A Computer Vision Based Web Application for Tracking Soccer Players

João Rodrigues[1,2], Pedro J.S. Cardoso[2], Tiago Vilas[2], Bruno Silva[3], Pedro Rodrigues[2], Antonio Belguinha[2], and Carlos Gomes[2]

[1] Vision Laboratory, LARSyS, University of the Algarve, 8005-139 Faro, Portugal
[2] Instituto Superior de Engenharia, University of the Algarve, 8005-139 Faro, Portugal
[3] Inesting, S.A., 8005-146 Faro, Portugal
{jrodrig,pcardoso}@ualg.pt

Abstract. Soccer is a sport where everyone that is involved with it make all the efforts aiming for excellence. Not only the players need to show their skills on the pitch but also the coach, and the remaining staff, need to have their own tools so that they can perform at higher levels. Footdata is a project to build a new web application product for soccer (football), which integrates two fundamental components of this sport's world: the social and the professional. While the former is an enhanced social platform for soccer professionals and fans, the later can be considered as a Soccer Resource Planning, featuring a system for acquisition and processing information to meet all the soccer management needs. In this paper we focus only in a specific module of the professional component. We will describe the section of the web application that allows to analyse movements and tactics of the players using images directly taken from the pitch or from videos, we will show that it is possible to draw players and ball movements in a web application and detect if those movements occur during a game.

Keywords: Applications, interfaces, soccer, web technologies, information system, computer vision.

1 Introduction

Soccer is a sport where everyone that is involved with it make all the efforts aiming for excellence. Not only the players need to show their skills on the pitch but also the coach, and the remaining staff, need to have their own tools so that they can perform at higher levels. Granting both parts a multi-functional information system (IS), with the objective of minimizing the adverse effects from the most critical and sensitive points of soccer. A match analysis, for example, can generate a huge amount of data. Consequently, it is very important to have a way to process that data, providing only the most important information to the coach, as soon as possible.

Footdata [1] is a project in development by Inesting, S.A., the University of the Algarve and the soccer coach Domingos Paciência. The goal is to build a web

C. Stephanidis and M. Antona (Eds.): UAHCI/HCII 2014, Part I, LNCS 8513, pp. 450–462, 2014.

application product for soccer, which integrates two fundamental components of this sport's world: i) a social network (FootData-Social), which provides the typical features adapted to the soccer reality, and ii) the professional component (FootData-PRO), which can be considered as a Soccer Resource Planning (SRP), featuring a system for acquisition and processing of information to meet all the soccer management needs. The latter includes (between other things) an automated platform to gather information from the teams. This platform will be based on a system that will process live images acquired on-site, using a single or a group of cameras placed together in the stands, or images gathered from a Full HD Handycam. One of the objectives is to add features which will allow the analysis of the soccer match structure, allowing rationalization and optimization of the team's actions regarding the occupation of spaces. All the application structure is supervised by the soccer coach Domingos Paciência.

There are a few commercial systems that partially integrate some of the components of this project, e.g., Kizanaro [3]. We can also find in the literature examples that explore some aspects of the project, like video technology for coaching [4] and computerized video analysis of soccer [5]. Even so, none of those integrate all the technologies and goals of the Footdata project. Please refer to [1, 2] for a more comprehensive review of the literature.

In this paper we will describe the design of part of the web application, tacking special focus on: (i) the needs and restrictions of the soccer coach, and (ii) on the computer vision based module for tracking and analyzing the soccer player(s) movements and teams tactics. The main contribution is this interconnection between the acquisition and analysis systems and the web interface for designing the movements and tactics. Furthermore, this module connected with the FootData-Social can give universal access to the inside of the statistics and performance in the soccer's world. Of course, all the information from the FootData-PRO is by default confidential and will need permissions from the actors (e.g., clubs, athletes and coaches) to be published in the social platform.

The remaining paper is divided as follows: a brief description of the FootData-PRO system and the involved technologies to implement it are presented in section 2, in section 3 we introduce one web interface of the application, sections 4 and 5 describes the ball and tracking of players and the game analysis, respectively. The final section presents conclusions and future work.

2 The System and The Technologies

Footdata-PRO [1] features an acquisition and information system divided in five main modules: (a) Field module, where the ball's and players' location in the pitch is done by computer vision. This information is sent through an Internet connection to a server, where the match tactical plans are verified, based on the information available from prepared tactical schemes, trainings, individual players, and collective information from each team. Supported in the above procedure, in the coach's (user) profile and in what is occurring in real-time in the field, selected information is sent to the coach's mobile device. (b) Center

module, where all the data and the main computational procedures are congregated (including the processing and management of previously recorded videos). (c) Coach module, where the technical staff inputs and gets access to detailed information of every topic related to the soccer match, scouting, medical department, etc. (d) Player module, used to send specific information to the players, and the (e) Presentation module, where semi-automatic presentations are prepared for the players in preparation moments. An integrated web application, where usability is a key factor, will combine all the above features in a single place, including video manipulation tools, tactical planning/drawing tools, semi-automatic presentation tools, etc.

Being FootData-PRO a web application (inserted in a context of cloud-computing), the technologies employed to built the application discussed in this paper can be summarized to the following: The front-end has implemented using HTML5, CSS3 and JavaScript [9]. With regard to the back-end, this module has structured using Node.js [10] and Python. In both cases, we resorted to various JavaScript frameworks. In the latter case it was also used Django [7], which is a high-level Python web framework [12]. Python was also the programming language used to produce all the statistics as well to "decode" the drawn movements and team tactics (see Sec. 3) into code. The produce code is then used to compare the information sketched in those plans with the information retrieved from the tracking (Sec. 4) and achieve the game analysis (Sec. 5). The tracking system was implemented using C++ and OpenCV [16]. WebSockets [17] and Socket.IO [18] were used for the communication between the different modules. For the storage, necessary for the IS, we are at the moment using two NoSQL databases: a document oriented databased, MongoDB [8], and a graph database, Titan [11]. Finally, we can also refer Yeoman [6], which encompasses a number of frameworks (e.g., AngularJS), providing a development and production environment for a web application by automating some tasks related to the initialization, construction and testing of the application.

3 Interfaces

All the interfaces were projected having the best practices in consideration and bearing in mind three fundamental key points: (a) they must be user-friendly and very intuitive for the users (e.g., technical staff, players); (b) functional, i.e., they have to provide an easy way to add and edit text and shapes, along with some other functions (e.g., drag, resize, recolor or rotate); and (c) most of them had to "work" in a wide range of devices (e.g., mobile phones, phablets, tablets and personal computers), i.e., Responsive Design [13]. Furthermore, some of the interfaces have to show very dynamic situations since we want to make animations with the shapes we draw, while other have to show videos and allow to put layers of shapes and/or text over them. Bottom line, since almost all the interfaces were quite demanding in terms of design and usability, a Design Thinking approach [14] was used to create them.

In order to integrate the following sections, in this paper we only show a flavor of those interfaces. Figure 1 shows an example of a planification drawn in the

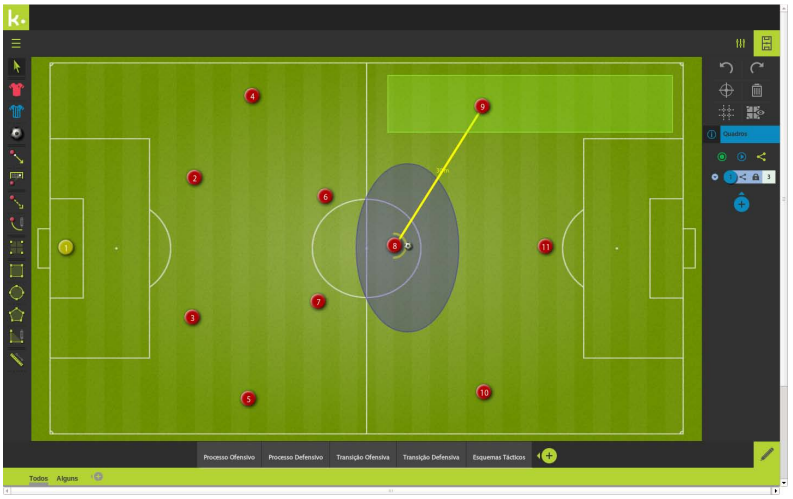

Fig. 1. Example of a planification drawn in Field Editor interface

Field Editor, a tool which targets the following generic points: (a) provide the coaches with a web platform where he can easily represent players movements to be applied in the training sessions and in matches; (b) provide an animation feature to help the coach to illustrate its intentions regarding a certain play or movement; (c) provide a way to extract data from a represented movement or action so it can be processed and analyzed later by matching it with the tracking obtained from a game. In this paper we will focus mostly in that last point. Using the Field Editor, the coach can draw a wide range of situations, from "static" players positions, e.g. a player or group of players appearing in a specific region of the field, to "animations," like the very simple one shown in the figure: player $\#_{tA}$ in region 1 passing ball to player $\#_{tB}$ in region 2. However, much more complex animations (movements, tactics) are also possible to edit, where all the team players are involved and different sectors of the team have to do different (tactical) movements in function of the moment of the game and the position of the ball. More information about this tool with the a alpha interface can be seen in [2]. The shown interface is in Portuguese, but all the main languages will be soon available. Also, due to the confidentiality agreement, some very specific parts of the interface are omitted.

4 Classification and Localization of Soccer Players and Ball

The tracking of the soccer players and ball is done in five stages: (1) the detection of the soccer pitch, (2) the detection of players and ball, (3) the assignment of players to their teams, (4) the tracking of players and ball, and (5) computation of their localization (in meters) in the pitch.

(1) **Soccer pitch detection** – let $I_{RGB}(x, y, t)$ and $I_{HSV}(x, y, t)$ represent a frame with dimensions $M \times N$, acquired for instance by a camera (typically an HandyCam) in the stands, where M is the width and N is the height of the image, t is the instant when the frame was acquired and RGB and HSV the color spaces [15]. The first step was the soccer pitch detection, which consists in (a) the segmentation of the green (grass) regions for each frame using I_{HSV}. This was done by a semi-automatic process, where selected pitch patches were used to compute the HSV thresholds intervals. For the pitch shown in Fig. 2 top, it was used the following thresholds levels: $H \in [0\%, 50\%]$, $S \in [12\%, 100\%]$ and $V \in [27\%, 100\%]$, which returns a binary image I_c, with $I_c = 1$ corresponding to the green pixels.

The next step has the purpose to eliminate the areas outside the pitch. This was carried out in several stages: First (b), in I_c was chosen a central point of the image as the seed (in most frames, the central area corresponds to the pitch, the exceptions are zooms to players or to the stands). The central point of I_c has to have a green color (see above), and recursively from this initial point, using 3×3 neighborhood, all central pixels that have all neighbors green are kept, $I'_c = 1$, the remaining are put to 0 ($I'_c = 0$). Over the last image, but only in areas where exists pitch, i.e. $I'_c = 1$, was applied (c) the Canny edge (I_{Ce}) detection [15] on I_{RGB}. After this, (d) the Hough transform [15] was applied over the I_{Ce}, in order to detect the lines (I_{Hl}) that limits the pitch. Finally (e), using I_{Hl} the most horizontal and vertical lines (left, right, up and down) are computed. Those lines correspond to the limits of the pitch, and everything outside these limits (lines) was removed. The final results consist only in the segmentation of the pitch (with some black blobs inside), I_{fj}.

(2) **Players and ball detection** – for the player detection it was used the I_{fj} image. In this image, most of the work for the players detection was already done, once the pitch area was delimited, the black blobs inside this area are likely players. Nevertheless, 3 major problems can arise: (a) blobs that are not players, e.g., areas with short grass, (b) players that almost disappear due to low image definition, or being partially equipped in "green," and (c) the overlap of various players. To solve these situations and to obtain only the contours of the players, morphological filters were applied, i.e., first it was applied the dilation filter (D) followed by an erosion filter (E) and finally a 9×9 medium filter (M). The result was an image, $I_{jf}(x, y, t) = M(E(D(I_{fj}(x, y, t))))$, where the players were better defined and specific noise regions were removed. After this, since some player still have more than one region (usually due to the equipment that they use) a vertical mask was applied to connect those regions.

The next step was the confirmation that each blob corresponding to a possible player (in I_{jf}) has at least one edge contour correspondence in I_{Ce}. Finally, it was checked if the final blob region of each player (in I_{jf}) doesn't had more than 10% of green pixels (using I_H). The final image after this validation was called I_f. To differentiate between players and the ball, all pixels with the "white color" (in I_V) inside I_f were detected. Those pixels can represent lines belonging to the pitch, the ball, and if it was the case, teams that had predominantly white

equipments. For this purpose an threshold with $V \in [75\%, 100\%]$ was applied to I_V, returning a binary image, I_b, with all the white regions inside the pitch ($I_b = 1$). Combining the previous information with the size and shape of the blobs, the blob that corresponds to the ball can be classified. The first step, consists in detecting if the blob as more or less circular or oval shape, which was different from the player blobs that had (usually) a more "vertical-rectangular" shape. This way most of the players blobs were discarded. The second step, was to remove the areas corresponding to the pitch marks, eliminating the areas that correspond to the lines detected by the Hough transform in I_{Hl}. The final step consists in removing areas which exceed $M \times N/k$ pixels ($k = 100000$, this value was empirically determined). Figure 2 top shows the players delimited by rectangles, and the ball by a circle. Having the ball classified, it was necessary to classify the blobs as players from team A or B, referee or keepers.

(3) **Assignment of the players to their teams** – the players' classification to a team was based on the color of the equipment. Assuming that the video started at the beginning of the game, a group of 7 players on the right, and 7 players on the left of the middle field line were automatically selected. If the video didn't start at the beginning of the game, then a semi-automatic process selects the most probable 7 player from each team. Then for each team in separate, and using the blobs regions classified as players in I_f, the average of Hue using the I_H image was computed. Having the two average Hues values from team A and B, the middle threshold was computed to separate the teams.

The keepers were relatively easer to classify, once they correspond to the first blob that appear in the right and in the left of the image with significant difference from the two average Hues values computed for the teams (it is important to remember that the keepers use a different equipment from the teams). The referee was a major problem. If the video starts from the beginning of the game, the referee usually was very close to the middle field line. In those cases, the referee was a blob with a Hue different from the players near the middle line. If the video starts from a different position then a semi-automatic process is applied, i.e., the referee was considered the blob with the most different Hue (comparing with the Hues from the teams), and if by mistake was assigned as a team player, then a web tool (not shown in the paper) allows the user to manually change the player classified as referee and vise-versa.

All those Hues once detected were memorized by the system and adapted dynamically in function of the video conditions (e.g., light). The numbers assigned to the players were in function of the position they occupy in the pitch, and those numbers were associated to the real player number and name in function of the position posted in the game sheet (e.g., right-back). If necessary, this can be corrected later using the web tool mentioned above. Finally, obviously there is also the validation that it was impossible to assign more than 1 referee, 2 keepers and 10 players per team. Figure 2 shows the result of the player's assignment to a team and referee identification.

(4) **Players' positions in the pitch** – in order to make a correct analysis of a soccer game, it is important to know as accurate as possible the correct

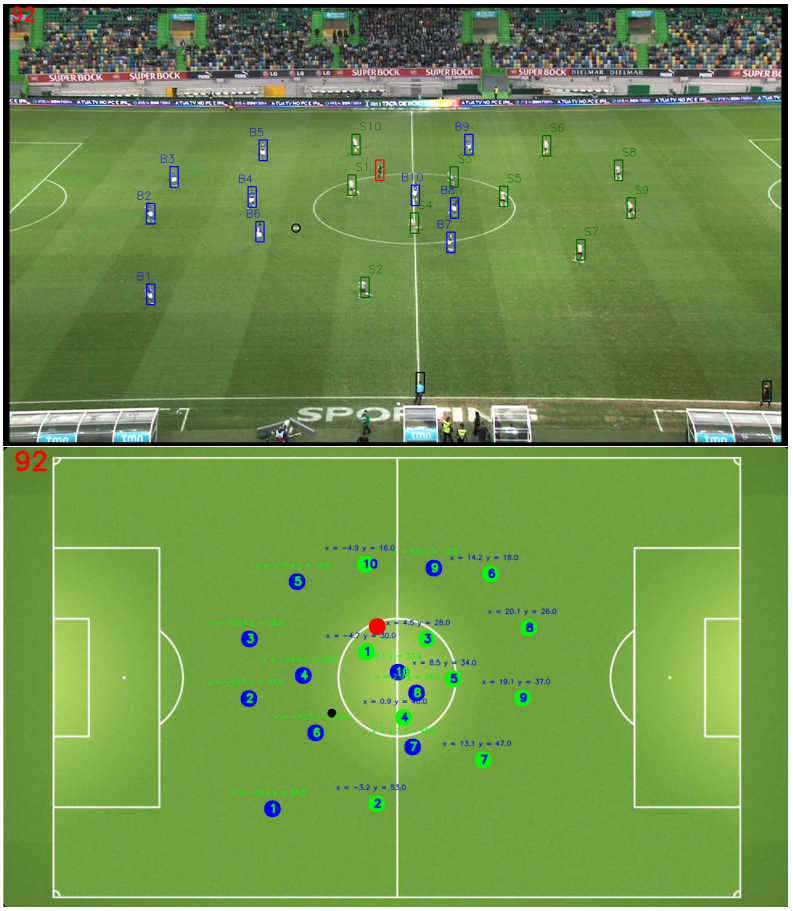

Fig. 2. In the top, the result of the ball, referee and players detection and assignment to their respective team. Bottom, the representation of the players' position considering as the origin of coordinates system the interception between the middle field line and the top lateral line.

position of the players in the pitch. To calculate that position, a perspective transformation (homography) [15] from each frame to a normalized pitch was needed, see Fig. 2 bottom. For this purpose a set of references in the pitch was computed. Those were extracted from the lines detected in the Hough transform, I_{Hl}, and the pitch delimitations in I_f. On the other hand, for the position of the players, it was considered the middle point in the bottom line of the box that limits the players, which corresponds in most cases to the coordinates of the players' feet, see Fig. 2 top and the respective projection in the bottom.

Since the pitches have different sizes (weight and height), it was important to compute the size of a pixel (in our case, in meters). For this computation, in the first frames (or as soon as possible) the lines of the center circle, penalty

area or goal areas had to be automatically detected to calibrate the value of a pixel in meters, using the fact that their dimensions are the same for all pitches. An automatic process was then applied every time there was a small zoom, or a zoom to a player.

(5) **Tracking of players and ball** – the players and ball tracking doesn't require the repeating for all frames of all the operations mentioned in steps 2 and 3. Though simple, the implemented tracking process was effective, except in situation of great confluence of player, such as corners or discussions with the referee. Nevertheless, the main focus of this framework is not to follow the players everywhere, but to compute the distances between players and from the players to the ball during open plays. The tracking was divided into five steps for each frame t: (a) the distance (in meters) was calculated between each player in the previous frame, $t-1$, to the players in the current frame, t. Making the correspondence to the player that was closer. If there were two or more players at the same distance (non-overlapping) the player was assigned to the one that in the previous frame has the nearest Hue value. (b) If there was no players in frame t near to the position occupied by some player in frame $t-1$ (few meters apart), then a player with the same color as the one in frame $t-1$ was looked up in frame t, using circular areas with increasing diameters and centered in the original coordinates. Furthermore, it was used a trajectory computed from the previous 5 frames to limit within the circle the "sectors" of the search. (c) For situations in which a particular player was not tracked/located during a long frame set (2 seconds), this player was then searched from the position where he was "lost" using again circular areas with increasing diameters. The first blob found that was not tracked and has the same Hue value was assigned as the lost player. (d) It is important to note that in the majority of the frames, there was the possibility that some players were not detected because they are not in the field of view of the camera, e.g., at the beginning of the game, due to fact that the camera doesn't capture the entire pitch. Every player that appear at the left or right (top or bottom, depending if there has a zoom applied) was considered as new player to track, except if a player was "lost" on a previous frame in a nearby position. (e) Finally, it was also necessary to verify the existence of collisions. If after the above procedure (a-c), a player was not found, it was assumed that there was a collision (junction of two or more players). For those situations, all blobs that correspond to players, that were in positions very close and were lost, were associated with a single blob, assigning to 2 or more players that were tracked in $t-1$ in a nearby position. When the collision ends (separation of two or more players) the Hue of each blob were checked and players reassigned.

After a collision there was always the possibility of players being wrongly identified, for instance when the collision was between two players of the same team. To solve this, the already mentioned web tool can be used to correct the the players assignment. The process of tracking the ball and referee is similar to the players. Having the players and ball tracked, i.e., the x and y coordinates of each one on the field, now we can compare/match the "draws" (tactics) done in the Field Editor and return the game analysis.

5 Game Analysis

In this paper we only focus on return: (1) heat, (2) passes and (3) ball losses maps, which can be a good resource to measure a player/team performance and a base to their improvement. More complex movements drawn in the Field Editor are also possible to be automatically analyzed, nevertheless: (a) movements from different coaches need to be drawn and tested for a complete validation of this tool and (b) due to the confidentiality agreement some of the tested tactics (game moments) are not allowed to be shown.

(1) **Heat maps** – the heat maps are graphical representation of data where the positions of the players are inserted into an incremental matrix which is then represented with colors. In sports, the heat map can have different uses. Depending on the sport, heat maps help to visually analyze which areas a team, a player, or a set of players preferably occupy. In the particular case of soccer, they can be used to infer if a player has a preferred side, area, or if the defense is occupying the expected positions.

In our case, the data structure behind the heat map is a matrix, HM, that has the same size as the image resolution used to draw it. HM is build using the data from the tracking system, that returns for each frame the players' and ball's positions in meters, $\{frame_{id} : t, team_A : \{player_1 : (x_1, y_1), player_2 : (x_2, y_2), \ldots\}, team_B : \{\ldots\}, ball : (x, y)\}$. Those positions are relative to the pitch top left corner and are obtained using an perspective transformation (see Section 4 – *Players' positions in the pitch*).

To compute HM we start by setting it to the null matrix, $HM = [0]$. Then for each frame a cross-multiplication is used to get the corresponding positions of the players in the pitch, (x, y), to their corresponding entry in HM matrix, (x', y'), and $HM_{x'y'}$ is incremented by one unit. Once all the frames and players are processed, the matrix is normalized, i.e., each entry of the matrix is divided by the matrix maximum element. Finally, the HM values (ranging from 0 to 1) are used to build the output image, by setting the color opacity of the pixels that form the heat map. Figure 3 (top left) sketches a team heat map from one minute of a soccer match. The heat map can also be filtered within different time intervals, and multiple or individual players (on the right).

(2) **Passes maps** – these maps are a graphical representation of the passes made by the players in the game. It shows the passing player's number and the receiver's number. The passes map is a tool which allows several analyses, like how a team organizes the attack or maintains the ball possession. Other statistics can also be inferred like who are the most important players in the teams' strategies, in the sense that they have more ball actions. The passes are detected using the following process: for every frame, (a) the distance between each player and the ball is calculated. (b) If the ball enters in a player's radius (the Field Editor tool sets to 3 meters by default; this value can be adjust for more/less precision on the pass detection) then that frame and player are tagged, and the ball's vectorial velocity and direction are calculated using the current position and its position five frames before. (c) If the ball leaves that player's radius then the vectorial velocity and direction are also calculated, considering the ball's

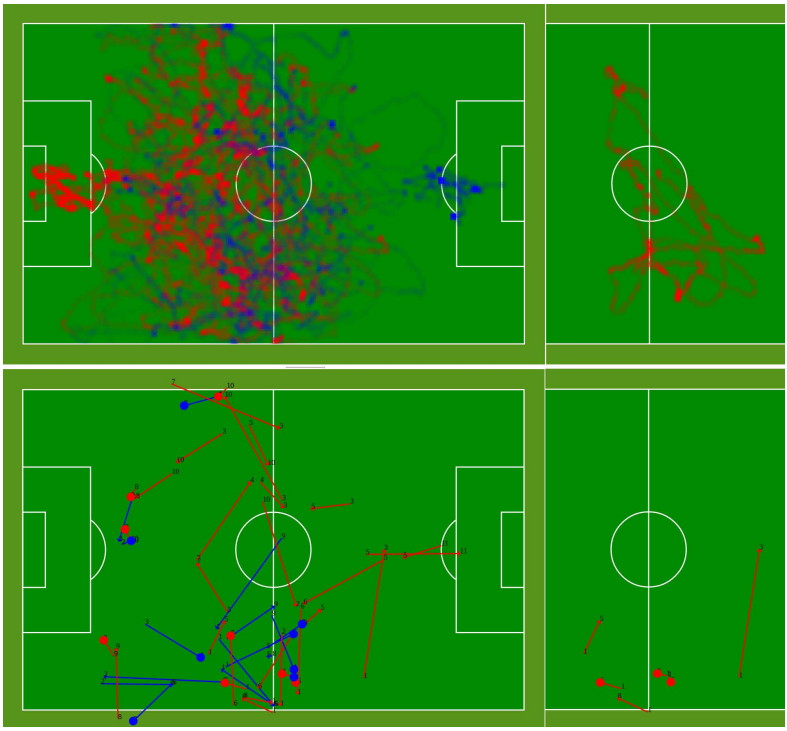

Fig. 3. Example of team heat map (top left), single player heat map (top right), teams passes/losses map (bottom left), and single player passes/losses map (bottom right) from one minute of a soccer match

current position and its position five frames after. (d) The velocity and direction values are then subjected to a set of conditions in order to probabilistically state if it is a pass or not, i.e, if there is a significant change in the ball's direction or velocity then it is probabilistically considered as a pass. Figure 3 (bottom) shows a passes map where it's possible to see all the passes made by both teams (bottom left) or by a single player (bottom right). Simultaneously, a passes log is built as the game is processed, containing a list with the intervening players numbers, teams and time of the pass execution and reception.

The passes can also be filtered using the Field Editor interface. For example, Fig. 1 illustrates a pass between two players. The drawn actions are converted using the (JSON) strings created from the Field Editor interface to another JSON document that contains the relevant information about the depicted movements. This structure is then used to create a set of coded functions (basically containing if-then-else statements) which are then matched to the tracking data, in order to detect the drawn movements (a more detailed explanation of this procedure is out the focus of this paper).

(3) **Ball loss maps** – similar to the passes maps, we have the ball loss maps that can also be filtered by time interval and by player/team (see Fig. 3 bot-

tom where loss passes are marked as blue an red circles). Using this map, it's possible to see which player loses more balls, or in which areas more ball are loss. Analyzing both the passes and loss ball map, we can have a general view about the players and teams performance, by seeing how many correct passes were made and how many balls were loss. The loss ball map is generated using the passes log as base. The passes algorithm counts the passes made by players from different teams, so that when a pass is made to a player of the other team we have a ball loss. Although, we have to distinguish two cases: ball loss made when passing or when dribbling. In order to distinguish between both cases, its considered the distance between the position where the pass was made and where it was received. If the distance is less than two times the player's radius then it is considered that the ball was lost while dribbling, otherwise the ball was lost while passing.

Despite the fact that this paper discusses only heat, passes and ball loss maps, much more movements can be drawn in the Field Editor. For example, specific zones of the field, player or group the players can be specified, even different situation of the game can be specify, this is only a small sample of can the tool do. More complex movements (game models) including dynamic movements of all the team are under the confidentiality agreement.

6 Conclusions

Paper support maintains until our days as one of the basic ways to log the player's and team's actions (e.g., passes, shots on target, ball possession, recurrent movements). At the end of the matches, those documents are statistically analyzed by elements of technical staff which in turn notify the coaches and the players about their actions. The log work is made by someone that is observing the game (live or recorded) and marking those moments by hand. This is a slow process and, in live games, can require more than one person to process all actions with an optimum precision.

This paper shows (a module of) an web application which is an alternative to that kind of work. In this case, we have shown a small part of the overall system that is being built to help coaches and technical staff in the improvement of their players and team's performance. The part of the system shown allows the user to draw a movement or a group of movements in a web application. After that, those movements can be detected automatically during the game, using the tracking data obtained from Full HD videos or, in a near future, on site in real time from live images acquired during a game. The returned data is then used to show individual or team actions. With this process, the information can be given in real time, allowing an instantaneous analysis of the players and teams actions, which gives to the technical staff the possibility to make in game adjustments, based on that information. As expected, knowing the opposite team performance is also possible with this system, which can give an extra advantage to those using these features.

The system is for now being optimized for soccer, but the difference to other ball sports is not big, which makes us believe that in a near future our system will be adapted to other sports.

The work presented is this paper is focused on the technical team. Nevertheless, different levels of accesses will be available inside each club for the entire Footdata-PRO web application. This FootData-PRO module is directly connected with FootData-Social, which will allow to give universal access to the non-confidential information (set by the users/clubs) retrieved by the system. In the future the consortium is studying how to use the information (e.g., from the position of the players, ball, heat maps, passes, ball losses) to develop applications for specific publics, to whom the system can explain in detail what is happening in the field, by automatically describe the game in several languages, e.g., in Braille to deaf people or as a complement to the traditional "narration" to blind people.

Acknowledgments. This work was partly supported by the Portuguese Foundation for Science and Technology (FCT), project LARSyS PEst-OE/EEI/LA0009 /2013 and project FootData QREN I&DT, n. 23119. We also thanks to project leader Inesting, S.A. [www.inesting.com], and the consultant soccer coach Domingos Paciência.

References

[1] Rodrigues, P., Belguinha, A., Gomes, C., Cardoso, P., Vilas, T., Mestre, R., Rodrigues, J.M.F.: Open Source Technologies Involved in Constructing a Web-Based Football Information System. In: Rocha, Á., Correia, A.M., Wilson, T., Stroetmann, K.A. (eds.) Advances in Information Systems and Technologies. AISC, vol. 206, pp. 715–723. Springer, Heidelberg (2013)
[2] Rodrigues, P., Cardoso, P., Rodrigues, J.M.F.: A Field, Tracking and Video Editor Tool for a Football Resource Planner. In: 8th Iberian Conf. on Information Systems and Technologies, Lisbon, Portugal, June 19-22, vol. 1, pp. 734–739 (2013)
[3] Kizanaro (2013), `http://www.kizanaro.com`
[4] Wilson, B.: Development in video technology for coaching. Sports Technology 1(1), 34–40 (2008)
[5] Duh, D.J., Chang, S.Y., Chen, S.Y., Kan, C.C.: Automatic broadcast soccer video analysis, player detection, and tracking based on color histogram. In: Juang, J., Huang, Y.C. (eds.) Intelligent Technologies and Engineering Systems. LNEE, vol. 234, pp. 123–130. Springer, Heidelberg (2013)
[6] Yeoman (2013), `http://yeoman.io/`
[7] Django (2013), `http://www.djangoproject.com`
[8] mongoDB (2013), `http://www.mongodb.org/`
[9] JavaScript (2013), `http://www.w3schools.com/js/`
[10] Node.js (2013), `http://nodejs.org/`
[11] Titan (2013), `http://thinkaurelius.github.io/titan/`
[12] Python (2013), `http://www.python.org/`
[13] LaGrone, B.: HTML5 and CSS3 Responsive Web Design Cookbook. Packt Publishing (2013)

[14] Lockwood, T.: Design Thinking: Integrating Innovation, Customer Experience and Brand Value. Allworth, New York (2010)
[15] Sebe, N., Lew, M.: Robust computer vision: Theory and applications. Kluwer Academic Publishers, The Netherlands (2003)
[16] OpenCV (2013), http://opencv.org/
[17] WebSockets (2013), http://www.websocket.org/
[18] Sockets.IO (2013), http://socket.io/