An Automated Refereeing and Analysis Tool for the Four-Legged League*

Javier Ruiz-del-Solar, Patricio Loncomilla, and Paul Vallejos

Department of Electrical Engineering, Universidad de Chile

Abstract. The aim of this paper is to propose an automated refereeing and analysis tool for robot soccer. This computer vision based tool can be applied for diverse tasks such as: (i) automated game refereeing, (ii) computer-based analysis of the game, and derivation of game statistics, (iii) automated annotations and semantic descriptions of the game, which could be used for the automatic generation of training data for learning complex high-level behaviors, and (iv) automatic acquisition of real game data to be used in robot soccer simulators. The most attractive application of the tool is automated refereeing. In this case, the refereeing system is built using a processing unit (standard PC) and some static and/or moving video cameras. The system can interact with the robot players and with the game controller using wireless data communication, and with the human spectators and human second referees by speech synthesis mechanisms or using visual displays. We present a refereeing and analysis system for the RoboCup Four-Legged League. This system is composed by three modules: object perception, tracking, and action analysis. The camera placement issue is solved by human controlled camera's placement and movement. Some preliminary experimental results of this system are presented.

1 Introduction

One of the RoboCup main goals is allowing robots play soccer as humans do. One interesting extension of this idea is having automated refereeing in the soccer games. The long-term goal could be developing robot referees, but an interesting intermediate step, to be developed in the next few years, is building automated refereeing using static and/or moving video cameras. Such a refereeing system can interact with the robot players and with the game controller using wireless data communication, and with the human spectators and human second referees by speech synthesis mechanisms or using visual displays.

Video-based refereeing employs computer vision methods that allow automating tasks like tracking, refereeing (taking refereeing decisions), annotation, indexing and generation of semantic descriptions. Therefore, an automated soccer referee is an analysis tool that can also be employed for performing a computer-based analysis of the game, an automated summary of the game, or for obtaining game statistics (% of the time that the ball is in each half of the field, % of ball possession of each team, number of goals of each team player, number of direct kicks to the goal by each team,

* This research was partially supported by FONDECYT (Chile) under Project Number 1061158.

G. Lakemeyer et al. (Eds.): RoboCup 2006, LNAI 4434, pp. 206–218, 2007.

number of faults of each team and each team player as ball holding, illegal defender, etc.). This statistical analysis can be also carried out offline, that means that the tool can be employed for analyzing game videos obtained using different recording systems.

Moreover, the same analysis tool can be used for obtaining annotations and semantic descriptions of the game, which could be used for the automatic generation of training data for learning complex high-level behaviors, as for example robot passing or team strategy planning. Furthermore, taking into account the availability of realistic simulators in some of the RoboCup soccer leagues (e.g., SimRobot [1] and UCHILSIM [2] in the Four-Legged league), the obtained annotations and semantic descriptions of the game can be used for the acquisition of real game data to be used in simulations. In this way, a robot controller can be adapted in a simulation of a real game situation, previously acquired by the automated analysis tool.

In this context, the aim of this paper is to propose such an automated refereeing and analysis system for the RoboCup Four-Legged League. To the best of our knowledge a similar system has not been proposed in the literature.

This paper is organized as follows. In section 2 some related work is presented. The here-proposed refereeing and analysis system is presented in section 3. In section 4 some experimental results of the application of this system are presented. Finally, in section 5 some conclusions of this work are given.

2 Related Work

Computer vision based analysis of sport videos has been addressed by many authors (e.g. [3]-[23]), and nowadays is a hot topic within the multimedia video analysis community. There are also successful video analysis programs that are being used by TV sport channels (e.g. Hawkeye [24] and Questec [25]). Applications have been developed in almost all massive sports such as tennis ([13][14][16][20]), soccer ([4][6][7][17][18][22][23]), baseball ([10][15]), basketball ([12]), and American football ([11]). However, to the best of our knowledge the automatic analysis of robot sports, as for example robot soccer, has not been addressed in the literature.

The major research issues for the automatic analysis of human sports include (see a survey in [20]): ball and players tracking, landmarks detection (goal mouth, oval, side lines, corners, etc.), tactic analysis for providing training assistance, highlight extraction (ace events in tennis, score events in soccer and basketball, etc.), video summarization (automatic generation of game summaries), content insertion (e.g. commercial banner replacement in the field), and computer-assisted refereeing (e.g. offside detection). Some of these research issues are still open as for example fully autonomous tactic analysis, soccer offside detection considering player's intention, or video summarization; current solutions are semi-automatic and provide assistance to human referees or video operators.

We believe that many of the accumulated knowledge in the automatic analysis of human sports can be employed for the automatic analysis of robot sports. Probably the main obvious difference being that in the robot sport case, robot players need to be detected and identified. The main problem for identifying robots is the fact that all players look the same. However, the robots can be individualized using the team uniform color [26], the player's number [28], or both.

The here-proposed automated refereeing and analysis system for the RoboCup Four-Legged league makes use of the accumulated knowledge in automatic analysis of human sports, and the image analysis tools developed in the Four-Legged league in the last years.

3 Proposed Automated Refereeing and Analysis System

3.1 General Description

The block diagram of the proposed refereeing and analysis system is shown in figure 1. The system is composed by three main subsystems *Object Perception*, *Tracking*, and *Motion & Action Analysis*, and makes use of two databases *Game Rules* (input) and *Game Statistics* (output).

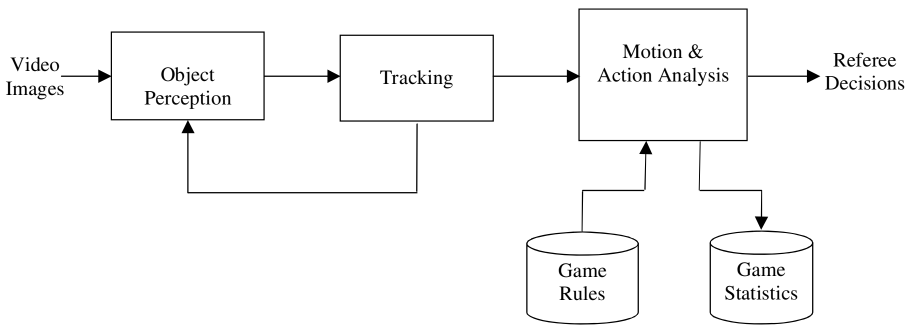


Fig. 1. Block diagram of the refereeing and analysis tool

The *Object Perception* module has three main functions: object detection, object identification, and camera self-localization. First, all the objects of interest for the soccer game (field carpet, field and goal lines, goals, beacons, robot players, ball) are detected using color segmentation and some simple rules. No external objects, as for example, spectators or human referee's legs are detected. The detection operation allows also the identification of goals, beacons and the ball, because each of them has a defined form and color composition (see current definitions in [30]). The identification of field and goal lines is performed using the relative distance from the detected lines to the already identified beacons and goals The identification of the robot players is much more difficult, and is performed using local image descriptors as proposed in [28]. Real-time robot identification is achieved thanks to the computation of the local descriptors only in the detected robot regions, and because of this identification is carried out only at the beginning of each robot tracking sequence (see the feedback connection from *Tracking* to *Object Perception* in figure 1). The camera self-localization is computed using the pose of the landmarks (goals and beacons) and the lines, and the internal parameters of the camera.

The *Tracking* module is in charge of tracking the moving objects, i.e. the ball and the robot players. The implemented tracking system is built using the mean shift algorithm [29], applied over the original image (not the segmented one). The seeds of the tracking process are the detected ball and robot players. As in [31], a Kalman Filter is employed for maintaining an actualized feature model for mean shift. In addition, a fast and robust line's tracking system was implemented. Using this system it is not necessary the detection of the lines in each frame.

The *Motion & Action Analysis* module is in charge of analyzing the game dynamics and the actions performed by the players (e.g., kicking or passing), and detecting game relevant events (goal, ball out of the field, illegal defender, etc.). This analysis is carried out using information about static and moving detected objects, and the game rules, which are retrieved from the *Game Rules* database. The outputs of the *Motion & Action Analysis* module are refereeing decisions (e.g. goal was scored by team A), and game statistics (e.g. player 2 from team A score a goal) that are stored in the corresponding database.

3.2 Object Perception

(i) *Beacons, Goals, Carpet and Ball Detection and Identification.* Landmarks (beacons and goals), the field carpet and the ball are perceived using a standard color-based vision method that consists of three main stages: color classification (segmentation), blob formation and object detection. Color classification and blob formation are relatively common principles amongst RoboCup teams and standard solutions for these tasks already exist (for further information read for example [27]). In the RoboCup Four-Legged league all game objects has a distinctive form and a distinctive color composition, therefore they can be detected (identified) in the color space using simple rules (e.g. [26]). For instance, the ball is spherical and orange, and there is no other orange object in the field. Usually, false detections are filtered out using some geometrical considerations (e.g. to be near the image horizon). See [26] for details.

(ii) *Robots Detection and Identification.* Sony legged robots are non-rigid objects, and therefore their form is difficult to describe. However, they can be detected in the color segmented images using some simple heuristics, and the facts that (a) robots are over the field carpet, and (b) robots have a different color composition than the other field objects (landmarks and ball). Therefore, after detecting all other field objects, non-green patches over the field carpet are robot candidates. The color composition of these patches is checked. Robots are white (ERS7), while their uniforms are blue or red. Hence, each robot candidate patch should include two of these colors in a given amount to be considered a robot. This color information can also be used to classify the robot in one of two classes: blue team robot or red team robot. Certainly, other colors can also be present in the candidate patches, but in a much smaller amount (black because of the shadows, or other colors because of a wrong segmentation). After the robots' detection and team classification, the remaining task is the robot identification, that is, to know the robot identity or number. This identification is performed by calculating local descriptors (SIFT features [32]) in the candidate patch, and by comparing those descriptors against descriptors of robots' templates, already stored in a robot model database. For performing this task we use the same methodology proposed in [28]. It is important to remember that all robots and also the ball are tracked in the *Tracking* module. Therefore, during the time that the robots (and the ball) are tracked (several frames) they do not need to be detected and identified again.

(iii) *Lines Detection and Identification.* Images can be analyzed very quickly using a grid of vertical and horizontal scan virtual rays; lines perpendicular to these scan rays can be found easily, generating line keypoints, which can be used to achieve line detection. There are two ways for generating these line keypoints: (a) by detecting differences in the intensity channel along the scan line, and selecting those that exceeds a given threshold, or (b) by making a convolution between the scan line and a set of edge detection filters (e.g. $[-1;+2;-1]$) of different sizes (fast implementation using differences between pixel's values), and selecting those points where the filtering operation exceeds a given threshold and that correspond to a local maximum over the line and over the neighboring scales. We choose the second option because is more robust against motion blur, and line scale and orientation changes in the image. Candidate line keypoints that do not have green (carpet) pixels around are filtered out. Then, the remaining candidate line keypoints are summarized using a Hough Transform, which allows the determination of the lines' parameters. A last test for accepting the candidate lines consist on generating three parallel scan lines, where the middle scan line corresponds to the detected line. If the mean intensity value of the central scan line is much larger than the intensity value of the lateral scan lines, the line candidate is accepted. After line detection, the line classification is performed using the relative distance from the detected lines to the already detected beacons and goals, and some simple rules. For example, the yellow goal line is the nearest parallel line to the yellow goal.

(iv) *Camera Self-Localization.* This feature is not required in all applications, but in cases where it is desired to obtain an exact description of the game (e.g., automatic generation of training data for learning complex behaviors or real game data to be used in simulations). The camera self-localization is implemented using the standard procedures employed in the Four-Legged league, which makes use of the landmarks information, lines information, and intrinsic parameters of the camera (see for example [26]).

3.3 Tracking

The *Tracking* module is in charge of tracking all moving objects: the ball and all visible robot players (up to 8). The implemented tracking system is performed using the mean shift algorithm [29] applied over the original image (not the segmented one). We use RGB color histograms as feature vectors (model) for mean shift, with each channel quantized to 32 bins. The feature vector is weighted using an Epanechnikov kernel (see [29]). For each detected and identified moving object a certain mean shift model is initialized.

As in [31], a Kalman Filter is employed for maintaining an actualized feature model for mean shift. The state vector of the Kalman filter is the weighted feature vector (RGB histogram) of the object under tracking. Thus, in each tracking process a

Kalman filter estimates the state of the corresponding mean shift model. This allows that longer tracking episodes can be maintained. Each time that the tracking of a moving object is lost, the corresponding object is detected and identified again.

Considering that the camera can also be mobile, it is important to track the lines. The lines have a second-order dynamic model whose state corresponds to the line inclination and position, which permits future line position prediction. The lines are tracked in a simple and robust way by generating small lateral scan lines that are perpendicular to each line prediction. Then for each line, the maximum intensity value in each lateral scan line is determined. Maxima points that are positioned over one of the two extremes of the lateral scan line are discarded. The remaining maxima points for each line are clustered into line segments. Line segments with less than 8 maxima points or with no overlap (over the line axis) with the previous line detection are discarded. Then, a least-square analysis of the surviving maxima gives the new line position estimation. If no line segment has more than 8 maxima points, the sizes (scale) of the perpendicular scan lines are enlarged and the process is repeated. If the scan lines are larger than a threshold (12 pixels in our implementation), the line tracking process fails and the line model is destroyed.

Using the described tracking processes, we are able to track in real time (30 fps) all game moving objects and the lines.

3.4 Motion and Action Analysis

This module is in charge of analyzing the game dynamics, determining the actions performed by the players, and the game relevant events. This analysis is carried out using the information of the static and moving detected objects, and the game rules (defined in [30]).

Most of the game situations can be analyzed using information about the position in the field of the ball and the robot players, the exact localization of the field lines and goal lines, and the identity of the last robot that touches the ball. For instance, for detecting a goal event, a ball crossing one of the two goal lines should be detected. The identity of the scoring robot is the identity of the last robot that touches the ball. Thus, using simple rules the following situations (definitions taken from [30]), can be analyzed:

- *Goal:* “A goal is achieved when the entire ball (not only the center of the ball) goes over the field-side edge of the goal line, i. e. the ball is completely inside the goal area”.
- *Robots kickoff positioning:* “In the ready state, the robots should walk to their legal kick-off positions. These positions are always located inside their own side of the field. Only two field players of the attacking team can walk to positions between the centerline and the middle of their side. They may put their leg(s) on the center circle line, but no leg may be inside the circle line. The other field players (one of attacking team, three of defending team) have to be located behind the middle of their side (none of their legs are allowed to be in front of the line connecting the centers of beacons), and must have at least two feet outside the penalty area. In contrast, the goal keeper must have at least two feet inside the penalty area”, and “The robots have a maximum of 30 seconds to reach their positions”.

- *Ball leaving the field:* "A ball is considered to have left the field when there is no part of the ball over the green field inside the boundary line".
- *Global Game Stuck:* "No robot touches the ball for 30 seconds".
- *Robot leaving the field:* "A robot that leaves the 4mx6m carpeted area will be removed for 30 seconds as per the Standard removal penalty".
- *Illegal defense:* "The vertical projection of the goalie to the goal line should not occupy more than 50 percent of the length of the goal mouth".

Moreover, in the case of the *Goal*, *Ball leaving the field* and *Robot leaving the field* situations, the identity of the protagonist robot (the scoring robot, the robot leaving the field or the robot that through out the ball) is determined (SIFT descriptors).

However, there are some other situations that are much harder to analyze, because it is required either to judge the intention of the players (e.g. robot obstruction) or to solve visual occlusions that difficult the determination of the relative position of the robot legs or the ball (e.g. ball holding). We are currently working towards a robust detection of these harder situations (see list bellow). For instance, using SIFT descriptors for determining which robots are looking to the ball (a similar idea was implemented in [28]), allows solving the *Robot obstruction* situation. The list of non-completely solved situations is (definitions taken from [30]):

- *Ball Holding:* "A robot which does not leave enough open space around the ball will be penalized as 'Ball Holding' if that situation continues more than 3 seconds or if the robot moves the ball over 50cm while continuing to hold the ball".
- *Goalie Pushing:* "When the goalie is in its own penalty area (2 feet on or inside line), no attacker may be in contact with the goalie for more than 3 seconds or be pushing the goalie indirectly through the ball for more than 3 seconds."
- *Field Player Pushing:* "Any robot pushing another robot for more than 3 seconds will be removed from play for 30 seconds as per the standard removal penalty", and "The closest robot (including the goal keeper) to the ball on each team, if it is within 2 dog-lengths of the ball, cannot be called for pushing"
- *Illegal defender:* "Having three legs inside the penalty area is the definition of being in the penalty area and that situation is not allowed for defending field players", but "This rule will be applied even if the goalie is outside of the penalty area, but not if an operational defender is pushed into the penalty area by an opponent."
- *Robot obstruction:* "When a robot that is not heading for the ball is actively, and intentionally, blocking the way to the ball for a robot of the other team, this is an obstruction"

Summarizing, for refereeing purposed we can automatically detect the following situations: correct *Robots kickoff positioning*, *Goal event*, *Ball leaving the field*, *Robot leaving the* field, *Global Game Stuck*, and *Illegal defense*. The game statistics that can be computed in our current implementation are: % of the time that the ball is in each half of the field, number of goals of each team and team player, number of direct kicks to the goal by each team and each team player, number of times that each team and each team player sent the ball out of the field, number of illegal defense events of each goalie, number of times that each team player leaves the field, number of global game stuck events, and time required for automatic kickoff positioning by each team.

4 Experimental Results

Experiments were carried out with the objective of characterizing the proposed system. Quantitative results are presented for the tracking of objects (lines, players and ball). Qualitative results are presented for the some events detection.

4.1 Object Tracking

Line tracking. For characterizing the line tracking subsystem, 19 video sequences were analyzed (30 fps, around 14.5 seconds each one, all together 8,313 frames). Lines to be tracked are initialized and the tracked. During the video analysis most of the lines are correctly tracked (see table 1). However, in some cases lines cannot be tracked, generally due to strong camera movements, and the line's model is destroyed. In other cases the tracking of some lines became confuse and other object (generally other lines) starts to be tracked. All these cases, including the situation when lines get out of the image as the camera moves are summarized in table 1.

Player tracking. For characterizing the player tracking subsystem, the same 19 video sequences used for line tracking analysis were employed. During the video analysis most of the players are correctly tracked (see table 1). However, in some cases players cannot be tracked, generally due to strong camera movements, and the player's model is destroyed. In other cases the tracking of some players became confuse and other object (generally other players) starts to be tracked. All these cases, including the situation when players move out of the image as the camera moves are summarized in table 1.

Ball tracking. Ball tracking is done using two sources of information: the mean shift information and (very fast) ball detection subsystem. These both information sources are fused using a Kalman. The same 19 video sequences were used for the ball tracking analysis In all the analyzed frames, the ball tracking has no lost tracking or incorrect tracking problems.

Table 1. Tracking statistics

	Line Object	Player Object
Total number of analyzed frames	8313	8313
Total number of initialized objects	83	38
Total number of objects which get out of the image	26	13
Total number of objects with lost tracking	9	3
Total number of objects with incorrect tracking	2	2

4.2 Event Detection

The event detection capabilities of the proposed system are still not well characterized, we just have qualitative results, which are very promissory. In figures 2 and 3 we show two exemplar video sequences where it can be seen the detections of a *Goal* event and a *Ball leaving the field* event, respectively. In these sequences the ball tracking window is shown in blue, while the players' tracking windows are shown in red. Every time a

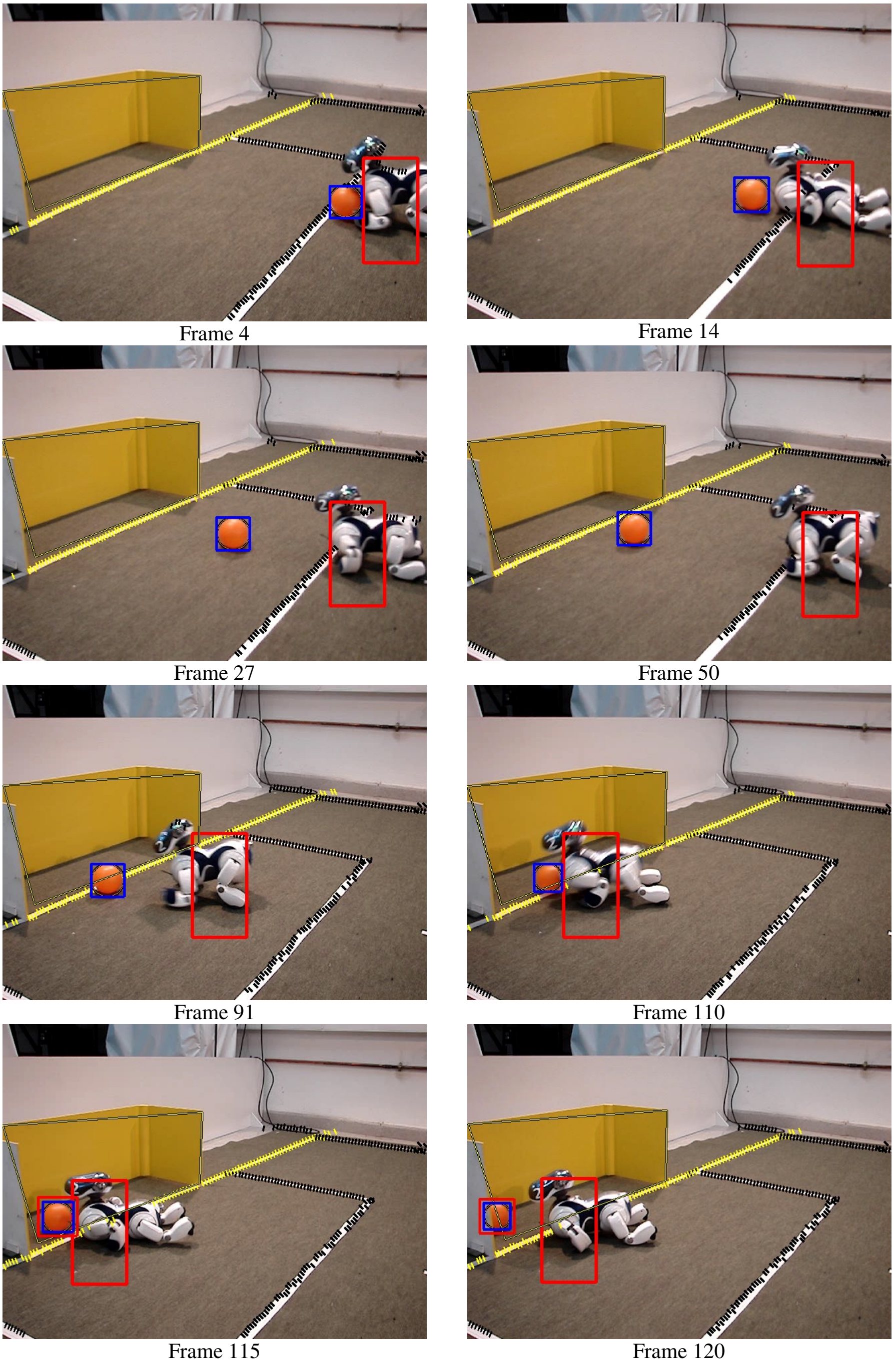
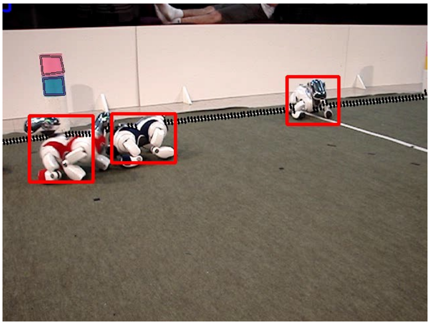
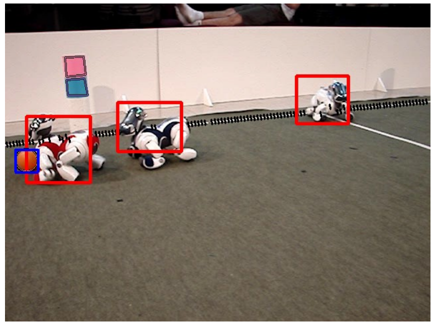
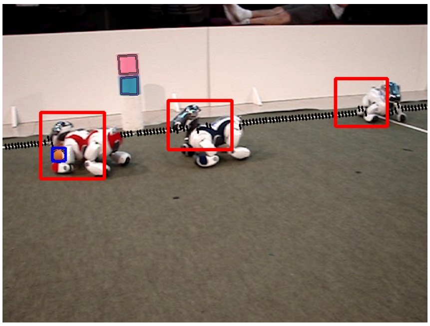
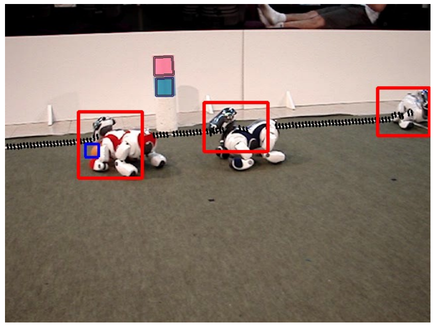


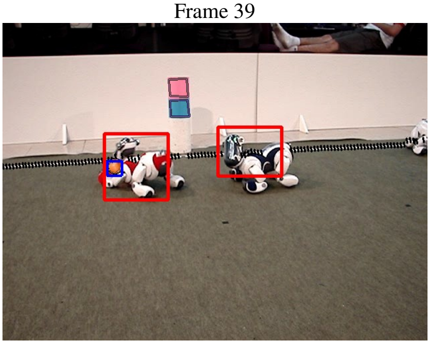
Fig. 2. Selected frames from a robot scoring sequence. Robot/ball tracking window in red/blue. The goal detection event (frame 115) is shown by a red window out of the ball blue window.

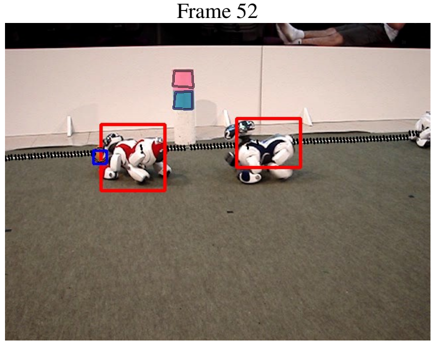

Frame 2

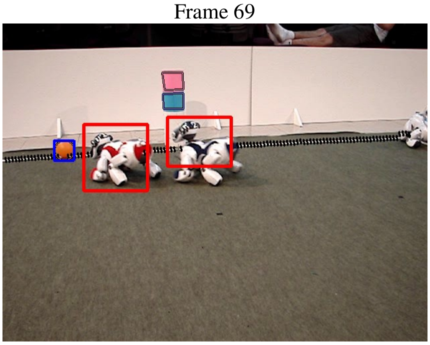

Frame 16


Frame 39


Frame 52


Frame 69


Frame 152


Frame 172

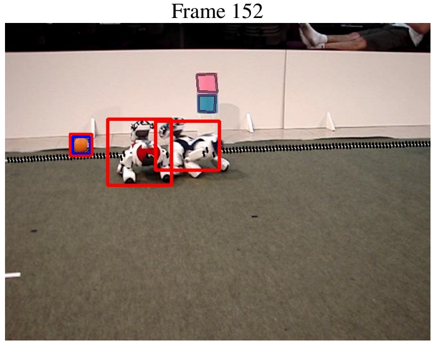

Frame 197

Fig. 3. Selected frames from a ball leaving the field sequence. Robot/ball tracking window in red/blue. The ball out of the field is marked by a red window out of the blue one (frame 197).

ball event occurs, a red window is displayed out of the ball's blue window. In the sequences are also shown the detected and identified lines; in figure 2, the yellow goal line in yellow and the field lines in white, and in figure 3 the field lines in white. It can also be seen the small lateral lines (perpendicular to each goal and field line) used for line tracking. Finally, the detected and identified landmarks are also indicated with boxes (yellow goal in figure 2 and pink-cyan beacon in figure 3).

5 Conclusions

In this paper was proposed an automated refereeing and analysis system for robot soccer. Although this system is originally developed for the Four-Legged league, it can be extended to other robot soccer contexts.

Currently, the developed system can detect and identify all Four-Legged defined field objects, and perform the tracking of the moving objects in real-time. However, not all defined situations can be robustly detected, mainly because the detection of some complex situations requires either to judge the intention of the players or to solve visual occlusions. In this sense the proposed system is still under development, but we think that the preliminary results show its potential.

We believe that developing a robust automated referee could be a joint initiative of the whole Four-Legged league. Using the framework here-proposed, other teams could develop and test their own analysis module. Moreover, in a near future the proposed automated referee could be integrated with the game controller. This could allow the automation of the game controlling process.

Another pending issue is the placement of the referee camera. We propose that this issue can be solved by using a cameraman who should continuously move the camera and focus it towards the game significant situations. In this way the whole game analysis and decision process could be automated, and humans will only need to move the referee camera o cameras, and to manipulate the robots in some specific situations (manual placement, robot pick-up, and removal of a penalized robot from the field of play). This is a necessary intermediate step towards having robot referees in a future.

References

1. Laue, T., Spiess, K., Röfer, T.: SimRobot - A General Physical Robot Simulator and Its Application in RoboCup. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, Springer, Heidelberg, 2006 (in press)
2. Zagal, J., Ruiz-del-Solar, J.: A Dynamically and Visually Realistic Simulator for the RoboCup Four Legged League. In: Nardi, D., Riedmiller, M., Sammut, C., Santos-Victor, J. (eds.) RoboCup 2004. LNCS (LNAI), vol. 3276, pp. 34–45. Springer, Heidelberg (2005)
3. Assfalg, J., Bertini, M., Colombo, C., Del Bimbo, A.: Semantic Annotation of Sports Videos. IEEE MultiMedia 9(2), 52–60 (2002)
4. Assfalg, J., Bertini, M., Colombo, C., Del Bimbo, A., Nunziati, W.: Semantic Annotation of soccer videos: automatic highlights identification. Computer Vision and Image Understanding 92(2-3), 285–305 (2003)

5. Babaguchi, N., Kawai, Y., Kitahashi, T.: Event Based Video Indexing by Intermodal Collaboration. IEEE Transactions on Multimedia 4(1), 68–75 (2002)
6. Bertini, M., Del Bimbo, A., Nunziati, W.: Soccer Video Highlight Prediction and Annotation in Real Time. In: Roli, F., Vitulano, S. (eds.) ICIAP 2005. LNCS, vol. 3617, pp. 638–644. Springer, Heidelberg (2005)
7. D'Orazio, T., Ancona, N., Cicirelli, G., Nitti, M.: A ball detection algorithm for real soccer image sequences. In: Proc. Int. Conf. Patt. Recog. – ICPR'02, Canada, (August 11-15, 2002)
8. Duan, L.Y., Xu, M., Chua, T.S., Tian, Q., Xu, C.S.: A mid-level representation framework for semantic sports video analysis. In: Proc. of ACM MM'03, Berkeley, USA, pp. 33–44. ACM Press, New York (November 2-8, 2003)
9. Ekin, A., Tekalp, A.M.: Automatic soccer video analysis and summarization. IEEE Trans. on Image Processing 12(7), 796–807 (2003)
10. Han, M., Hua, W., Xu, W., Gong, Y.H.: An integrated baseball digest system using maximum entropy method. In: Proc. of ACM MM'02, pp. 347–350. ACM Press, New York (2002)
11. Li, B., Sezan, M.I.: Event detection and summarization in American football broadcast video. In: Proc. SPIE Conf. on Storage and Retrieval for Media Databases, vol. 4676, pp. 202–213 (2002)
12. Nepal, S., Srinivasan, U., Reynolds, G.: Automatic detection of 'Goal' segments in basketball videos. In: Proc. ACM MM'01, Ottawa, Canada, pp. 261–269. ACM Press, New York (2001)
13. Pingali, G., Jean, Y., Carlbom, I.: Real time tracking for enhanced tennis broadcasts. In: Proc. IEEE Conf. Comp. Vision and Patt. Rec. – CVPR'98, pp. 260–265. IEEE Computer Society Press, Los Alamitos (1998)
14. Pingali, G., Opalach, A., Jean, Y.: Ball tracking and virtual replays for innovative tennis broadcasts. In: Proc. Int. Conf. Patt. Recog. – ICPR'00, Barcelona, Spain, pp. 4152–4146 (2000)
15. Rui, Y., Gupta, A., Acero, A.: Automatically extracting highlights for TV Baseball programs. In: Rui, Y., Gupta, A., Acero, A. (eds.) Proc. of ACM MM'00, pp. 105–115. ACM Press, New York (2000)
16. Sudhir, G., Lee, J., Jain, A.K.: Automatic classification of tennis video for high-level contentbased retrieval. In: Proc. of IEEE Int. Workshop on Content-based Access of Image and Video Database, pp. 81–90. IEEE Computer Society Press, Los Alamitos (1998)
17. Tovinkere, V., Qian, R.J.: Detecting semantic events in soccer games: Towards a complete solution. In: Proc. ICME'01, pp. 1040–1043 (2001)
18. Wan, K., Yan, X., Yu, X., Xu, C.S.: Real-time goalmouth detection in MPEG soccer video. In: Proc. of ACM MM'03, Berkeley, USA, pp. 311–314 (2003)
19. Wan, K., Yan, X., Yu, X., Xu, C.S.: Robust goalmouth detection for virtual content insertion. In: Proc. of ACM MM'03, Berkeley, USA, pp. 468–469. ACM Press, New York (2003)
20. Wang, J.R., Paramesh, N.: A scheme for archiving and browsing tennis video clips. In: Proc. of IEEE Pacific-Rim Conf. on Multimedia - PCM'03, Singapore, IEEE Computer Society Press, Los Alamitos (2003)
21. Wang, J.R., Parameswaran, N.: Survey of Sports Video Analysis: Research Issues and Applications. In: Conferences in Research and Practice in Information Technology VIP'03, Sidney, vol. 36, pp. 87–90 (2004)
22. Yow, D., Yeo, B.L., Yeung, M., Liu, B.: Analysis and presentation of soccer highlights from digital video. In: Proc. ACCV'95 (1995)

23. Yu, X., Xu, C.S., Leong, H.W., Tian, Q., Tang, Q., Wan, K.W.: Trajectory-based ball detection and tracking with applications to semantic analysis of broadcast soccer video. In: Proc. of ACM MM'03, Berkeley, USA, pp. 11–20. ACM Press, New York (2003)
24. http://www.hawkeyeinnovations.co.uk/
25. http://www.questec.com/
26. Röfer, T., et al.: German Team, Technical Report, RoboCup 2005, Four-legged league (February 2006) Available on http://www.germanteam.org/GT2005.pdf
27. Quinlan, M.J., et al.: The NUbots Team Report, RoboCup 2005, Four-legged league (February 2006), Available on http://www.robots .newcastle.edu.au/ publications/ NUbot FinalReport2005.pdf
28. Loncomilla, P., Ruiz-del-Solar, J.: Gaze Direction Determination of Opponents and Teammates in Robot Soccer. In: Bredenfeld, A., Jacoff, A., Noda, I., Takahashi, Y. (eds.) RoboCup 2005. LNCS (LNAI), vol. 4020, Springer, Heidelberg (2006)
29. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-Based Object Tracking. IEEE Trans. on Pattern Anal. Machine Intell. 25(5), 564–575 (2003)
30. RoboCup 2006 Four-legged Leagues official rules: Available on February 2006 in http://www.tzi.de/4legged/pub/Website/Downloads/Rules2006.pdf
31. Peng, N.S., Yang, J., Liu, Z.: Mean shift blob tracking with kernel histogram filtering and hypothesis testing. Pattern Recognition Letters 26, 605–614 (2005)
32. Lowe, D.G.: Distinctive Image Features from Scale-Invariant Keypoints. Int. Journal of Computer Vision 60(2), 91–110 (2004)