A Visual Framework for Event Understanding in Soccer Matches

M. Leo, N. Mosca, P. Spagnolo, P. L. Mazzeo, T. D’Orazio, A. Distante
Institute of Intelligent Systems for Automation - C.N.R.
Via Amendola 122/D-I, 70126 Bari, ITALY
E-mail: leo@ba.issia.cnr.it

Abstract

In the last decade, several research efforts have been undertaken in soccer video analysis. This increasing interest is motivated by the possible applications over a wide spectrum of topics: indexing, summarization, video enhancement, team and players statistics, tactic analysis, referee support, etc. Soccer video analysis requires different challenging tasks: ball and players have to be localized in each frame, tracked over time and, above all, their interactions have to be detected and analyzed. The latter task is fundamental, especially for statistics and referee decision support purposes, but, unfortunately, it has not received adequate attention from the scientific community. In this paper a multi-view system able to understand in real time the interactions between the ball and the players is presented. 3D ball trajectories are extracted by triangulation from multiple cameras and used to detect the interactions between the players and the ball. Inference processes are then introduced to determine the player kicking the ball and to fix the instant of the interaction. The system has been tested during several matches of the Italian first division football championship and experimental proofs of its effectiveness are reported.

I. INTRODUCTION

Sport Video analysis is becoming one of the most important challenges in computer vision due to the wide spectrum of possible applications such as referee support, tactics analysis, automatic highlights identification, video annotation and browsing, content based video compression, automatic summarization of play, customized advertisement insertion, graphical object overlaying for better enjoyment of events, players and team statistics, etc. Due to the growing popularity, most technologies available for sport video analysis were focused on soccer, although a rather short list of works concerning analysis of other sports as American Football [1], basketball [13], baseball [8], cricket [17], tennis [7][18] and volleyball [9] could be found in literature. Regarding soccer video analysis, two different research areas could be identified in literature:

1) low level processing to segment and temporal track the ball and the players;
2) high level processing to understand semantic events as interactions between ball and players.

In this paper we want to focus on the problem of the exact localization of the interactions between ball and players both in terms of position in the field and in terms of temporal localization of exact frame in which the interactions
A. Related Works

In the recent decades many papers faced the problems of segmentation and tracking of players and ball using cameras suitably placed around the play-field. In [11] players and ball are firstly segmented by background subtraction, then shadows are removed introducing some color constraints and finally player correspondences are found using homography. Tracking of players is instead performed in [6] 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. This approach was, afterwards, applied in [2] to analyze the distances covered by Brazilian soccer players. An approach to tracking players in multiple cameras is also proposed in [21]. Players are tracked not only in the image plane of each camera but also in the ground plane by individual particle filters. Ball detection and tracking problem was faced in [5] using Circular Hugh Transform and neural networks and in [16] using motion information.

Some works also deal with segmentation and tracking of players and ball directly in broadcast videos. For example in [10] a scheme to detect and locate the players and the ball on the grass play-field in soccer videos is proposed. Authors put forward a shape analysis based approach to identify the players and the ball from the roughly extracted foreground, which is obtained by a trained, color histogram-based play-field detector and connected component analysis. Euclidean distance transform is used to extract skeletons for every foreground blob, and then shape analysis is performed to remove false alarms. In [23] ball trajectories are computed using motion analysis: firstly, a set of ball candidates for each frame is selected and then, among them, the ball is detected as the "most active" object in the scene. In [22] authors present a method to estimate players’ and ball’s positions from monocular broadcast soccer video. With the relationship between objects and the camera in perspective projection, they derive the formula for estimating the moving objects’ positions in real world, even when the ball is in the air. A visual tracking system that determines the coordinates and trajectories of football players in camera view based on TV broadcasts is introduced in [3]. The system solves a complex probabilistic estimation problem that consists of three subproblems that interact in subtle ways: the estimation of the camera direction and zoom factor, the tracking and smoothing of player routes, and the disambiguation of tracked players after occlusions.

Few works deal instead with the understanding of semantic events in soccer as, for example, the interactions between ball and players although their detection, analysis and time-space localization are, undoubtedly, fundamental tasks especially for referee support applications: for example the accurate detection of the time instant where ball passing or shooting occurs and, eventually, the identification of the players kicking the ball are two pivotal steps towards the automatic detection of soccer events as offside. Two pioneering papers in this research field could be considered [4] and [24]. In [4] a system able to recognize, from broadcast videos, three different kinds of actions (keeping control over the ball, passing and shooting) is presented. Actions recognition is performed by estimating camera parameters and then reconstructing and comparing 3D positions of both ball and players. Fine localization in time of recognized events is not addressed and furthermore, as the same authors assert, processing
is time expensive, object localizations is not robust and action recognition is very inaccurate. In [24] broadcast video segments corresponding to the interactions among players and ball, as "ball-passing" and "ball-dribbling", are automatically extracted using similarity metric between the shape of trajectories of players and ball.

The main aim of both aforementioned papers is to conduct tactic pattern analysis: the identification of play regions and their following analysis for action classification analyzing multi-agent activities are their outcomes. As a consequence, they did not face at all the challenging problems of exact identification of player kicking and fine localization in time of detected interactions that, as said before, are fundamental steps in some interesting applications, especially in those aiming real time referee support.

B. The problem

The problem of detecting the exact localizations of ball and players in the field is fundamental for developing automatic systems for high level processing and interpretation of complex actions. The perspective conditions due to the camera position introduce parallax errors that do not allow the precise identification of the players position in the field by using broadcast images. The Cameras used by all the Sport broadcast Channel, are placed just on one side of the pitch and they do not assure the simultaneous and the complete covering of the field with their images. Besides, since generally moving cameras are used, the calibration problem has to be continuously solved in order to project the positions of the players in the field. For these reasons it is necessary to use multiple stationary cameras, properly placed in the field in order to minimize the parallax errors, and that are dedicated to the task of acquiring the images of all the field. Unless a large number of cameras is used, it is likely that the cameras have a large camera view. In this case, the problem of recognizing the ball and detect the interactions with players becomes much more complex. Like for humans, it is quite simple to recognize the ball when it moves far from players; the task is much more difficult when the ball is next to players or partially occluded by them. The well known approaches based on pattern recognition algorithms cannot be used since the ball diameter in the images is very small and above all the ball texture is not visible. Approaches based on ball trajectory evaluation, even if they are able to recognize the ball among a set of ball candidates, fail when the ball is motionless or attached to players. Besides, trajectory based approaches do not solve the main problem of detecting the exact frame in which the shot starts and the recognition of the player involved in the interaction.

C. Our Contribution

The main objective of our work is to propose a new visual framework able to recognize interactions between the ball and the player, to finely localize them in time and to point out the player involved.

In figure 1 the proposed visual system is outlined. Six high resolution cameras have been placed on the two sides of the field assuring double coverage of all the areas by either adjacent or opposite cameras. The acquired images are transferred to six processing nodes by fiber optic cables. The acquisition process is guided by a central trigger generator that guarantees synchronized acquisition between all the cameras.
Each node, using two hyper-threading processors, 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 processing nodes, are connected to a central node, which has the supervisor function. It synchronizes data coming from nodes and performs high level processing.

The main contribution of this work is the multi-view integration of the data coming from the processing of each view in a 3D reference system. The 3D trajectories of the ball and the 2D trajectories of players are estimated. High level inference procedures are introduced to evaluate the intersection and proximity of these trajectories in order to detect the player who shoots the ball and above all to estimate the exact frame in which the interactions happen.

Figure 2 shows the steps executed by each node for the processing of each view and by the supervisor for the multi-view integration.

Each node, first of all, extracts moving areas using proper background subtraction and updating procedures. Extracted objects are then processed simultaneously by a player detection algorithm and a ball detection algorithm. An unsupervised clustering algorithm, based on feature color analysis, is then used for player classification, i.e. to separate players belonging to different teams and referees. Finally, a real time procedure for player tracking follows the players solving the problems of occlusions and blob splitting. At the same time a ball detection procedure recognizes the ball and tracks it on the image plane.

All data are collected and synchronized by the central supervisor executing data fusion and performing:

1) localization of both ball and players on a virtual play-field;
2) computation of 3D ball trajectories and 2D players trajectories;
3) detection of interactions between ball and players (revealing possible shoots on goal or passing events);
4) localization of detected interaction events both in time and space;
5) selection of the players interacting with the ball.

The rest of the paper is organized as follow: section II describes the algorithms that process each view; section III details the computational framework performing the multi-view integration for detecting the interactions between ball and players. Finally section IV reports experimental results whereas final discussion and conclusions can be found in section V.

II. SINGLE VIEW PROCESSING

Each node uses a background subtraction algorithm for motion detection. It is based on a modified version of the well known approach [12] for background creation and maintenance: a pixel \((x, y)\) is considered as a moving one if it differs from the background model \(B(x, y)\) more than twice the standard deviation \(V(x, y)\). In order to build a background model without being affected by moving foreground objects (moving players are always present in the scene) the energy information of each image point, evaluated in a small sliding temporal window, to distinguish static points from moving ones, is used. In this way a statistical background model with only the contribution of static points, without the effects of foreground objects, is obtained [15]. This step is crucial in the soccer context...
Fig. 1. The position of the cameras in the field

Fig. 2. The scheme of the visual system

because the constraint of a static background for modeling the scene is not applicable before the beginning of the match. Figure 3 shows an image acquired at time instant \( t \) and the relative segmentation outcome obtained by the implemented algorithms.

Information relative to moving objects are then sent to two parallel processing threads: the first one performs
human blobs detection, classification and tracking; the second one performs ball detection. Human blobs are detected on the basis of topological hypothesis (area and width/height ratio) and each of them is assigned to one of five possible classes: Team I, Team II, goal keeper team I, goal keeper team II, referee. 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 [20], an unsupervised approach substantially independent from human interaction. Then, at runtime, each segmented object is assigned to one of the classes previously extracted. Further details could be found in [14].

After segmentation and classification, each node makes use of a probabilistic procedure to temporally track each player. Events as player merging, splitting, entering and leaving the scene are predicted and properly managed by each node [19].

Moving blobs that do not accomplish human detection hypothesis are the input of the ball recognition process described in the next subsection.

A. Ball Recognition

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 the candidate moving regions 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 a region separate 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 an area not compatible with the ball size are discarded whereas the remaining moving regions are labeled as candidate ball region and provided to a pattern recognition procedure. In [5] the ball recognition task was performed by using a neural network after preprocessing based on a wavelet transform. In this work the ball diameter in the image has large variations according to the ball position in the field and the ball texture information is not always appreciable. Neural network approaches are ineffective in
these situations. For this reason a correlation procedure has been preferred to evaluate shape and texture similarity between the candidate ball regions and a comparative set of manually selected reference examples of ball.

The correlation measure

\[
\rho = \frac{\sum_{i=1}^{X} \sum_{j=1}^{Y} ((I_{i,j} - m) \cdot (M_{i,j} - m))}{\sqrt{\sum_{i=1}^{X} \sum_{j=1}^{Y} (I_{i,j} - m)^2 \cdot \sum_{i=1}^{X} \sum_{j=1}^{Y} (M_{i,j} - m)^2}}
\]

between the candidate region \(I\) and a reference model \(M\) is computed where \(X\) and \(Y\) are respectively the image width and height and \(m\) is the mean value of the reference model \(M\).

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 under 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 diameter, 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. In figure 4 the effects of different lighting conditions (first raw) and distance from the camera (second raw) on the ball appearance are pointed out. Indeed, the first raw reports a ball image acquired under artificial lighting conditions (on the left) and a ball image acquired in a sunny day (on the right) whereas the second raw reports the ball appearance when the ball is far (on the left) and near (on the right) to the camera.

![Fig. 4. Different appearances of the ball under different lighting conditions (first row) and distance from the camera (second row).](image)

The correlation measure is evaluated between a patch centered on the candidate region and a reference model for selected comparative set of ball examples. The reference model is the mean images of all the examples in a set. In this way, only one correlation measure has to be performed for each comparative set. 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 values then it is selected as the one with the maximum correlation coefficient.

For example figure 5 shows two regions of the same image providing high correlation values. The one on the left is the foot of one player, while on the right there is the ball. The system correctly selected as ball the one on the right because its correlation value is the greatest one.

Fig. 5. Two regions of the same image that produced high correlation values

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 $\theta$ of the ball in the image plane are computed as follow:

$$V = \sqrt{V_x^2 + V_y^2} \quad \theta = \arctan\left(\frac{V_y}{V_x}\right)$$  \hspace{1cm} (1)

where

$$V_x = \frac{(P_{xt} - P_{xt-1})}{n} \quad V_y = \frac{(P_{yt} - P_{yt-1})}{n}$$ \hspace{1cm} (2)

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

Then a ball position probability map, covering all the point of the processing image, is build as follow:

$$P(x, y) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left(-\frac{\left[(x - \bar{x} + V_x \text{sign}(\cos \theta)) + (y - \bar{y} + V_y \text{sign}(\sin \theta))\right]^2}{2 \sigma^2}\right)$$ \hspace{1cm} (3)

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

$$\sigma = \frac{R_p V_{\text{max}} n}{R_{cm} T}$$ \hspace{1cm} (4)

where $R_p$ is the Ball radius in pixels, $R_{cm}$ is the Ball radius in centimeters and $V_{\text{max}}$ is the maximum admissible velocity of the ball (in cm/sec), $T$ is the camera frame rate and $n$ is the number of frame between the past and actual ball detection (1 if, in this case, the two frames are consecutive). 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 going away 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 where
a patch of the same size of the reference model is slid 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 grater than $T*3$) the past information is considered outdated and the ball search procedure starts again considering all the candidate ball regions.

III. Multi-View Integration

As illustrated above, each node acquires and evaluates, independently from the others, a portion of the whole soccer field. Even if the acquisition process is synchronized, the six nodes could take different length of time to process the corresponding frame. At each time instant $t$ the node $i$ sends, in an asynchronous way, to the central processing unit named “supervisor”, the information $F_{t}^{i}$ obtained by processing the acquired frame. The supervisor synchronizes incoming information using a queue: each queue element is associated to a particular trigger pulse and is composed of six slots, one for each processing 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. The supervisor continuously stores data and analyzes the content of the queue elements. When either the corresponding slots are ready to be analyzed (all six nodes sent relative information) or it is possible to infer that relative data are missed (for example data related to successive acquisition pulses are already available) the supervisor starts to analyze the information available in order to perform the computational steps listed in section I-C and detailed in the following subsections.

A. Ball and Players Localization in a Virtual Play-Field

The supervisor uses a virtual play-field (having the same dimension of the real play field) where all the information are projected. Information relative to the players and the referees are projected onto the virtual play-field by homographic transformation assuming that the feet of the player/referee are always in contact with the ground. An initial calibration phase was carried out observing a number of reference points in the field whose 2D positions in the play-field were known. The reference point coordinates in the image planes of the six cameras and their known positions into the play-fields allowed to build a 3x3 transformation matrix $M$ for each camera so that each point of the image planes can be projected on to the play-field and viceversa. One of the images used for the calibration is shown in figure 6. The matrix $M$ can be determined using only 4 reference points, but in order to reduce calibration errors more reference points were used: different sets of reference points were alternatively used for each camera in order to obtain more accuracy in the transformation matrix. The accuracy of each set of reference points was verified using the remaining points for testing. At the end of these evaluations, the best set was chosen. Anyway, the projection of the same player using data relative to different cameras could be not coincident due to the possible different segmentations into the image planes caused by different appearances of the same player (different position with respect the camera, different lighting conditions, shadows and so on). This drawback has
been overtaken considering, for further processing, as unique representation of the player in the virtual play-field the mid-point of the line connecting the different projections.

The projection of the ball position into the virtual play-field requires a different procedure. The constraint of having the ball on the play-field (as for players) is not valid. The 3D ball position has to be firstly recovered by triangulation (if ball information coming from two opposite or adjacent views is available) and then its projection onto the virtual play field can be performed. Assuming that the ball is observed from two cameras $c_1$ and $c_2$, we obtain the corresponding projections $b_1$ and $b_2$ on the ground plane using homography. Let $l_1$ and $l_2$ be the two lines from $c_1$ to $b_1$ and $c_2$ to $b_2$, respectively. Theoretically, the intersection point between $l_1$ and $l_2$ should correspond to the 3D ball position but, in practice $l_1$ and $l_2$ do not intersect due to errors caused by camera calibration and object detection. Thus, the 3D ball position, can be estimated by assuming the camera error of both cameras of the same order, and fixing the ball position $P(x, y, z)$ as the mid-point of the minimum distance segment between $l_1$ and $l_2$. The projection of $P(x, y, z)$ in the virtual play-field is finally obtained setting to 0 the z-component i.e. $P(x, y, 0)$. In figure 7 the virtual play-field is reported. The red and cyan rectangles indicate players positions as computed relating data coming from two opposite or adjacent views (relative IDs assigned from nodes are also reported) whereas the ball position is indicated by the yellow cross. The white lines behind each object indicate recent ball and players displacements.

**B. Ball and Players Trajectory computation**

Starting from the estimated ball and players positions their temporal trajectories can be computed.

The player trajectories in the virtual play-field cannot be mathematically modeled by straight lines or curves; they vary continuously in an unpredictable way and then they can be represented only collecting the player positions into the play-field.

For the ball, instead, trajectories in the virtual play-field can be approximated by straight lines: this allows the system to predict the successive positions, to recover missed intermediate ones, and to introduce high level reasonings useful to understand the soccer game developing. For the sake of precision we have to explain that we...
dispose of both 3D and 2D ball trajectories. In order to detect shots, as abrupt changes of trajectories, we consider in this paragraph only 2D trajectories, obtained by projecting the 3D ball positions on the virtual play-field. This simplification allows to avoid false shot detections when there are ball rebounds on the field. In particular the ball trajectories $T^i$ are data structures consisting of the following items updated each time a new ball point is available:

1) lifetime $T^i_l$;
2) angular coefficient $T^i_\theta$;
3) acceleration $T^i_a$;
4) velocity $T^i_v$;
5) ball points $T^i_p$, belonging to the trajectory $T^i$;
6) acquisition pulses $T^i_F$, relative to the ball points $T^i_p$;
7) status $T^i_S$.

The lifetime item $T^i_l$ indicates the number of ball points belonging to the considered trajectory $T^i$. The angular coefficient $T^i_\theta$ of the straight line obtained by the linear interpolation of the available ball points, describes the slope of the relative trajectory $T^i$ and it varies in the range $[0, \pi]$; the velocity $T^i_v$ and the acceleration $T^i_a$ describe instead the ball motion; the position range $T_\Delta$ fixes the set of following collinear points that could be considered belonging to the current trajectory. The vector $T^i_F$ contains the ball points $\{P^{T^i_i}(x_j, y_j)|j = 1..n\}$, which have been included in the considered trajectory $T^i$; the vector $T^i_p$ contains the acquisition pulses $\{F^{T^i_i}(x_j, y_j)|j = 1..n\}$ of the ball points $T^i_F$. The status $T^i_S$ of the trajectory can assume two different values, initialized or validated according to the number and the temporal distances among points included in the trajectory. In particular, given a ball position $P(x_t, y_t)$ in to the virtual play-field, the trajectory status $T^i_S$ is set to “initialized” when at least one of the following situations happen:

1) does not exist any previous trajectory ($i = 1$);
2) $P(x_t, y_t)$ does not satisfy collinearity requirement with the points $T^i_p^{-1}$ of the existing trajectory $T^{i-1}$, i.e. the distance between the point $P(x_t, y_t)$ and the straight line obtained interpolating the points $T_p^{-1}$ belonging
to $T_{i-1}$ is greater than a certain threshold (experimentally set taking in account an error due to calibration phase and numerical approximation);

3) $P(x_t, y_t)$ is outside an acceptable range around the predicted position of the ball obtained considering the trajectory cinematic parameters $T_{i-1}$. That is

$$|P(x_t, y_t) - [P^T_{i-1}(x_n, y_n) + (T_{i-1}^T - F^T_{i-1}(x_n, y_n)) + \frac{1}{2}T_{i-1}^T(T_{i-1}^T - F^T_{i-1}(x_n, y_n))^2]| > \varepsilon;$$

This assures the correct identification of new trajectories in the case of deviation from the trajectory direction (case 2) and velocity variation along the trajectory direction (case 3). Anyway a new trajectory has to be confirmed by successive coherent ball observations in order to avoid the generation of a new trajectory in the case of an isolated false positive of ball detection. The status $T_S^i$ is switched to “validated” when $n$ collinear occurrences of the ball are available. Parameter $n$ varies from 1 to 3 depending on the number of acquisition pulses elapsed from the last detection of the ball. In other words, $n$ is computed as the closest integer number to $(2 + \frac{1}{\Delta F}) + 1$ where $\Delta F$ is the number of elapsed frames from the last ball detection of the previous validated trajectory. This choice derives from an experimental assumption: higher $\Delta F$, higher the possibility that the ball is tracing a new trajectory and then it can be validated earlier.

Finally, at the time instant $t$, a validated trajectory $T_i$ is deleted by the system if $(t - F^T(x_N, y_N)) > F_{min}$ where $F^T(x_N, y_N)$ is the last acquisition pulse number of the trajectory and $F_{min}$ is a fixed threshold.

1) Handling Ball Occlusions: From these considerations follow that the capability of the system to correctly compute ball trajectories is strictly dependent on the frequency of ball occurrences from multiple views. Unfortunately, multiple views of the ball cannot be available in each acquisition frame both for ball occlusions and for ball exiting the cameras field of view. For this reason, to improve the ball trajectory computation performance, further analysis procedures have been introduced.

The first procedure starts when some 3D ball positions are known and are spaced out by few acquisition pulses with ball information coming from only one camera. The procedure interpolates the 3D ball trajectory by using the known 3D ball positions (see figure 8). For all the intermediate time instants where information of the ball position is coming from just on view, the center of the ball on the image plane $c_1$ is projected using homography on the ground plane obtaining the point $b_1$. The ball viewing line $l_1$ passing through the points $c_1$ and $b_1$ is computed. If the segment on the minimum distance line between the 3D ball trajectory and the ball viewing line is under an acceptable threshold the midpoint on this segment can be considered as an estimation of the 3D ball position. If the segment is over the threshold, it means that a false positive was found and the point is discarded. In figure 8 the point $b_1$ can be used to estimate the 3D ball position while the point $b_2$ is a false positive and it is not considered.

The second procedure starts up when ball positions into the virtual play-field are not available for many consecutive acquisition pulses and, at the same time, proximity and motion reasonings among moving blobs bring the system to infer that probably a player has kept the ball control. A typical situation of this kind is reported in figure 9 where a defender (wearing blue shirt) stopped the ball (upper image), then he kept the ball control for about 3 seconds
Fig. 8. The 3D ball trajectory interpolated by the system. In some intermediate points the ball was visible by one view.

(75 acquisition pulses), moving around into the play-field, and finally he kicked the ball (bottom image). When the ball was occluded by the defender the system was not able to evaluate the ball trajectory. However the system assumes that missed ball positions overlapped the player positions and that, as soon as new ball data arrive from the processing nodes, the player has kicked the ball (an interaction occurred). A new trajectory is initialized and immediately validated.

The third procedure is activated when the system detects, using motion information, that many consecutive acquisition pulses without ball information are due to the ball exiting the cameras field of view during its flying time. In these situations, the system using the last 3D ball positions interpolates the ball trajectory assuming that the ball after the initial force impressed by the shot is only subject to the gravity. Using the equation of the bullet parabolic motion it is possible to predict the position in which the ball will enter again in the camera field of view. When the ball appears again the system does not initialize a new trajectory but it updates the cinematic parameters of the previous one.

C. Detection of the interactions between the Ball and the Player

Starting from the data projected into the virtual play-field, the interactions between the ball and the player can be localized between two consecutive validated ball trajectories T and S. However the simple intersection of two consecutive validated trajectories cannot be reliable since, as described before, when the ball is very close to the player the system is not able to detect the ball. Generally, the ball is occluded in a number of frames before the shot and only when it is completely separated by the players it is clearly visible by the cameras. Therefore the problem becomes how to localize the interactions between the ball and the player in a more precise way both
in terms of time and space. Geometrical and cinematic hypotheses have been used. The exact spatial localization of the interaction can be assumed to be in the intersection point \( P(x_s, y_s) \) between the two lines that interpolate the two validated trajectories. Besides it is necessary to detect also the instant in which the observed interaction occurred, that for high level processing has to be translated in the exact frame in which the shot was recognized.

Using the estimated kinematic parameters (direction, velocity and acceleration) of the two trajectories, two possible solutions can be considered. Starting from the last points of the first trajectory \( T \) or the first points of the second trajectory \( S \) we can solve the inverse problem of detecting the time \( t_s \) in which the two trajectories assumed the value \( P(x_s, y_s) \). However we decided to use the first trajectory \( T \) since its kinematic parameters are more precise than those of the second trajectory \( S \) (obtained with just few initial points).

The final problem is to detect the player who interacted with the ball. This is done by evaluating, at the instant \( t_s \), the relative Euclidean distances \( d_1, d_2, d_3 \) among the players in that area and respectively:

1) the intersection point \( P(x_s, y_s) \);
2) the last detected ball point \( P^T(x_N, y_N) \) belonging to first trajectory \( T \);
3) the first detected ball point \( P^S(x_0, y_0) \) belonging to second trajectory \( S \).

The player having the minimum overall distance \( D = d_1 + d_2 + d_3 \) is labelled as the player who shot the ball (see figure 10). Anyhow, if the obtained minimum distance value is greater than an experimentally set threshold, the system infers that no player was close to the ball at the time \( t_s \) and none association is done and the relative detected interaction \( P(x_s, y_s) \) is discarded.
Fig. 10. The shot detection in the point \( P(x_s, y_s) \), the intersection of the two lines that fit the trajectory \( T \) and \( S \), and the association between the ball and the player.

IV. EXPERIMENTAL RESULTS

We have applied the proposed method to several image sequences acquired during some matches of the Italian first division football championship 2006-2007. The system was installed in the Friuli Stadium (Udine). The imaging solution uses DALSA Pantera SA 2M30 cameras, achieving a resolution of 1920x1080 pixels (Full HD) at 25fps. The particular model uses a single CCD operating using a Bayer filter. The acquired images are transferred to the six processing nodes by fiber optic cables. Each node is equipped with two Xeon processors (Nocona series) running at 3.4Ghz with hyperthreading enabled. Each node features 2GB of RAM and uses 8 SCSI disks with an individual size of 73GB (configured in RAID0) to store the match, and another 120 GB SATA disk for the operating system and the software. The graphics sub-system is handled by a Geforce 7900GT card with 256MB. Figure 11 shows the six monitors with the images acquired simultaneously.

Fig. 11. The six monitors with the images acquired simultaneously.

In order to evaluate the effectiveness of the system it was necessary to perform a quantitative analysis of the detected shots and relative associations. For this reason a number of image sequences were observed and labeled...
Fig. 12. The global view of the field provided by the system at time $t$ joining the images acquired from the six cameras placed in the stadium.

by a human operator who indicated the real position of the shots and the relative players involved, together with the indication of the time instants in which they happened (frame numbers). At the end of this analysis the ground truth was generated.

Considering that the system requirements could change depending on the application contexts, four different experiments were carried out. Firstly the capability of the proposed system to detect interaction events, i.e. shots, was considered (system evaluation phase 1). The detection of the shots, without additional information about which player generated them, could be, in fact, satisfactory in applications of content based video compression, video annotation and browsing or automatic highlights identification. Then the capability of the system to associate, for each detected shot, the player who really kicked the ball, was evaluated (system evaluation phase 2). This feature becomes fundamental in applications such as tactics analysis or automatic summarization of play. Fine localization in time is actually strategic in applications for real time referee support. Many researchers are actually working on this subject for the increasing interest of automatic solutions that could solve numerous game controversies. For this reason the precision of the proposed system to localize the exact frame in which the detected shots happened, was checked (system evaluation phase 3). Finally performance comparisons with a single view system were carried out in order to point out the real advantages of the proposed multi-view strategy (system evaluation phase 4).

The experiments 1-3 were carried out on 30 minutes long video sequences whereas the experiment 4 was carried out on 3 minutes long video sequences. All the aforesaid test sequences were acquired during a first division match. Actually the system was tested on the images acquired during four matches for a total of about 360 minutes, but during these matches only qualitative analysis of the performances were possible. In the considered sequences the ground truth was manually generated and compared with system results. The six different views of the play-field provided by the system at each acquisition pulse were logically joined in a unique global view (like that reported in 12). From this point of view, the experimental data set consisted respectively of 45000 (experiments 1-3) and 4500 (experiments 4) global views of the field.

The available global views were accurately examined by a human operator: for the 30 minutes long image sequences he labelled 44408 global views as not containing interactions between the players and the ball and 592 global views containing a player shooting or passing the ball; for the 3 minutes long image sequences he labelled 4453 global views as not containing interactions between the players and the ball and 47 global views containing a
player shooting or passing the ball. It should be specified that among the labelled shots there were also some ball
stops by players and also some weak shots due to players who run with the ball.

TABLE I
CAPABILITY OF THE SYSTEM TO CORRECTLY DETECT THE SHOTS: THE TESTS WERE CARRIED OUT ON 30 MINUTES LONG IMAGE
SEQUENCES

<table>
<thead>
<tr>
<th>Correct Detections</th>
<th>Miss-Detections</th>
<th>False Positive Detections</th>
</tr>
</thead>
<tbody>
<tr>
<td>539/592</td>
<td>53/592</td>
<td>18</td>
</tr>
</tbody>
</table>

The table I reports the system performance for the shot detection (system evaluation phase 1). As reported 539
over 592 shots were correctly detected by the system (about the 91%). The miss-detections happened for different
reasons: some missed events were due to the fact that the ball remained occluded between consecutive interactions
by neighboring players, a few missed events were caused by the failure of the ball detection procedure, and others
were lost since the interactions were not appreciable for the slow ball motion. The figure 13 shows three pairs of
images acquired in different moments by two opposite cameras during a complex action: the player in the first two
images on the top stops the ball, then in the central images he runs with the ball next to his feet (weak shot), in
the last two images on the bottom he shoots the ball. While the last interaction is correctly detected by the system,
the first two interactions are not visible because the ball is occluded by one camera. The system provided also 18
false positive interactions caused by spin shots that produced misleading variations in the trajectories. In figure 14,
a spin shot generating an intermediate false interaction (indicated by a light green cross), is reported.
Fig. 13. Three pairs of images taken by the opposite cameras during a complex interaction between the player and the ball: in the images on the top the player stops the ball, in the central ones he run with the ball, in the images on the bottom the player finally shoots the ball.
Fig. 14. A spin shot during which a false interaction was generated by the system.

Fig. 15. Situation in which the player who shot the ball was not correctly identified by the system.

The table II reports the results of the second experiments to detect the system capability of correctly associating the player to the shots (system evaluation phase 2). As reported 493 detected interactions were associated to the player that really shot the ball, 4 shots were associated to a wrong player and finally 18 of them were discarded since no player was found next to the ball at the interaction instant. Actually the 18 discarded interactions, that were not associated, coincided with the 18 wrongly generated by the system due to spin shots or errors in the homographic ball projection (reported in table I). The four wrong associations were due to the proximity of different players to
the ball at the moment of the interaction event: an example of this situation is reported in figure 15, where two opponent players are very close to the ball and the system was not able to correctly detect which of them really kicked the ball.

**TABLE II**
CAPABILITY OF THE SYSTEM TO CORRECTLY ASSOCIATE THE PLAYER TO THE BALL: THE TESTS WERE CARRIED OUT ON THE 539 SHOTS DETECTED BY OUR SYSTEM

<table>
<thead>
<tr>
<th>Correct Association</th>
<th>Wrong Association</th>
<th>No Association</th>
</tr>
</thead>
<tbody>
<tr>
<td>493/539</td>
<td>4/539</td>
<td>18/539</td>
</tr>
</tbody>
</table>

Table III details the system precision to localize the exact frame in which the 539 correctly detected shots happened (system evaluation phase 3). The value $\Delta f_r$ in table III indicates the number of frames that differ between the frame in which the interaction really occurred (ground truth) and the frame in which our system detected the interaction. The experiments demonstrated that the precision in time localization strongly depended on the number of ball point detected after the real interactions. The greater the number of detected ball positions and the more satisfactory the localization performance. More points allows to better estimate the trajectories and their relative kinematic parameters.
Generally, the time localization performances were encouraging: in fact, about the 80% of the detected interactions were localized with a $\Delta f_r$ less or equal to 5 frames (i.e. 0.2 s.). In particular all the situations in which a player executed a long forward pass toward a teammate (that could generate critical situations such as offside events) were correctly processed. For example in figure 16, the two opposite images of one offside event recorded in the considered test sequence are reported: the system correctly localized both in time and space the interactions between the forward player and the ball. The red arrows indicate the ball position next to the player. In figure 17 the exact localization of the interaction is reported in the virtual field.
Fig. 17. The exact localization in the virtual field of the interaction between the forward player and the ball during the offside event occurred in the test sequence.

In about 9% of cases the system was able to detect the shots between the sixth and the eight successive frames and in the remaining 11% the system detected the shots in the successive frames. These delays in the shot detection were due to different reasons: first of all, the ball, after a high speed shot, can be unfocused or deformed and therefore the ball detection algorithm failed; if the ball was occluded from one camera, the lack of 3D ball position estimations can preclude the correct trajectories intersection evaluation; if the kinematic parameters of trajectories are estimated by few points, the estimation of the instant in which interactions happen is not precise.

TABLE III

<table>
<thead>
<tr>
<th>( \Delta f_r )</th>
<th>( \Delta f_r )</th>
<th>( \Delta f_r )</th>
<th>( \Delta f_r )</th>
<th>( \Delta f_r )</th>
</tr>
</thead>
<tbody>
<tr>
<td>[0, 1]</td>
<td>[2, 5]</td>
<td>[6, 8]</td>
<td>[9, 12]</td>
<td>&gt; 12</td>
</tr>
<tr>
<td>192/539</td>
<td>241/539</td>
<td>46/539</td>
<td>35/539</td>
<td>25/539</td>
</tr>
</tbody>
</table>

The obtained results can be considered satisfying both in terms of shot detections and in terms of their temporal localization, especially if we compare the performance with those obtained using single view approaches. For this reason a further experiment was carried out on a sequence of 3 minutes in which the capabilities of a single view approach and the proposed multi view approach were compared (system evaluation phase 4). Tables IV and V report the relative experimental results. The first row of table IV reports the correct detections of the two approaches: some shots that were detected with our multi-view trajectory interpretations, were not visible if observed by the single view approach, both because the ball detection algorithm failed when the ball was close to the player, and because the ball motion could not be perceived by the camera (the shots were in the direction of the camera optical axis). The second row of table IV reports the occurrences of false positive interactions: the single view approach worked worst than the multi-view one because all the rebounds are pointed out as changes in the ball motion.
and many of them generate false interactions due to the presence of some neighboring player (see figure 18). The proposed multi-view approach eliminated these drawbacks reconstructing the ball 3D trajectories and projecting them on the virtual field.

However the most interesting results are those obtained with the temporal localization of detected shots. Table V reports the delays (in number of frames) between the actual shots and those detected by the two approaches: the multi-view approach gave better performance than the single view approach due to the capability to backward infer the temporal localization of shots independently of the frames in which the ball was recognized by the system. Single view approaches, instead, localize the interactions in the frame in which the ball position does not agree with the expected one, on the basis of estimated motion parameters. In this way they introduce a delay from the actual shot frame because shots are associated with the first ball detections that generally correspond to some frames following the actual shots (mainly due to the occlusions caused by the kicking player).

The proposed framework has been implemented using Visual C++ and, although not all the code optimizations have been completed, it is able to process about 10 frames per second on a using images sized 1920x1080 pixels. Further improvements can be obtained by introducing code optimizations and specialized hardware to allow the real time processing of all the acquired frames (25 fps).

TABLE IV
PERFORMANCE COMPARISON IN TERM OF SHOT DETECTION BETWEEN MULTI-VIEW AND SINGLE VIEW APPROACHES: THE TESTS WERE CARRIED OUT ON 3 MINUTES LONG IMAGE SEQUENCES

<table>
<thead>
<tr>
<th></th>
<th>Multiview Approach</th>
<th>Single view approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct Detections</td>
<td>41/47</td>
<td>39/47</td>
</tr>
<tr>
<td>False positive detections</td>
<td>2</td>
<td>25</td>
</tr>
<tr>
<td>Miss-detections</td>
<td>6/47</td>
<td>8/47</td>
</tr>
</tbody>
</table>

TABLE V
PERFORMANCE COMPARISON IN TERM OF TEMPORAL LOCALIZATION OF THE INTERACTIONS BETWEEN MULTI-VIEW AND SINGLE VIEW APPROACHES: TESTS WERE CARRIED OUT ON THE SHOTS DETECTED IN 3 MINUTES LONG IMAGE SEQUENCES

<table>
<thead>
<tr>
<th></th>
<th>$\Delta f_r$ [0, 1]</th>
<th>$\Delta f_r$ [2, 5]</th>
<th>$\Delta f_r$ [6, 8]</th>
<th>$\Delta f_r$ [9, 12]</th>
<th>$\Delta f_r$ &gt; 12</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiview Approach</td>
<td>17/41</td>
<td>15/41</td>
<td>5/41</td>
<td>3/41</td>
<td>1/41</td>
</tr>
</tbody>
</table>

V. DISCUSSION AND CONCLUSIONS

The problem of detecting the interactions between ball and players is fundamental to developing automatic systems for soccer action analysis. In particular, in order to build a visual system for automatic offside detection it
is necessary to use dedicated cameras that minimize the parallax errors, and above all to have visual algorithms able to process in real time the acquired images, detect the exact frame in which the interactions happen and associate the shot with the correct player.

The main contribution of this work is the application of an inference process to the 3D and 2D trajectories to detect interactions between ball and players that for their small dimensions in the image plane cannot be solved with standard pattern recognition algorithms. It should be noted that by using three cameras to cover each side of the field (about 120 meters long), the players dimension is about $40 \times 100$ pixels, and the ball diameters vary in the range $[8, 20]$. Under these imaging conditions, the ball texture information is completely lost and also the ball circular shape can be deformed. The presence of shadows, especially when the ball is on the ground, produces a segmentation with many artifacts. When the ball moves with high velocity it appears with an elliptic shape. The ball recognition task is very difficult, whatever algorithm is used, and above all, it cannot guarantee the recognition when the ball is very close to a player. In this work we experimented how these limits can be overcome applying the multi-view integration for 3D trajectory interpolation and evaluation.

The system has been tested during several matches of the Italian first division football championship and experimental proofs of its effectiveness on a 30 minutes long sequences is reported. Moreover a comparison with single view approaches on 3 minutes of image sequences was carried out. Experimental results demonstrated the effectiveness of the proposed approach. Good performances have been obtained in term of correct shot detections, correct player-shot associations, and in terms of frame precision for shot localization. The multi-view integration greatly improved the results obtained with a single view analysis. Observing the frames correctly provided by the system as those corresponding to the interactions it was possible to see how, although the ball was not detected by the single view analysis, the interpretation of both the 3D ball trajectory and the 2D player trajectories provided satisfying results. Also the processing times of the proposed framework are encouraging: without any code optimization we processed 10 frames per second.

Future works will deal with human posture analysis in order to solve the cases of wrong identification of the
player kicking the ball. Moreover a more complex ball recognition approach in case of merge situations will be considered to assure better time localization of the interactions among ball and players.

REFERENCES
