and supported by the Italian Football Federation, was intended as an automatic support tool for referees to reduce wrong interpretations and also to provide an objective analysis of complex events. The main novelties of this paper are in part in the proposed hardware architecture (in terms of imaging, processing, and communication) and in the software modules that were devised in order to be a compromise between reliability and real-time performance.

The hardware architecture consists of six high resolution cameras that have been placed on the two sides of the field, with the optical axes parallel to the goal line plane to reduce perspective errors. The acquired images are synchronized and transferred to six nodes by fiber optic cables. Each node, configured with two processors with hyper-threading technology, records all the images of the match on its internal storage unit, displays the acquired images and simultaneously processes them with parallel threads, in an asynchronous way

with respect to the other nodes. The six nodes, are connected to a central node, which has the supervisor function. It has the main tasks of data synchronization and decision making to evaluate offside events.

Fig. 1 shows the processing steps executed by each node and by the supervisor that will be detailed in the following sections. Here we resume the main novelties proposed, in this paper. First, a background subtraction algorithm based on the local energy evaluation along a temporal window allows the detection of moving areas that can be associated both to players and ball candidate regions. The algorithm has been proven to be robust in the background modeling when moving foreground objects are present in the scene and also in the background updating when a foreground object keeps still for a long period (it avoids the ghost problem in the images). Besides a temporal image analysis is used to detect the camera autoiris and prevent wrong segmentations due to an inadequate background. A connectivity analysis based on topological considerations of the moving areas allows the segmentation of foreground objects removing at the same time the shadows that could compromise further processing.

Second, the player team classification is carried out without any human intervention by using an unsupervised clustering algorithm that processes the images when players enter the stadium generating automatically the five classes corresponding to the strips of the two teams, the two goalkeepers and the referees. Normalized color histograms are used to generate the class prototypes and are continuously updated during the match to handle the varying lighting conditions. The classification algorithm is able to produce reliable results independently of the player postures in the field. A tracking algorithm follows the players while moving and is able to solve the merge situations when two or more players form a group region.

Third, the ball is detected and, by the analysis of its cinematic parameters, pass events are recognized. Due to the large camera views, the ball is small and has irregular shape according to the position in the field (relative to the camera position). The ball detection algorithm, starting from candidate moving areas whose dimensions are compatible with the expected ball size, compares the selected region with different sets of reference examples of ball. A probability map, based on the past information about the ball motion, is used to evaluate the likelihood that any point of the image represents the ball center.

Finally, the data coming from different cameras are projected onto a virtual field by a central process that interprets the players and ball trajectories. The application of the active/passive offside rules, as described in [20], imposes the temporal analysis of the ball trajectory from the player who passes the ball to the one who receives the ball. Then, in the virtual field a multiple trajectory analysis is applied to detect the possible intersections and evaluate the semantics of the occurred events.

C. Structure of the Paper

The rest of the paper is organized as follows. Section II describes the processing steps of each node. The data fusion and decision making processes executed by the supervisor node are illustrated in Section III. Experimental results are reported in Section IV. Section V reports the discussion and some conclusions.

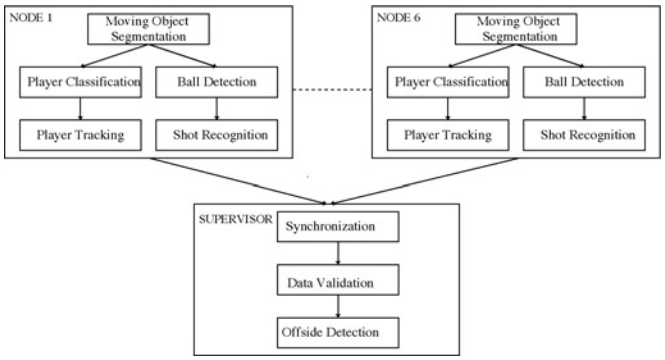


Fig. 1. Scheme of the visual system.

II. Processing Nodes

A. Moving Object Segmentation

In this paper, we used a background subtraction algorithm for motion detection. We started from the approach [21] for background creation and maintenance: a pixel (x, y) is considered as a moving one if it differs from the background model by more than twice the standard deviation. In order to build a background model without being affected by moving foreground objects, (moving players are always present in the scenes) we introduced the energy information of each image point, evaluated in a small sliding temporal window, to distinguish static points from moving ones. The window dimension W has been set to 60 frames (that correspond to about 2.5 s), because experimental observations have demonstrated that it is quite improbable to have players that keep still more than 2.5 s. We applied a coarse-to-fine approach: the first image of each window is the coarse background model $B_C(x, y)$. We evaluated mean and standard deviation only for those points whose intensity value is substantially unchanged with respect to the coarse background model, that is $|I^t(x, y) - B_C(x, y)| < \text{th}$ where th is a threshold experimentally selected. In this way, at the end of the analysis of the first W frames, for each point the algorithm evaluates the energy content as follows:

$$E(x, y) = \sum_{t \in W} |I^t(x, y) - B_C(x, y)|^2. \tag{1}$$

After the processing of the first W fames, the first fine model of the background B_F is generated, as

$$B_F(x, y) = \begin{cases} [\mu(x, y), \sigma(x, y)], & \text{if} \quad E(x, y) < \text{th(W)} \\ \phi, & \text{if} \quad E(x, y) > \text{th(W)}. \end{cases} \tag{2}$$

The threshold th(W) is proportional to the window dimension W. A low energy content means that the considered point is a static one and the corresponding statistics are included in the background model, whereas high energy points, corresponding to foreground or moving background objects cannot

Fig. 2. From left to right: the original, the foreground, the segmented images.

contribute to the model. The whole procedure is iterated on another sequence of W frames, starting from the frame $W+1$ that becomes the new coarse background model. The relevant difference with (2) is that now the new statistical parameters are averaged with the previous values, if they are present; otherwise, they become the new statistical model values. So, the new formulation of (2) becomes

$$B_F(x,y) = \begin{cases} [\mu(x,y), \sigma(x,y)] \\ \quad \text{if,} \quad \mathrm{E(x,y)} < \mathrm{th(W)} \wedge \mathrm{B_F(x,y)} = \phi \\ \beta B_F(x,y) + (1-\beta)\mu[\mu(x,y), \sigma(x,y)] \\ \quad \text{if,} \quad \mathrm{E(x,y)} < \mathrm{th(W)} \wedge \mathrm{B_F(x,y)} \neq \phi \\ \phi \quad \text{if,} \quad \mathrm{E(x,y)} > \mathrm{th(W)}. \end{cases} \tag{3}$$

The parameter β is the classic updating parameter introduced in several works on background subtraction [21] (generally set to 0.1). The whole above described procedure is iterated, in order to adapt the background model to the continuous variations occurring in the lighting conditions. However, since the system has to work for all the match, the cameras have to periodically modify the iris apertures in order to maintain an average intensity level. When the iris is modified, the background is no longer consistent with the current image, and it has to be quickly updated. For this reason we modified the system parameters according to the quantity of the global motion in the image.

In order to generate the foreground regions a connectivity analysis has been applied to the resulting segmented images. The connectivity analysis eliminates shadows by using geometrical considerations about the shape of each region. Extensions of areas in the orthogonal directions with respect to the expected vertical position of players are removed during the construction of connected regions. In Fig. 2, the original, the foreground, and the segmented images are shown. In the foreground image, the players have been extracted with their shadows. In the segmented image, after the connectivity analysis, it is clearly visible that the blob dimensions are smaller than the actual foreground regions that contain shadows.

B. Players and Referee Classification

After the player segmentation, it is necessary to correctly assign each of them to the relative class. The players' textures are not known in the beginning of the match, so they can vary for each game. For this reason it is necessary to classify moving objects by using an unsupervised procedure, that does not require any human intervention. The classification procedure is composed by two steps: firstly, the classes are created by means of a clustering algorithm based on a modified version of the BSAS algorithm [23], an unsupervised approach substantially independent of human interaction. Then, at runtime, each segmented object is assigned to one of the classes previously extracted.

The clustering procedure is very important for the reliable classification of players and referee. Since our main objective was to be independent of the manual selection of players for building the clusters, we decided to collect randomly in a training set a number of segmented objects and provide the corresponding normalized histogram to the BSAS algorithm detecting the interest classes. The BSAS algorithm requires only a similarity measure $d(x, C)$ and a similarity threshold (th). The idea is to assign every new vector to an existing cluster or create a new cluster for this sample, depending on the distance to the already defined clusters. Experiments demonstrated that final results depend on the chosen distance function and on the order in which the samples are presented. Besides it is very important the value of the threshold (th) that effects the number of resulting clusters. A small value for (th) produces unnecessary clusters, whereas large value provides few clusters. In our implementation we have modified the original algorithm fixing initially the threshold (th) to a small value and afterwards increasing it if the number of detected clusters exceeds a predefined value. In this way the algorithm converges on the correct cluster configuration with the best (smallest) value of (th). Finally, in order to smooth the dependence from the order in which the samples are presented, a merge procedure is carried out on the output clusters, using (th) as a merge threshold. If the algorithm is not able to detect a consistent number of clusters, the training set is cleared, a new training set is built, and the whole training procedure is repeated. Typically this situation occurs if the training set is composed only by objects relative to the same class (i.e., players of the same team).

At runtime, each segmented player is compared with the cluster prototypes. The Manhattan distance is used to select the winner class C_k by means of a minimum distance criteria. The winner prototype is updated in order to adapt itself to the variations in light conditions as follows:

$$C_k = \frac{1}{w_k + 1}(w_k C_k + V) \tag{4}$$

where C_k is the prototype of the cluster k, V is the feature vector of the examined objects, and w_k is the number of objects classified as belonging to the cluster k in the last two temporal windows W. In this way, we are able to manage the natural model drifts that continuously occur during the whole match.

C. Player Tracking

After the moving object segmentation and classification we can represent each player with a Bounding Box (BB). The state vector of the ith player is defined by $x_t^i = (p_{x_t}^i, v_{x_t}^i, d_{x_t}^i, l_{x_t}^i, c_{x_t}^i, s_{x_t}^i)$ where:

1) $p_{x_t}^i$, $v_{x_t}^i$, and $d_{x_t}^i$ represent the BB position, velocity, and dimension respectively;
2) $s_{x_t}^i$ is the BB status. It assumes the values: 1 for a single blob in the image, 2 for a merge blob, 3 for an exiting blob, 4 for a disappeared blob, and 5 for single blob belonging to a group blob;

3) $l^i_{x_t}$ is a single label if the blob is a single blob, or a set of labels if the blob is a merge blob;
4) $c^i_{x_t}$ is a single class number (ranging from 1 to 5 as the output of the classification step) if the blob is a single blob or a set of class numbers if the blob is a merge blob.

We denote the multipeople configuration at time t with $X_t = \{x^i_t | i = 1, ..., N_t\}$, where N_t is the number of predicted BB in the image. In the same way we describe the measurement vector $Z_t = \{z^j_t | j = 1, ..., M_t\}$ where z^j_t are the observation instance vectors $z^j_t = (p^j_{z_t}, c^j_{z_t}, d^j_{z_t})$, M_t is the number of BB observed at the time t. The observations Z_t are the results of the segmentation and classification steps. They are independent of previous instances, but to avoid false blobs due to noise (after a background updating many artifacts can be introduced in the segmented image) they have to be validated by successive observations. Therefore, M_t is not the number of BB observed but actually the number of BB validated by consecutive and coherent temporal and spatial observations.

At each step we have to predict the new state configuration considering the past state evolution and then validate this prediction with the new measurements. Suppose a linear model (f) of the player motion we can predict the new state configuration as $X_t = f(X_{t-1}) + N$, where N is a gaussian noise. In this new prediction according to the positions and cinematic parameters of the players we can have:

1) x^i_t is a single track, that is a position change of one previous blob, if there is an x^j_{t-1} whose predicted position is in the image, (the state s^i_t is 1);
2) x^i_t is a merge, if there are two or more blobs (x^j_{t-1}, x^h_{t-1}, ...) whose predicted positions fall close in the image, (the state s^i_t is 2);
3) x^i_t can be an outgoing blob if there is a blob x^j_{t-1} whose predicted position is outside the image, (the state s^i_t is 3).

In the case of prediction of merge blob, the previous instances (x^j_{t-1}, x^h_{t-1}, ...) that generated the group blob are still maintained in the prediction (x^j_t, x^h_t, ...) with state s^j_t and s^h_t equals to 5. The prediction is also carried out for the x^m_{t-1} that have the status of disappeared blob (equal to 4).

As soon as the new measurement Z_t is available at the time t the prediction X_t has to be validated. By comparing all the observations z^h_t with $h \in M_t$ and the predictions x^i_t, several situations may happen: 1) some observations are close to the predictions have the same class; 2) there are some state predictions x^i_t that do not correspond to any observation; and 3) there are some observations z^h_t that do not match any state prediction. In the first case, the predictions are updated considering the information of the corresponding observations (in particular the position, velocity and dimension fields are estimated). In the second case, if the prediction x^i_t has not a correspondent among the observations and it isn't on the image border (it is not in an outgoing situation), it means that the foreground segmentator was not detecting the blob and then the status vector is maintained setting $s^i_t = 4$ (disappeared blob). In the third case different situations may be occurred: 1) the observation z^j_t could be a new entry blob if its $p^j_{z_t}$ is on the image border, then a new prediction x_t is generated with an incoming state; 2) the observation z^h_t could be a resumed blob if it is close to a prediction with a disappeared state; and 3) the observation z^h_t could be generated by noise, then a new entity x_t is created and observed along a temporal window until a decision on its persistency is taken.

Further analysis is required for merge blobs. We can predict that two or more blobs will merge (we set the status equal to 2), but since we need to maintain their vector status separated we have to split them in the corresponding observation. This splitting procedure can be difficult especially when two or more players are very close to each other and the occlusion is almost total. However, when a merge blob is detected by the tracking procedure, it also maintains the information about the class numbers of the grouped players and the labels identifying the single tracked blobs. Starting from this information, a splitting procedure evaluates the group blob and searches for subregions having the same color features of those searched. The search starts from the positions predicted by the single blob vector status and is enlarged as soon as the algorithm finds the best match. At the end of this step the state vector of each segmented blob is maintained by updating its position in the merge blob and setting the status $s^i_t = 5$ (blob belonging to a merge blob). The maintenance of state vectors for solved blobs in merge blobs allows us to recognize splitting situations. In fact when a split occurs (a single blob at time $t-1$ is divided in two or more blobs at time t) we have two or more observations z^j_t, z^h_t, z^k_t matching with a prediction x^l_t having $s^l_t = 2$ (merge status) and with two or more predictions x^m_t, x^n_t, x^r_t having $s^m_t = 5$ $s^n_t = 5$ $s^r_t = 5$. According to the number of objects in the merge blob and the number of observations we discriminate between a simple split or a complex split in single blobs and merge blobs. The decision is taken considering for each observation blob dimension, color features and the best correspondence with the prediction having the status equal to 5. In this way we assign to the single blob x^m_t the new status $s^m_t = 1$ (single track) and we remove or modify the merge blob x^l_t with $s^l_t = 2$ reducing its number of internal objects.

D. Ball and Shot Detection

The ball recognition process consists of two different steps. The first step selects, between all the moving areas, the regions that, for their dimensions, are candidate to contain the ball; the second step analyzes the region appearance in order to recognize the ball. The selection of candidate moving region depends on the information about previous occurrences of the ball. If no information about the ball position and motion is available (for example when the system starts-up) the selection of moving regions is performed by assuming that the ball is in a region separated from other moving regions (such as players) in the scene and its area is in a range depending on the imaging parameters. In this way all the regions having area not compatible with the ball size (players, referees, noise, etc.) are discarded whereas the remaining moving regions are labeled as candidate ball region and provided to a pattern recognition procedure. A correlation procedure has been used to evaluate shape and texture similarity between the candidate ball regions and a comparative set of manually selected reference examples

of ball. In order to manage different lighting conditions three different comparative sets were used: the first one contains examples of the ball taken in sunny days, the second one contains examples of the ball taken at night (using artificial lighting) and the third one contains examples of the ball taken on cloudy days. The number of examples in each set varies and is related to the appearance variation in the same lighting condition: for example on sunny days, the presence of self-shadows on the ball, makes it necessary to select more reference examples than in other cases. The selection of the set is done by the operator at the beginning of the match, but can be modified as soon as the lighting conditions change. In order to manage different ranges of the ball diameters, the ball examples have been separated in three sets, the first one for the ball examples with a diameter between (8, 10) pixels, the second one for diameters between (10, 12) pixels and the last one for (12, 14) pixels. In this case the selection of the proper set is done automatically by using the position of the ball candidate in the image to evaluate the expected ball diameter.

The correlation measure is evaluated between a patch centered on the candidate regions and a reference model for selected comparative set of ball examples. The reference model is the mean image of all the examples in a set. Only one correlation measure has to be performed for each comparative set. In this way the correlation value is computed and if it is greater than a selected threshold the corresponding moving region is labeled as Ball. If in the same image different regions produce high value then it is selected the one with the maximum correlation coefficient. The ball has to be detected in more consecutive images in order to be sure that a true positive has been found. In this case a different and more reliable procedure for selecting candidate moving regions is used (tracking phase). The local velocity V and the direction θ of the ball in the image plane are computed as follows:

$$V = \sqrt{V_x^2 + V_y^2}, \quad \theta = \arctan\left(\frac{V_y}{V_x}\right) \tag{5}$$

where

$$V_x = \frac{(P_{x_t} - P_{x_{t-n}})}{n}, \quad V_y = \frac{(P_{y_t} - P_{y_{t-n}})}{n} \tag{6}$$

and P_{x_t} is the position of the ball in image $I(t)$, $P_{x_{t-n}}$ is the position of the ball in image $I(t-n)$, T is camera frame rate and n is the number of frame between the past and actual ball detection (1 if the two frames are consecutive).

Then a ball position probability map, covering all the points of the processing image, is built as follows:

$$P(x, y) = \exp\left[\left(-|(x - |\widetilde{x} + V_x \text{sign}(\cos\theta)|) + (y - |\widetilde{y} + V_y \text{sign}(\sin\theta)|)|^2 / 2\sigma^2\right) / \sigma\sqrt{2\pi}\right] \tag{7}$$

where $(\widetilde{x}, \widetilde{y})$ is the last known ball position and

$$\sigma = \frac{R_p V_{\max} n}{R_{\text{cm}} T} \tag{8}$$

where R_p is the Ball radius in pixels, R_{cm} is the Ball radius in centimeters and $V_{\max}$ is the maximum admissible velocity of the ball (in cm/s), T is the camera frame rate and n is the number of frames between the past and actual ball detection. In this way the maximum probability value is related to the point where, on the basis of past information about the motion of the ball, the ball should be found (predicted point). The probability value decreases exponentially as the distance from the predicted point and becomes close to 0 for points far from the last known ball position that cannot be reached considering the upper speed limits (usually 120 km/h). In the following frames, the probability map is used to select candidate moving regions (as those with a probability greater than 0) on which the corresponding reference model is shifted to find the maximum correlation value associated to the ball position. In this way, the ball can be detected both in case of merging with players and in case of partial occlusions. The ball velocity, direction and probability map are always updated using the proper value for n (i.e., the number of frames between the actual frame and the last ball detection). If the ball is not detected for three consecutive seconds (i.e., n becomes greater than $T * 3$) the past information is considered outdated and the ball search procedure starts again considering all the candidate ball regions.

Shot events have to be accurately recognized in order to detect offside events. The ball cinematic in the image plane is evaluated. Every time the ball velocity and direction are updated they are also compared with the previous values: if the new values differ from the past one, i.e., velocity varies more than the 30% or direction varies more than 15°, the system assumes that the ball motion has been perturbed by some players. These values have been selected after long experimental observations that monitored the changes of ball velocity and direction. Each view recognize a shot when there is a variation either of the ball direction or of the ball velocity. In that case the nodes send a message to the main console containing all the information about the ball position, velocity and direction, as well as the position of all the players in the field of view.

III. Data Fusion and Decision Making

As illustrated beforehand each node acquires and evaluates independently from the others a portion of the whole soccer field. To evaluate offside conditions, the system has to be able to integrate and share information coming from all nodes to increase the capabilities of a correct interpretation of the whole scene. To achieve this, we need to network the nodes together. The developed system uses a master/slave approach. There is a special node, a supervisor, commanding the other nodes and receiving from them the essential information. In this way, every node, completely independent from the others while processing the images acquired with its camera (asynchronous process), attaches a time-stamp to each frame and sends the processed data to the central computer. The acquisition process is guided by a central trigger generator that guarantees synchronized acquisition between all the cameras. The supervisor has the task of synchronizing data by using the time-stamp field.

A. Synchronization

The processing time of each frame depends on the complexity of the scene. Even if the acquisition process is synchronized, the six nodes will take different amounts of time to process the corresponding frame. Then the data will be sent to the central supervisor in an asynchronous way. The synchronization phase has the aim of aligning the processed information and detecting critical situations when one or more nodes do not work properly. The supervisor must be able to take decisions although it works with incomplete data.

Synchronization is done using a queue. Each queue element is associated to a particular trigger pulse and is composed of six slots, one for each offside node. As soon as nodes send the processed data with a particular time-stamp (associated with a new pulse trigger), data are copied in the associated slot.

If one node sends the processed data corresponding to the frame t, the other slots corresponding to the other nodes of the same frame t can be in any of these three states.

1) UNKNOWN—nothing is known about the related node both at that time and after that time.
2) READY—information has arrived and is ready to be used.
3) MISSING—nothing is known about the related node at that time but there is evidence of information ready after that time. It is worth noting that since every node processes the information in order of time, MISSING information is supposed to be missing forever.

The supervisor continuously stores data and analyzes the content of the queue elements. It takes a decision if every slot of the queue element is classified either as READY or MISSING. If not all the nodes have transmitted their data for the frame t, but it is evident that the missing information is going to be persistently missing, there is no point in the supervisor delaying decisions any further. It takes a decision with the partial information available at that time. When some slots in the element queue are still in the UNKNOWN state the queue element cannot be processed because data could be produced later.

B. Data Validation

The processed data sent to the supervisor include various information, such as: the time-stamp (as a frame number), the identifier of the sending node (knows as RoleID), the number of identified moving objects, the ball description (found/not found, position in the taken image, speed, acceleration and direction angle on the image, detection of speed changes caused by players). Each moving object, detected as a player, includes: the size in pixels of the identified blob, the coordinates on the image plane of the base projection of the center of mass, the classification given by the node (team 1, team 2, goalkeeper team 1, goalkeeper team 2, neutral), the tracking code given by the node, the blob lifetime (in frames), and the stable classification lifetime.

The supervisor has to join all the data coming from the nodes and produce an unique representation in a common reference system. The reference system is a virtual playing field, whose dimensions are proportional to the real pitch, where players are mapped by using homographic transformations. The supervisor cross-correlates on this virtual field blobs and ball information coming from all the nodes. If the supervisor finds a mismatch it can decide to skip the supposed wrong data and keep the last one, or validate the data that seems more reliable. The supervisor evaluates the overall player configurations for offside detection only after the data validation.

C. Offside Detection

The offside detection module can be configured to signal immediately offside conditions or to judge about passive and active player behaviors. In the first and simplest case, each frame is judged separately from the others, and in general the frame information is checked extensively only if one of the nodes has detected that the ball trajectory has changed drastically. In this case it is necessary to determine which player has been responsible for this influence. In the other case, the most complex, the system must be able to judge on the active play. This requires an observation of a temporal window after the frame with the players in offside conditions.

1) *Determining the Player Who Struck the Ball:* This is one of the most complex task to execute. There are often several players near the ball and it can be very difficult to judge who has effectively touched it, even for a human observer. The system tries to solve this problem within the 3-D reconstruction obtained through homography. The six cameras are fixed on each side of the stands, then it is possible to do an initial calibration, observing a number of points in the field whose 3-D position in the real plane is known. In this way it is possible to evaluate the transformation matrix M that relates the points in the image plane with the corresponding points in the real plane. Players position can be always located on the playing field, because, if the player's feet are supposed to be always on the ground (condition violated substantially only during headers), their global coordinates can be always extracted, even with only one view (although the redundancy of data coming from opposite nodes help to keep under control measurement errors introduced by inaccurate blob segmentation). In order to evaluate the 3-D position of the ball it is necessary to have information coming from both the opposite views. The intersection of the two viewing lines provides the estimate of the ball position in the real world coordinate system. The Euclidean distance between the ball and each player is evaluated in the real world coordinate system and the player with the minimum distance is considered involved in the strike. When the supervisor cannot obtain the homographic position of the ball it works directly on the distances in the image plane between the player blobs and the ball.

2) *Determining Active Offside:* When the supervisor is instructed to decide whether the offside position of one player becomes active or remains passive for that shot, the supervisor evaluates each successive frame, effectively evaluating a temporal window 3 s long. Three seconds is the estimated maximum time required for a long pass to reach the designated player. During this temporal window the players identified

as being in possible offside, as well as the ball, are tracked by the supervisor to evaluate if their motions are compatible with an interception, or are not going to match anywhere. In particular the supervisor monitors a 3-D sphere around each player and evaluates if the ball enters in one of these spheres influencing the game. As in other contexts, the supervisor carries on its evaluation on the reconstructed 3-D scene, but if this is not possible, it tries to achieve the same thing directly on the data extracted from the images, without using homography. Passive offside events are visually signaled on the virtual field and listed for replay or debug purposes.

IV. Experimental Results

We applied the proposed method to several image sequences acquired during some matches of the Italian football championship (series A) 2006–2007. The system was installed in the Friuli Stadium. The imaging solution uses DALSA Pantera SA 2M30 cameras, achieving a resolution of 1920×1080 pixels (full HD) at 25 fps. The particular model used a single CCD operating using a Bayer filter. The acquired images are transferred to the six processors by fiber optic cables. Each node was equipped with two Xeon processors (Nocona series) running at 3.4 GHz with hyperthreading enabled. In this way, the operating system "sees" four processors and the software can be optimized accordingly. Each node featured 2 GB of RAM and used 8 SCSI disks with an individual size of 73 GB (configured in RAID0) to store the match, and another 120 GB SATA disk for the operating system and the software. The graphics sub-system was handled by a Geforce 7900 GT card with 256 MB. This accelerated graphics card, as well as visualization purposes, is also used to speed-up the system performing image processing operations through GPGPU techniques. Nodes were synchronized by using a Keithley KUSB-3102 trigger (with a maximum throughput of 100 kHz), connecting the supervisor, in charge of starting and stopping the trigger generator, to the cameras receiving the trigger signals.

The calibration of the six cameras to compute the transformation matrices for the homography evaluation was done just once, during the installation of the system, by using some reference points properly placed in the field. In this phase we evaluated also the errors introduced by the homographic transformation both to select the best set of reference points that produced low errors, and to estimate the errors in the measurements of the player position in the field. We evaluated that in each view the positioning errors can vary in the range 0–20 cm, according to the object position in the image. In general, when the points are in the central part of the image, errors are less than those obtained on the image border. Another possible source of error could be introduced by a poor positioning of blobs on players due to inaccuracies of the segmentation algorithm (the performance details of this step will be discussed later). Since we used high resolution cameras we evaluated that the pixel dimension is 2 cm in the near zone and 4 cm in the far zone. Errors are proportional to the number of pixels: a shift of n pixels in the blob position could introduce in the worst case an error in player projected position of $n \cdot 4$ cm. However, the effects of these errors in the final evaluation of offside events are reduced by the use of two opposite views to estimate the player and the ball position, because objects that are far in one view are near in the opposite view. When the supervisor receives information of the same entity from both the opposite views it considers as more reliable the one coming from the camera having the player in the near zone.

In order to evaluate the effectiveness of the system it was first necessary to estimate the performance of each step (Section IV-A), then to evaluate the system as a whole (Section IV-B). Quantitative analyses are possible only if the ground truth of all events is available. In this context, we do not have real positions of players and ball in the field. Another measurement system would be necessary on the players and inside the ball to estimate their positions in a global reference system. What is commonly done is a manual labeling of the image sequence in order to superimpose, on each player and on the ball, bounding boxes that are used as ground truth values to evaluate the algorithm performances. In the following section we will refer to ground truth as the data obtained during a manual labeling phase. We extracted a number of image sequences that were observed by a human operator who established, for each image, the player positions, the class he belongs to, the ball position, and the shots. To test the algorithms in different light conditions and on players with different shirts, eight test sequences were selected from different matches played with natural or artificial lights and between different teams. The sequences were 2 min long obtaining about 3000 frames for each view (all six views were considered). An interpolation program was devised to automatically assign the players and the ball positions between consecutive manual assessments. In fact the operator assigned, by using an *ad hoc* interface, the ground truth at every fifth frame, for a total of 600 frames for each sequence. The player ground truth was generated at each frame only for one view of one particular sequence, for testing the tracking step. This was necessary to verify in more depth the ID persistence for all the tracked entities.

All the steps (segmentation, tracking, classification, and ball detection) were implemented using Visual C++ and took about 0.066 s to process an 1920×1080 image, allowing the processing of 15 frames per second. For this reason the tests of the whole system during real matches were carried out processing every second frame. The system performances were evaluated during eight matches of the Italian football championship (Series A) 2006–2007. Offside events were detected with a latency depending only on the temporal window during which the system evaluated the active offside condition. So, in the worst cases offside events were detected after 3 s. Also, the quantitative evaluation of the whole system, presented in Section IV-B, was done during off-line tests on four selected matches. In this case the ground truth was built by a human operator who observed the selected matches and established the frames corresponding to offside events associating passive and active labels. In the off-line tests, all the acquired frames were processed.

TABLE I
RESULTS OF THE MOVING OBJECT SEGMENTATION ALGORITHM COMPARED WITH M.G. ALGORITHM ON EIGHT SEQUENCES 3000 FRAMES LONG

	Mean Errors and Variance (in pixels)					
	X Pos.	*Y* Pos.	*X* Var.	*Y* Var.	FP (%)	FN (%)
Prop. appr.	2.75	4.20	2.33	2.15	1.33	2.67
M.G. [27]	4.80	9.25	3.9	7.75	5.33	6.67

TABLE II
RESULTS OF THE PROPOSED CLASSIFICATION ALGORITHM COMPARED WITH A SUPERVISED APPROACH ON THREE SETS OF TEST IMAGES

	Correct Cassifications		
	Set (0, 15) (%)	Set (15, 30) (%)	Set (30, 45) (%)
Proposed Approach	97.26	96.43	96.66
Supervised Approach [17]	93.75	90.34	86.65

TABLE III
RESULTS OF THE TRACKING ALGORITHM COMPARED WITH A GRAPH MATCHING (G.M.) APPROACH FOR THE MAINTENANCE OF THE ID ON THE GOAL KEEPER AND ON SINGLE PLAYER TRACKING

	Goal Keeper (1 Track of 2883 Frames)		Players (45 Tracks of 15 226 Frames)	
	TF	TDR	TF	TDR
Prop. appr.	7	0.95	5.84	0.69
G.M. [26]	172	0.09	10.26	0.59

A. Experimental Results of Each Step

In this section, we report the results of the evaluation of each processing step. The proposed moving object segmentation algorithm was compared with a Mixture Gaussian approach [27] in terms of correct detection of player blob positions. In Table I, the results of the mean errors and variances are reported. The X–Y positions are the coordinates of the blob center projected on the blob base. The positions provided by the two algorithms were compared with the ground truth values and the mean errors were estimated. The proposed algorithm produced blobs that are better centered on the players than the Mixture Gaussian (M.G.) [27] especially for the X position that is fundamental for offside detection. Also in terms of false positives, i.e., the percentage of blobs not containing any moving object, our approach was more stable and assured a quick recovery from wrong segmentation due to autoiris problems. The number of false negatives, the percentage of players not segmented at all, depended on an oversegmentation of players that produced small separate blobs not considered as candidate players. Also in this case our approach was more robust producing a smaller percentage of False Negatives. Anyway, it should be considered that the most consistent segmentation errors could be filtered considering the position prediction obtained by the player kinematic evolution that could be assumed constant for a few frames given the high temporal resolution of the camera used.

Team discrimination is an important step for any semantic analysis of the game. In many related works, this step was solved considering supervised procedures that build the player models after the manual selection of candidate regions. The proposed unsupervised classification approach was able to build in the first seconds of the game (when the players enter into the field and arrange themselves in their initial game positions) the team models and to continuously update these models during the game to deal with the continuous lighting variations. Experiments demonstrated that in the worst case the six nodes took 20 s to generate the five class models. Then in a few tenths of second the supervisor was able to associate the team models between each pairs of opposite views. In order to evaluate the unsupervised classification approach and to assess its ability to update the model we considered a test set extracted from the first half of a match during which three different temporal ranges were selected: in the set $(0, 15)$ we collected the images extracted in the first 15 min of the match, in set $(15, 30)$ the images in the next 15 min, in the last set $(30, 45)$ the images of the last 15 min of the first half of the match. In Table II, we compare the classification results with a supervised approach that built the model class at the beginning of the match, proposed in a related work for automatic offside detection [17]. It is clearly evident that, nevertheless the autoiris procedure maintained the lighting conditions at nearly the same level, the team models were modified by a natural drift throughout the game. The proposed algorithm was able to maintain high percentages of detection not only after the prototype modeling phase, i.e., the first minutes of the match, but also in the successive phases when the natural drift modified the class models.

The ability of the tracking algorithm to maintain the same ID code when a player is tracked is fundamental for our system because we have to evaluate the active-passive condition. The player in offside position has to be monitored, for at least 3 s, to determine if he remains active for that shot. If, during this temporal window, the player ID is changed for any segmentation, classification or tracking problems, the active offside condition cannot be detected. In Table III, we compare the proposed approach with a graph matching algorithm presented in [26]. We evaluated only the ability of the two approaches to maintain the same ID on players. We used two metrics, introduced in [25], to compare the tracking approaches: the track fragmentation (TF), i.e., the number of result tracks matched to ground truth track (it is an indication of the number of ID, changes on the same track), and the track detection rate (TDR), i.e., the number of true positives for tracked object divided by the total number of ground truth points for objects (it provides an indication of the maximum length of each track segment with the same ID). In the ideal case of correct tracks both the metrics should assume the value 1. The tests were carried out on just one view sequence in which all the frames were manually labeled for the ground truth generation. In particular we considered one track of the goal-keeper (2883 frames) and on 45 tracks of players (for a total of 15226 frames). In the last case we reported the mean values of TF and TDR evaluated over the 45 considered tracks. The goal-

TABLE IV
RESULTS OF THE TRACKING AND SPLITTING ALGORITHMS ON THE GROUP BLOBS ON EIGHT SEQUENCES 3000 FRAMES LONG

	Tracking Algorithm (%)	Splitting Algorithm (%)
Two merged blob	91	95.7
Three merged blob	60.5	82.6
More merged blob	30	70.5

keeper remained for a long time in the same position, then the segmentation algorithms could fail by including the player in the background. The proposed tracking algorithm was able to recover these situations and maintain for a longer period the same label. Also for players our tracking algorithm provided good results both in terms of ID change, producing a TF of 5.8 against 10.26 provided by the graph matching approach, and in terms of TDR. The approach proposed in [26] was not able to solve the merge situations, while our approach, combining the results of the classification algorithm, provided a prediction of group situations and a splitting procedure to separate players. In Table IV, the results of the tracking algorithm and the splitting algorithm on group situations are reported. Tests on two, three and more merged blobs were executed. In nearly 91% of two merged blobs the tracking algorithm was able to recognize a merge, while for groups with three players the tracking performances were around 61%. The tracking performances decreased when the number of blobs in group situations became more than three. The splitting procedure was evaluated only on those blobs that were correctly predicted by the tracking procedure. In these group situations, the splitting procedure was considered correct if the classification algorithm centered the blobs on the players. In fact, in order to solve merge blobs the tracking procedure imposes the search of the expected classes. Then the results of the splitting algorithm can be evaluated only in terms of correct positioning of the blobs on the players. We considered a blob correctly centered if its distance (in the horizontal direction) from the player center was less than the half of the expected player width. Actually the percentage reported in these cases were related to the complexities of the observed merge situations: when the players were very close and the occlusions were almost total the probability of wrong classifications increased and the positioning of the blobs corresponding to the occluded players were quite random. However, the low percentages of blob tracking in merge situations reported in Table IV referred to the results of single cameras and did not necessarily affect the final offside decisions that were taken by the supervisor considering the results coming from opposite views.

We also evaluated the ball detection algorithm on the same sequences. In Table V, the results are reported. In this case, the ground truth established the number of frames in which the ball was visible from two opposite views, just from one view and not at all visible. When the ball was visible from both the opposite cameras, the ball detection algorithm was able in 80% of cases to detect the ball with two views, in 18% of cases to detect the ball with one view, and only in 1.6% of cases the algorithm was not able to see the ball. In about 82% of cases

TABLE V
RESULTS OF THE BALL DETECTION ALGORITHM ON EIGHT SEQUENCES 3000 FRAMES LONG

	Ground Truth		
	No Ball (%)	Ball in One View (%)	Ball in Two Views (%)
No ball	96.5	18	1.6
Ball in one view	3.5	82	18
Ball in two views	–	–	80.4

TABLE VI
RESULTS OF THE SHOT DETECTION ALGORITHM ON 235 SHOTS EXTRACTED FROM THE EIGHT TEST SEQUENCES

	Ground Truth	
	Shot	No Shot
Shot	210	48
No shot	25	–

in which the ball was visible from one camera, the algorithm detected the ball correctly. Few false positives were generated when the ball was not visible. Ball detection performance in Table V could be improved combining data coming from opposite views, for example introducing inference processes to overcome misdetections or occlusions and coupled tracking strategies to filter false detections. However, the impact of these modifications on the latency of the whole system and real improvements in offside events detection has to be accurately evaluated. The number of shots recognized by the algorithm is reported in Table VI; in the eight test sequences considered, 235 shots were manually labeled. The algorithm correctly detected 210 shots and 48 false positives, but only two of these were real false positives obtained on the players' shoes. The remaining were ball rebounds, where both the velocity and the direction of the ball changed. This kind of false positives was solved by the supervisor that recognized the rebounds since no player was close to the ball.

B. Experimental Results for the Offside Detection

The evaluation of an active offside condition is difficult not only for referees but also for people observing the matches and it often depends on subjective judgments. In order to implement this rule (the player in an offside position is involved in active play by interfering with play or an opponent, or gaining advantage by being in that position) we decided to evaluate the distance between the player and the ball at the end of the shot. However, it could happen that a player, involved in an active play, is not the nearest one to the ball. For this reason some errors in terms of false positive active offside events and false negative passive offside events can be detected by the system. In Fig. 3, the images of an active offside event are reported. The four images represent the positions of the players when the shot was recognized by the system. Three players with dark strips were in offside position. The positions of all the players in the virtual field (that is the homographic projections of the players and ball position), as generated by

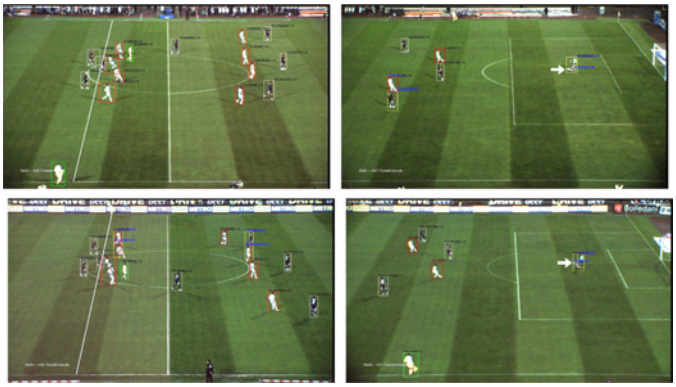


Fig. 3. Images of the field when a shot has been recognized. Three players with dark strips are in offside position.

TABLE VII
SYSTEM EVALUATION ON ''ACTIVE OFFSIDE,'' ''PASSIVE OFFSIDE,'' ''UNCERTAIN'' EVENTS

	Ground Truth			
	Act. Off. 45	Pas. Off. 600	Unc. Ev. 14	No Ev. 2335
Act. Off.	(77.8%) 35	(1.5%) 9	(7.1%) 1	(0.5%) 11
Pas. Offs.	(8.9%) 4	(91.7%) 550	(14.3%) 2	(5.6%) 130
Unc. Ev.	(8.9%) 4	(5.1%) 31	(14.3%) 2	(1.9%) 44
No Ev.	(4.4%) 2	(1.7%) 10	(64.3%) 9	(92.0%) 2150

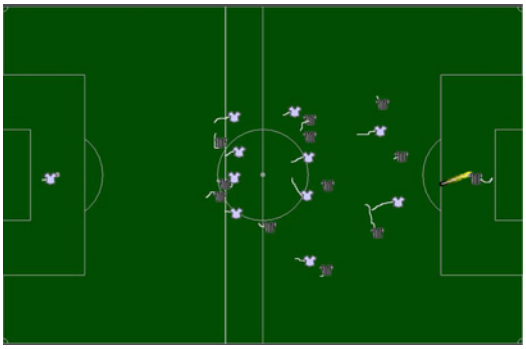


Fig. 4. Positions of players in the virtual field in the moment when a shot has been recognized.

Fig. 5. Images of the field when the ball reaches one of the players that was in offside position. In the image on the bottom-left the system detects that one of the players with dark strip receives the ball.

the central supervisor, are shown in Fig. 4. In order to establish if the action had to be penalized, the system has to follow the three players and detect if one of them received the ball. In Fig. 5, the four images, corresponding to the moment when the ball reached the players, are shown. This one was a difficult configuration to recognize since the two opponent players receiving the ball were close and each of them was visible from just one camera. The supervisor, taking into consideration the information coming from the two opposite cameras, detects the players receiving the ball. Only at this point the system can draw the white line of Fig. 3 centered on the player in an offside position. However, this situation was difficult for the referee. The assistant, who was looking at the goalkeeper, was not able to see in the same moment the situation of players 60 m from the ball (in the other half of the pitch). In fact the referee did not stop the game even though it was a real offside event.

In Table VII, the system results on the four matches are reported. The scatter matrix compares the ground truth, manually extracted from the off-line observation of the four matches, to the system responses. In the ground truth, active offside events, passive offside events, uncertain events were selected. The number of considered events is reported under each group. Few active offside events happened during a match. On the contrary, the number of passive offside events was very large because every time a player or a goalkeeper shot the ball toward the opposite goal, it was considered as a potential offside event even if the ball was directed to near teammates and not to those in offside position. The uncertain set contains all the events where it was difficult for the human operator observing the images to express a judgment (for the real difficulty of interpreting the rule, and also for the visibility conditions). In the set of no event, the last column of the table, there are all the shots (not in the offside direction considered as passive offside) on which the system could erroneously detect possible offside events (either active or passive or uncertain).

A good percentage of active offside events were correctly recognized by the system. The same can be said for passive offside events. The errors on false positive and false negative events were due to many concurrent factors: the ball could be occluded by players, the ball-player association could be wrong when many players were close, the classification of players could be incorrect because of occlusion problems, the player nearest to the ball was not interfering with play.

In order to better understand the system behavior and to evaluate its effectiveness in different situations, the active offside sequences were further divided in different subsets according to their difficulties in terms of distances between players (low, medium, and high difficulties) and complexity of the scene (simple and complex scenes). In particular, we considered in the set low difficulties the offside events in which the distance between the players (the defender and the forward) was greater than 1 m; in the medium set the offside with a distance between 0.5 m and 1 m; in the high difficult set all the events in which the players were very close and the distance was estimated less than 0.5 m. According to the complexity of the scene we labeled as Simple those events in which the ball detection was simple, i.e., the players

TABLE VIII
SYSTEM RESULTS ON ACTIVE OFFSIDE EVENTS DIVIDED ACCORDING THEIR DIFFERENT DIFFICULTIES AND COMPLEXITIES

			Active Offside		Passive Offside		Uncertain Offside		No Event	
DIST.	COMPL.	GT	One View	Two Views	One View	Two Views	One View	Two Views	One View	Two Views
Low	Simple	18	14	17	1	1				
Low	Complex	3	1	2			1	1		
Medium	Simple	9	7	7	1	1	1	1		
Medium	Complex	5	3	3	1	1			1	1
High	Simple	4	0	3					1	1
High	Complex	6	1	3		1	2	2		
Total		45	26	35	3	4	4	4	2	2

Comparisons between one view and two views.

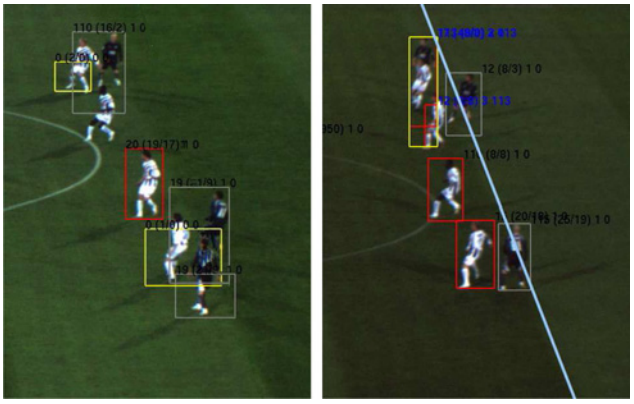


Fig. 6. Images of two views during an offside event. In the image on the left it was not possible to split the players so the offside was not identified with 1 View. The opposite image on the right was able to separate the blobs, so the offside detection procedure could be started.

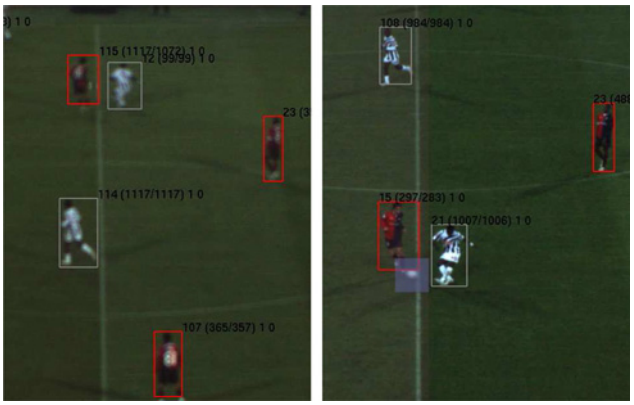


Fig. 7. Images of two views during an offside event. In the image on the left it was not possible to detect the ball and the shot so the offside was not identified with 1 View. The opposite image on the right was able to detect the ball, so the offside detection procedure could be started.

were far generating a single blob, and the player tracking was simple for the active/passive offside detection; while we considered complex all the situations in which there were at least two players in the group blob, the ball could be occluded and difficult to detect, the players tracking was complex for crowded group situations. In Table VIII, we analyzed in more details the results obtained on these active offside events. In the first column, the sequences ground truth divided into the different considered sets is reported (referred to as GT). The remaining columns report the corresponding

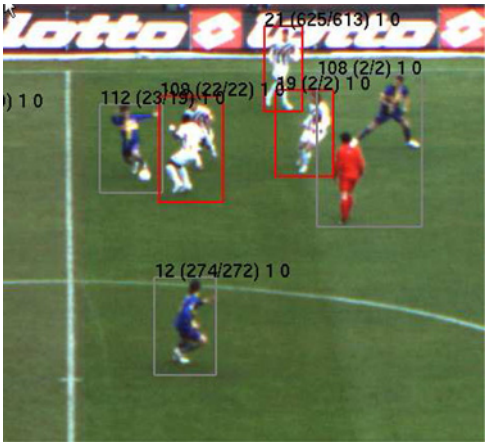


Fig. 8. Image where the ball-player association is not possible only with distance evaluations.

system evaluations in terms of active, passive, uncertain, and no events. In each of these cases we compared the results obtained by using one view (i.e., the three cameras on one side of the field), with the results obtained with two views (i.e., the six cameras on the opposite sides of the field).

It is clearly visible that the system with two views detected more events than the system with 1 view (35 against 26). The integration of information coming from opposite cameras allowed the solution of many ball and player occlusion problems. In Figs. 6 and 7, two cases are reported in which 1 view was not able to separate players and to detect the shot while the opposite view, having a different perspective, was able to start the offside detection procedure. In particular, the images of Fig. 6 were extracted from one of the sequence High Difficulties and Complex correctly detected by the two views, reported in the last row of Table VIII.

From the analysis of the system results it is clear that the difficulties were essentially due to complex situations in which the tracking step was not able to solve the group situations, and the supervisor could not take a decision producing four uncertain events, or there were wrong ball-player association, producing four passive offside events. The failure of the ball detection step was the cause of one of the events not detected at all by the system. This case was in reality difficult to solve since the ball was occluded in both the two opposite views. Future works will be focused on the analysis of the ball trajectory deviations to recover the exact frame in which

the ball was shot. The second undetected offside event was due to the errors in the player projections. The distance between opponent and defender players involved in the action was very short and then, the error either in homographic projection or in player segmentation produced inaccurate positioning in the virtual field. In Fig. 8, a complex situation, that gave rise to a false offside detection (one of the no event of Table VII detected by our system as active offside), is shown. Three players were very close to the ball, so the system failed to infer the correct ball-player association. In these cases further posture analysis is required to recognize the player throwing the ball, but we would need higher resolution images to give greater details of the players.

It should be taken in account that Tables VII and VIII report the average performance of the system on test sequences. Anyway, system performance can slightly vary from match to match, depending on lighting conditions, complexity of the offside event situations and the strips the players are wearing.

V. Discussion and Conclusion

The aim of this paper was to evaluate the feasibility of a real-time visual system for offside detection that could respect the main constraints imposed by the FIFA for the usage of technologies during official matches: first of all not to be invasive of the field and the players, to be independent of human intervention, and to have real-time responses. In this paper, we present the results obtained from the experiments carried out during the championship 2006–2007, with six cameras placed on the two sides of the field. Six computers processed the images acquired by the synchronized cameras, detecting the ball and the players position in real-time. The processing results were sent to a central supervisor, which evaluated the offside event probability and took a decision of active offside, passive offside, uncertain, no event. The system was tested during four official soccer matches for a total of 360 min of processing, during which the system performances were assessed.

In our opinion the proposed system could provide an effective aid to referees in many situations: experiments demonstrated that the use of multiple cameras with a high frame rate allowed the reduction of perspective errors and the detection of quick events that happened in distant parts of the field. The opposite cameras solved many cases in which the decision was difficult by using only one view. The tests showed that the system was able to recognize 35 offside events (over a total of 45). Four wrong passive offside or four uncertain events were generated by the system caused by the difficulty in interpreting the offside rule, the extreme imaging conditions that might happen in crowded situations, the complex configurations among players, the difficult interactions between the ball and the players. Anyway, in these cases, the system, having identified possible offside situations for the player positions, could send a warning signal to the referee committee, who is free to decide to accept or reject the suggestion. Further work is necessary to improve the performances in complex situations in which the passive and uncertain events are erroneously provided by the system. The proposed system suffers when it has to decide in crowded situations: in these cases either the use of more cameras in different positions of the field or more complex feature analysis could be introduced to separate players. Besides we simplified the active-passive offside rule considering active only the player who receives the ball. More complex tactical analysis would be required if we would interpret the rule in all its cases. In the same way, player posture analysis could help the system in the most extreme offside situations. Another possible source of errors is the imprecise localization of the exact frame in which the shots are recognized. A delay of few frames is the cause of false positives since the players move very fast and their configurations in the field change radically in few tenths of seconds. Ball trajectory analysis and player interaction analysis can be introduced to provide greater precision in the shot temporal identification.

The proposed system could be used in three different ways. The first one is to provide an automatic system for offside detection. We are confident that the performances in crowded difficult situations could be solved with further work by means of the introduction of more specialized hardware, more cameras with higher resolution placed on different points of view, and more complex image processing algorithms. The second way is a semi-automatic system to support the referee committee during the game. In this case, for example, the fourth assistant, by using our highly developed interface, can monitor system responses, verifying them, and then can notify the referee of his personal judgment. The third way is to use the system offline for referee training purposes. In fact, the system provides also the positions of referees and assistants and then it is possible to verify their alignments with the players during the game. In the last two cases, at least for the referees, the system, in its current version, could be used right now.

We firmly believe that artificial visual systems should be introduced in soccer matches, as other sports already have, to solve ambiguous situations. Technology is already available and its cost is comparable to broadcasting solutions. The research in the image processing and pattern recognition fields have obtained promising results and can be applied to the soccer context. The worldwide interests in sporting applications will certainly led to further investigations of the scientific community in these directions.

References

[1] C. L. Huang, H.-C. Shih, and C. Y. Chao, "Semantic analysis of soccer video using dynamic bayesin network," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 749–760, Aug. 2006.

[2] N. Inamoto and H. Saito, "Virtual viewpoint replay for a soccer match by view interpolation from multiple camerasces," *IEEE Trans. Multimedia*, vol. 9, no. 6, pp. 1155–1166, Oct. 2007.

[3] P. J. Figueroa, N. Leite, and R. Barros, "Background recovering in outdoor image sequences: An example of soccer players segmentation," *Image Vis. Comput.*, vol. 24, pp. 363–374, 2006.

[4] J. Renno, J. Orwell, and D. Thirde, "Shadow classification and evaluation for soccer player detection," in *Proc. Br. Mach. Vis. Conf. (BMVA)*, Kingston Upon Thames, U.K., Sep. 7–9, 2004, pp. 839–848.

[5] J. Ren, J. Orwell, G. Jones, and M. Xu, "Real time modeling if 3-d soccer ball trajectories from multiple fixed camera," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 350–362, Mar. 2008.

[6] J. B. Hayet, T. Mathes, J. Czyz, J. Piater, J. Verly, and B. Macq, "A modular multicamera framework for team sports tracking," in *Proc. IEEE Conf. Adv. Video Signal Based Surveill. (AVSS)*, Sep. 15–16, 2005, pp. 493–498.

[7] M. Kristan, J. Pers, M. Perse, S. Kovacic, and M. Bon "Multiple interacting targets tracking with application to team sports," in *Proc. 4th Int. Symp. Image Signal Process. Anal.*, Sep. 2005, pp. 322–327.

[8] P. J. Figueroa, N. Leite, and R. Barros, "Tracking soccer players aiming their kinematic motion analysis," *Comput. Vis. Image Understand.*, vol. 101, pp. 122–135, 2006.

[9] W. Du, J. B. Hayet, J. Piater, and J. Verly, "Collaborative multicamera tracking of athletes in team sports (CVBASE 2006)," in *Proc. Workshop Comput. Vis. Based Anal. Sport Environ.*, Graz, Austria, pp. 2–13.

[10] M. Beetz, N. Hoyningen-Huene, J. Bandouch, B. Kirchlechner, S. Gedikli, and A. Maldonado, "Camera-based observation of football games for analyzing multiagent activities," in *Proc. 5th In. Joint Conf. Auton. Agents Multiagent Syst. (AAMAS)*, 2006, pp. 42–46.

[11] N. Vandenbroucke, L. Macaire, and J. G. Postaire, "Color image segmentation by pixel classification in an adapted hybrid color space. Application to soccer image analysis," *Comput. Vis. Image Understand.*, vol. 90, pp. 190–216, 2003.

[12] M. Xu, J. Orwell, L. Lowey, and D. Thirde "Architecture and algorithms for tracking football players with multiple cameras," *IEEE Proc. Vis., Image, Signal Process.*, vol. 152, no. 2, pp. 232–241, Apr. 2005.

[13] X. Yu, T. S. Hay, X. Yan, and E. Chng, "A player possession acquisition system for broadcast soccer video," in *Proc. IEEE Int. Conf. Multimedia EXPO.*, Amsterdam, The Netherlands, Jul. 6–8, 2005, pp. 522–525.

[14] Y. Liu, D. Liang, Q. Huang, and W. Gao, "Extracting 3-D information from broadcast soccer video," *Image Vis. Comuput.*, vol. 24, pp. 1146–1162, 2006.

[15] T. D'Orazio, C. Guaragnella, M. Leo, and A. Distante, "A new algorithm for ball recognition using circle Hough transform and neural classifier," *Pattern Recognit.* vol. 37, pp. 393–408, 2004.

[16] X. Yu, H. W. Leong, C. Xu, and Q. Tian, "Trajectory-based ball detection and tracking in broadcast soccer video," *IEEE Trans. Mutlimedia*, vol. 8, no. 6, pp. 794–805, Dec. 2006.

[17] S. Hashimoto and S. Ozawa, "A system for automatic judgement of offsides in soccer games," in *Proc. IEEE Int. Conf. Mutlimedia EXPO 2006 (ICME 2006)*, pp. 1889–1892.

[18] K. Choi and Y. Seo "Tracking soccer ball in TV broadcast video," in *Proc. Image Anal. Process. (ICIAP)*, pp. 661–668, 2005.

[19] T. Shimawaki, T. Sakiyama, J. Miura, and Y. Shirai "Estimation of ball route under overlapping with players and lines in soccer video image sequence," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, 2006, pp. 359–362.

[20] FIFA. *Laws of the Game 2007/2008*. Available: http://www.fifa.com

[21] T. Kanade, T. Collins, and A. Lipton, "Advances in Cooperative Multi-Sensor Video Surveillance," *Darpa Image Understanding Workshop*, San Mateo, CA: Morgan Kaufmann, Nov. 1998, pp. 3–24.

[22] F. B. Maruenda, "Can the human eye detect an offside position during a football match," *Br. Med. J.*, vol. 329, no. 7480, pp. 1470–1472, Dec. 2004.

[23] S. Theodoridis and K. Koutroumbas, "Clustering algorithms I: Sequential algorithms," in *Pattern Recognition*. New York: Academic Press, 2006, ch. 12, pp. 517–540.

[24] PCT/IT02/00039 Regstered by CNR-ISSIA. "System and method for the measurement of the relative position of an object with respect to a point of reference," Int. Pub. WO 07/061684 A2.

[25] T. Black, T. Ellis, and P. Rosin "A novel method for video tracking performance evaluation," in *Proc. Joint IEEE Int. Workshop Vis. Surveil. Perform. Eval. Track. Surveil. (VS-PETS)*, 2003, pp. 125–132.

[26] M. Taj, E. Maggio, and A. Cavallaro, "Multi-feature graph-based object tracking," in *Proc. CLEAR Workshop (LNCS 4122)*, Southampton, U.K.: Springer, Apr. 2006, pp. 190–199.

[27] Z. Zivkovic and F. van der Heijden, "Recursive unsupervised learning of finite mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 5, pp. 651–656, May 2004.

Tiziana D'Orazio (M'08) received the computer science degree from the University of Bari, Bari, Italy, in 1988.

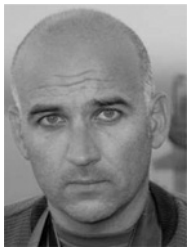
Since 1997, she has been a Researcher with the Institute of Intelligent Systems for Automation, Italian National Research Council, Bari, Italy. Her current research interests include pattern recognition, video analysis, and computer vision for video surveillance, domotics, intelligent transportation systems, and quality control. She is the author of more than 100 technical papers and book chapters in refereed conferences and journals in the areas of robotics and computer vision.

Marco Leo received the the engineering degree in computer science from the University of Lecce, Lecce, Italy, in 2001.

Since 2001, he has been a Researcher with the Institute of Intelligent Systems for Automation, Italian National Research Council, Bari, Italy. His research interests include image processing, computer vision, pattern recognition, digital signal processing, feature extraction, and neural networks. He is the author of more than 100 papers in national and international journals, and conference proceedings.

Paolo Spagnolo received the engineering degree in computer science from the University of Lecce, Lecce, Italy, in 2002.

Since 2002, he has been a Researcher with the Institute of Intelligent Systems for Automation, Italian National Research Council, Bari, Italy. His research interests include computer vision, image processing and analysis, pattern recognition, and classification. He is the author of several papers in international journals and coauthor of three international patents.



Pier Luigi Mazzeo received the engineering degree in computer science from the University of Lecce, Lecce, Italy, in 2001.

Since 2001, he has been with the Institute of Intelligent Systems for Automation, Italian National Research Council, Bari, Italy. He participated in a number of national projects on real-time defect detection systems for railway maintenance using image processing. His research interests include signal processing, image processing, and image understanding.

Nicola Mosca received the computer science degree from the University of Bari, Bari, Italy, in 2004.

Since 2004, he has been a Contract Researcher with the Institute of Intelligent Systems for Automation, Italian National Research Council, Bari, Italy. His research interests include human computer interaction, 3-D visualization of sport events, video analysis, and computer vision for video surveillance and intelligent transportation systems. He is the author of several technical papers and coauthor of three international patents on visual systems for event detection in sport contexts.

Massimiliano Nitti received the degree in computer science from the University of Bari, Bari, Italy, in 1996.

Since 2000, he has been a Researcher with the Institute of Intelligent Systems for Automation, Italian National Research Council, Bari, Italy. His interests include prototype developing, real-time algorithms, image acquisition machines, classification machines, multithread processing using multicore machines, SSE2 algorithm implementation.

Arcangelo Distante received the computer science degree from the University of Bari, Bari, Italy, in 1976.

He is the Director of the Institute of Intelligent Systems for Automation, and the Coordinator of the Robot Vision Group, Italian National Research Council, Bari, Italy. His research interests include computer vision, pattern recognition, machine learning, neural computation, robot navigation and computational models for the real-time processing of space/temporal image sequences. He is the coauthor of more than 350 scientific papers.