A Team Play Analysis Support System for Soccer Games

G. Lashkia, N. Ochimachi, E. Nishida and S. Hisamoto
Department of information and Computer Engineering
Okayama University of Science
1-1 Ridai-cho, Okayama, 700-0005 Japan

Abstract

In this paper, we present a computer software tool that can assist in soccer game scene analysis. The proposed system processes video images taken from a fixed positioned camera and creates 2D and 3D animations of the game. It uses a probability model for background and players in purpose to detect players' positions. Camera coordinates of the players are translated into field model coordinates, enabling to display the soccer game animations. The system makes possible to analyze motion by 3D display from any point on the field. It also can display the 'dominant areas' of teams, which are the areas that the team players can arrive first. The effectiveness of the proposed system is shown by experiments on soccer game scenes.

1. Introduction

Recently, developments in the field of sport have increased the need for scientific analysis of game strategies. In many situations the strategies are studied and analyzed manually, and require a lot of time and labor power. This kind of analysis is also biased, since it depends on human factor. Recently, automatic analysis of sport games has become a focus of studies for many researchers [1,2]. One of the challenging applications for automatic analysis of sport scenes is a soccer game. The soccer domain is one of the difficult areas in image processing, since a great number of problems have to be managed here, such as occlusions, variation of illumination, objects similar to the ball, and real time processing. Most related papers [3,4,5,6,7] are concentrated on the players and ball detection, and are based on automated tracking. Various methods were proposed for tracking, however, unexpected motions and occlusions, which often occur in soccer games are very difficult to track, and a reliable player tracker is still a hot topic in the soccer image processing research. In our proposed system we choose a different and simpler approach for players detection. Since the purpose of the proposed system is to analyze a team play, we do not intend to track individual players. Using color information and constructing a probability model for background and players we detect the players in each scene. To tolerate illumination changes the parameters of the model are updated automatically. Robustness against occlusions is achieved by using a 'minimal displacement' principle. When the number of players decreases due to occlusions, a mapping of players from the previous scene to the current scene is considered. The mapping that results in minimal displacement of players is used to determine occlusions. For the ball detection we used a special filter that collects high votes on white rounded areas. These areas are considered as candidate areas. The ball is then discriminated between these candidates by taking in account a continuous movement of the ball on consecutive frames of soccer game images.

Since we understand that by using current image processing methods and equipment it is difficult to precisely detect all objects on the field, we developed a correction program, that makes easy to correct all incorrectly recognized data. After all camera coordinates of players and the ball are determined, they are translated into field model coordinates. Field model data can be used for various analysis. In this study we only concentrate on dominant areas of teams and on the possibility to display a game from any point on the field.

For the system to be a practical aid to users, it is vital that all components are integrated into a single software application with an easy-to-use graphical interface. Such a system was developed using the programming language C++ and OpenGL, and the prototype version of the system can be obtained from the authors on request.

2. Detecting background and foreground objects

We consider the values of a pixel intensity over time as a pixel process [9]. The pixel process is a time series of red, green and blue pixel values. A mixture of three Gaussian distributions is used to model a pixel process, one for background and two for competing teams. Every new pixel value is checked against the existing Gaussian distributions until a match is found. As we use an intensity-based background model, any changes in illumination can cause false positives. A new pixel value will, in general, be represented by one of the components of the mixture model and used to update the model periodically to adapt to lighting changes. The parameters of the distribution, which matches the new observation are updated. The update makes the model robust to illumination changes and it is a well known method for the background scene modeling and maintenance [8,9].

A few steps are required to learn initial Gaussian distributions parameters. First, a pixelwise median filter over time is applied to several seconds of video to distinguish moving pixels from stationary pixels. Next, only those stationary pixels are processed to construct the initial background model. Pixel values that do not fit to the background distribution are considered foreground, and clustering is used to determine pixels with two dominant colors. We expect that these colors correspond to teams' uniforms (shirts) colors. Pixels that correspond to two dominant colors are processed to construct the initial teams models. For computational reasons in this study we assume that the red, green and blue pixel values are independent. While this is certainly not the case, the assumption allows us to avoid a costly matrix inversion at the expense of some accuracy. The parameters of each distribution, which are mean and standard deviation, are updated in the similar way as described in [9].

From the initially constructed background image the field lines are detected using Hough transform. The cross points of the field lines are used to transform the player's camera image coordinates P(x_c , y_c) (Fig 1) into the field model (Fig 2) coordinates P'(x_f , y_f). The transformation is performed using a projective invariant, the so-called cross ratio [10]. The X coordinate of P' is calculated from

AP:AB=A'P':A'B',

and Y coordinate of P' is calculated from

$$\frac{AD}{CD} : \frac{AE}{CE} = \frac{A'D}{C'D} : \frac{A'E}{C'E}.$$

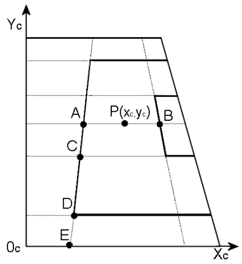

Fig. 1 Image plane.

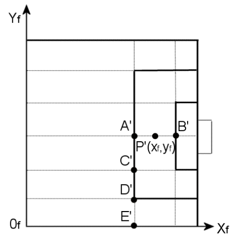

Fig. 2 Field plane.

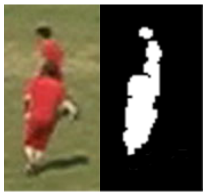

Fig.3 Overlapped players.

Fig.4 Screenshot of the correction program.

For each frame of a video sequence the detection of players coordinates is performed based on the set of pixels that match the team's model. Players are detected by a three stage process: noise cleaning, morphological filtering and connected component analysis. This image processing can't separate overlapped players. A result of players' detection in the case of overlap is shown in Fig. 3. To solve the players occlusion problem we use the minimal displacement principle. A monitoring area is set on the field to monitor incoming and outgoing players. If the number of players in the monitoring area decreases and no player is moved out of the area, the occlusion is detected. In this case, a mapping of players positions from the previous scene to the current scene is considered. The mapping that results in minimal displacement of players is used to determine overlapping positions.

The identification of the ball in soccer images is very important in order to obtain a comprehensive analysis of the game. Circular objects occur frequently in many natural and manmade scenes and a number of techniques [6] including circle Hough transform have been developed in the last decade for circle detection. However, most of methods are time consuming. In our system we employed an original ball detection filter, which is based on voting process. For each pixel of the foreground objects a direction of the intensity gradient is calculated and voting is performed on white pixels along. Rounded white objects collect high votes. Regions with high votes are considered as ball candidates. A few candidate regions could be detected. To deal with noise regions, we track each candidate by searching a neighboring area in subsequent frames. If tracking fails the candidate region is discarded. Unfortunately, the current version of the soccer game scene analyzer uses only one camera. This limits it to process only a grounded ball. Although it is possible to adapt the system to a stereo vision, the stereo vision equipment was not available in the development stage, and it will be a focus of a future research.

The back end of the players and ball detection module is the correction program. The purpose of it is to correct falsely detected coordinates. A screenshot showing the main window of the correction program is given in Fig. 4. At the top and the left of the window there are a set of buttons and tools available for the user. The correction program displays each frame of video sequence with the detected foreground objects on it. Detected objects are overdrawn by computer generated graphics, and therefore it is easy to see and correct incorrectly recognized data. When the position of an object has been mistaken, it can be corrected by simply clicking the right place with the mouse. The object selected for correction is displayed as a bold figure. When all necessary corrections are made, the process is passed to the animation program.

3. Motion analysis

The animation program displays 2D and 3D images of scenes based on the detected coordinates of players and the ball (Fig. 5). The system makes possible to view 3D motions from any point of the field. Fig. 6 shows the same scene from different viewpoints. A viewpoint can be selected by simply clicking the mouse on the 2D animation window. The dark dot on the 2D animation window in Fig. 5 represents the viewing point. The view direction in the 3D animation window can be easily changed by pressing a proper key.

Various kinds of tactical analysis can be performed using collected data. However, in this study we only consider dominant areas. The dominant area of

Fig.5 Screenshot of the animation program.

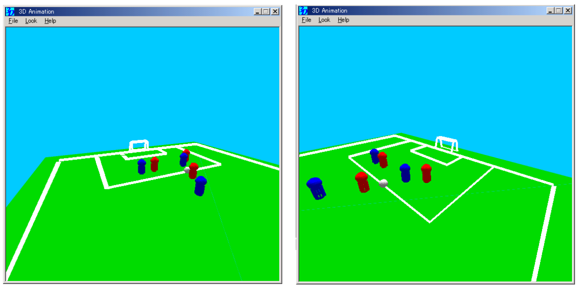

Fig.6 A scene displayed from different viewpoints.

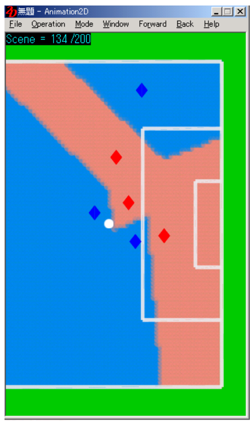

Fig.7 Screenshot of the dominant areas window.

a team is the closest region to the team players. In other words, it is the decision surface induced by the Nearest Neighbor classifier when the players are considered as training examples of a two class problem. The dominant area is very useful information for analysis of defense and offence tactics. Defenders try to cover bigger areas of the defense area, while offenders try to make gaps on it. The window displaying dominant areas is started from the menu bar of the 2D viewer. A screenshot showing a window of dominant areas is given in Fig.7.

4. Experimental Results

The effectiveness of the proposed system is shown by experiments on real soccer training match scenes. Two teams of blue and red uniforms with three players in each were considered. Fig. 8 shows an image of the input stream. We experimented with over a 10 minutes long soccer game goal scenes. Images were taken by a conventional digital video camera, which was then connected to the computer. The system showed its robustness to illumination changes and occlusions. The detection rate of players was 84.7%, and the detection rate of the ball was 84.8%. The necessary corrections were completed in a few minutes and the system was ready to run animations. 3D animations of ball scenes were especially interesting for soccer players. The game can be viewed from different points and this results in better understanding of the player's wrong and right decisions. Dominant areas were helpful for team play analysis. Gaps in defense and better offensive tactics were clearly seen.

4. Conclusion

In this paper we have described a software tool that can assist in soccer game scene analysis. The proposed system takes soccer scene images, detects players and the ball, and creates 2D and 3D animations of the scenes. It is robust to illumination changes and occlusions. It can be used for team play analysis and goal scenes display in soccer news coverage TV programs. The proposed system however, has some limitations. It does not integrate personal abilities of players, and its current version can deal only with the grounded ball. Future work will be focused on detection of 3D coordinates of the ball using a stereo vision, and on improvement of players' detection procedure by incorporating multiple cameras.

Fig.8 An input image and its 2D representation.

References

[1] A. Hayashi, R. Nakashima, T. Kanbara and N. Suematsu, "Multi-object pattern classification for visual surveillance and sport video retrieval", Proc. of Vision Interface, 2002.

[2] M. Naemura, A. Fukuda, Y. Mizutani, Y. Izumi, Y. Tanaka and K. Enami, "Morphological segmentation of sport scenes using color information", IEEE Trans. On Broadcasting, 46(3), 181-188, 2000.

[3] S. Sudo and S. Ozawa, "Scene analysis of soccer game", Proc. of QCAV'99, 1999.

[4] T. Misu, M. Naemura, W. Zheng, Y. Izumi, K. Fukui, "Robust tracking of soccer players based on data fusion", Proc. of ICPR'02, 2002.

[5] A. Yamada, Y. Shirai and J. Miura, "Tracking players and a ball in video image sequence and estimating camera parameters for 3D interpretation of soccer games", Proc. of ICPR'02, 2002.

[6] T. D'Orazio, N. Ancona, G. Cicirelli and M. Nitti, "A ball detection algorithm for real soccer image sequences", Proc. of ICPR'02, 2002.

[7] S. Lefevre, C. Fluck, B. Maillard and N. Vincent, "A fast snake-based method to track football players", Proc. of MVA, 501-504, 2000.

[8] I. Haritaoglu, D. Harwood, L. S. Davis, "A Fast Background Scene Modeling and Maintenance for Outdoor Surveillance", Proc. of ICPR'00, 2000.

[9] C. Stauffer, W. Grimson, "Adaptive background mixture models for real-time tracking. in Proceedings", Proc. of CVPR'99, 1999.

[10] R. Duda, P. Hart, Pattern classification and Scene analysis, John Wiley & Sons, 1973.