Multi-Camera Analysis of Soccer Sequences

Chris Poppe, Steven Verstockt, Sarah De Bruyne, Rik Van de Walle

2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance, pp. 26-31, 2010

http://www.computer.org/portal/web/csdl/doi/10.1109/AVSS.2010.64

Chris Poppe, Sarah De Bruyne, Steven Verstockt, Rik Van de Walle (2010). Multi-Camera Analysis of Soccer Sequences. *Proceedings of 2010 7th IEEE International Conference on Advanced Video and Signal Based Surveillance* 26-31. http://doi.ieeecomputersociety.org/10.1109/AVSS.2010.64

Multi-Camera Analysis of Soccer Sequences

Chris Poppe
chris.poppe@ugent.be

Sarah De Bruyne
sarah.debruyne@ugent.be

Steven Verstockt
steven.verstockt@ugent.be

Rik Van de Walle
rik.vandewalle@ugent.be

Multimedia Lab
Department of Electronics and Information Systems
Ghent University - IBBT
Gaston Crommenlaan 8 bus 201,
B-9050 Ledeberg-Ghent, Belgium

Abstract

The automatic detection of meaningful phases in a soccer game depends on the accurate localization of players and the ball at each moment. However, the automatic analysis of soccer sequences is a challenging task due to the presence of fast moving multiple objects. For this purpose, we present a multi-camera analysis system that yields the position of the ball and players on a common ground plane. The detection in each camera is based on a code-book algorithm and different features are used to classify the detected blobs. The detection results of each camera are transformed using homography to a virtual top-view of the playing field. Within this virtual top-view we merge trajectory information of the different cameras allowing to refine the found positions. In this paper, we evaluate the system on a public SOCCER dataset and end with a discussion of possible improvements of the dataset.

1. Introduction

Nowadays, video sequences are used to analyse sports games in order to improve the performance of a team. When analyzing several sequences a major effort is put in the selection of interesting game phases. Automatic analysis of the video sequences to yield and classify interesting game phases would provide a solution to this cumbersome task. In this paper, we will focus on soccer sequences to perform automated analysis. In broadcast video, a director chooses the view point (and corresponding camera) at each point. For analysis purposes this is not beneficial, so we focus on the original video feeds (so before editing has been applied). Typically, static cameras are positioned around the field capturing the entire game.

The automatic analysis of soccer games is not trivial due to the presence of fast moving occluding objects. In this paper, we present a multi-camera soccer analysis system. In each camera an initial moving object detection is applied using a code-book algorithm. This allows us to detect the players and ball visible in one camera. This detection is not totally accurate (due to occlusions, background noise or clutter and shadows), however for event detection it is generally not needed to get pixel-accurate detections. The position of the objects viewed by the different cameras is consequently projected on a common ground plane which can be seen as a virtual top-view of an entire soccer field. The information of the different cameras is consequently merged to find the real-world positions of the objects.

In the next section, we elaborate on related work in analysis of soccer sequences. Subsequently, Sect. 3 discusses our proposed system that is focused on detecting events in soccer sequences. The system is evaluated in Sect. 4 and some discussions are given in Sect. 5. Finally, concluding remarks are drawn in Sect. 6.

2. Related Work

Automated sports analysis using computer vision is a broad research domain [13]. When looking into soccer analysis two main directions can be found. The first group of systems analyse the broadcasted video. The second category uses one or more original video sequences that represent the entire soccer game.

D'Orazio et al. have shown the feasibility of the use of circular Hough transforms for ball detection [1]. Lang et al. analyse broadcast soccer video for the detection of the ball [6]. However, the presented results only correspond to specific parts or shot of a broadcast soccer video. More recently, Pallavi et al. focus on the detection of the ball in broadcast soccer video [9]. They first classify the sequence in medium and long shots and apply Hough transform and additional techniques to detect the ball. Zhu et al. also focus on broadcast video to analysis the tactics of a team during a goal event [14]. For the detection of the goal they combine both web-casting text as video analysis. Hua-Yong and Tingting search for semantic events in broadcast soccer video by integrating visual, auditory, and text features [3].

The use of planar homography has shown good results for localization of objects viewed by different cameras. Park and Trivedi have used the homography transform to monitor crowds [10]. Multi-view data is combined to track people on multiple scene planes in [4]. In our previous work, we applied homography transforms to localize objects in compressed videos [12].

Ren et al. present a multi-view analysis system for soccer video in which the focus lies on detection of ball and players [11]. They perform detection and tracking in single view and merge the results on a ground plane. However, much processing effort is needed for the background subtraction (using Gaussian mixture models) and the accurate single view tracking (due to occlusions, merges and splits). The detected objects are projected entirely on the common ground plane to get the accurate position. 3-D positions of the ball are detected. Note that, the actual height of the ball is not necesary for detecting events. If the detection of the ball does not correspond in the different views, it can be assumed that it is not touching the ground plane. Leo et al. apply neural networks to analyse the player actions in soccer games [7]. They combine detections in single camera views on a virtual top-view using homography. To find the accurate player positions they use the mid-point of the line connecting the different projections. The ball is tracked in the top-view by using straight lines. Our system is mostly related to the two latter works, however we want to avoid heavy processing on the single camera views and show that simple processing is sufficient for object localization and event detection.

3. System Overview

The proposed system performs some processing on the individual camera feeds before these are merged onto the virtual top-view.

3.1. Object Detection

In a first step, moving object detection is performed using a code-book method [5]. For each pixel in the image a code-book is created consisting of RGB values, minimum and maximum brightness, the frequency, first and last occurence time and the maximum period in which the code word has not occured. New pixel values are compared with these code words, if no match can be found the pixel is assumed to be foreground. Considering the static appearance of the soccer field, the code-book is only updated during a training fase. Although this allows to obtain higher speeds, global illumination changes might have a big influence. However these are easy to be detected since the changes will manifest itself on the entire field. As such, if too much foreground is detected the code-book is updated again. The object detection gives for each pixel in the image a classification between foreground and background. These results are morphologically filtered (closing after opening) to get rid of outliers. This is followed by a connected component analysis that allows creating individual blobs and bounding boxes are created. An example output on an image of the SOCCER dataset [2] is shown in figure 1. This dataset holds sequences of six static cameras of a soccer game.

To classify the objects we use color histograms of templates for each class. The different classes consist of *team 1, team 2, goalie team 1, goalie team 2, and referee*. Figure 2 shows the different templates. We take the center part of the detected blobs and calculate color histograms. These are compared, after normalization, with the color histograms of the templates using the Bhattacharyya distance. Automated clustering could be used, however this would increase the complexity of learning and produce more uncertain results. Selecting the templates can be very straightforward, (e.g., user input can be asked to classify the bounding boxes in the first frame).

3.2. Ball Detection

A second step is to detect the ball in the images. For this purpose the segmentation in foreground and background regions is used. The foreground regions are applied on the input image as a mask. As such, the background is represented by black pixels, the foreground is represented by the original pixels. The foreground image is first Gaussian smoothed and thresholded, lastly a canny edge detection is used to create an edge image. On this image a Hough transform is performed to detect candidate pixels that correspond with circles. Note that we only consider those pixels that were not marked as background pixels. Between the frames temporal information is used by creating a search window around the last found position of the ball. Only in this window, and only in the corresponding foreground blobs, the new ball location is searched. This allows to prevent that the entire frame has to be searched for the ball location, avoiding costly processing. To overcome propagating errors due to a misdetection of the ball, every 10 frames the entire foreground frame is searched for the best ball candidate. When the player is colliding with the ball the detection drops, however we can exploit this information. We denote

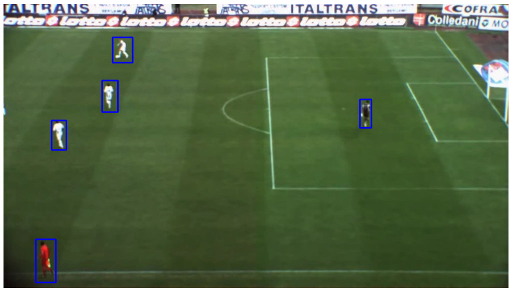

Figure 1. Object detection on single camera view.

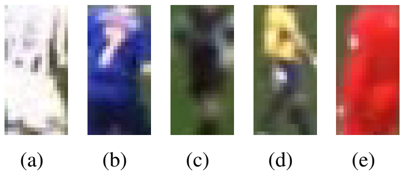

Figure 2. Templates for players of team 1, team 2, goaly of team 1, team 2 and the referee, respectively.

which player collides with the ball and extend the search window to include the bounding box of the player. As such, the ball can be detected when the player passes the ball. Finally, the detected circles in each frame are evaluated by comparing the average color in the circle with the average color of a template of the ball. This allows to yield a best match for each frame within each camera view.

3.3. Integration

The SOCCER dataset contains calibration data that can be used to calibrate the cameras. We have developed software to perform calibration of the images by asking for user input. This allows a user to select points in a camera view and to select corresponding points on a synthetic top-view playing field. This top-view is created using regular sizes of soccer playing fields. The according homography matrices are consequently automatically calculated. From experiments with our system we noticed that it is not needed to make these calibration images as present in the SOCCER dataset. The characteristics of the playing field, e.g., the corners, lines, and circles can be used for calibration. Future work consists of making this more automated by detecting these featurs automatically before calibrating the cameras.

Figure 3 shows the annotated image from the SOCCER dataset provided for calibration of the cameras. To show the correctness of our transformation we transformed this image to the top-view using the homography matrices calculated by our software, based on user input. As shown in Figure 4 the different points are correctly positioned on the top-view.

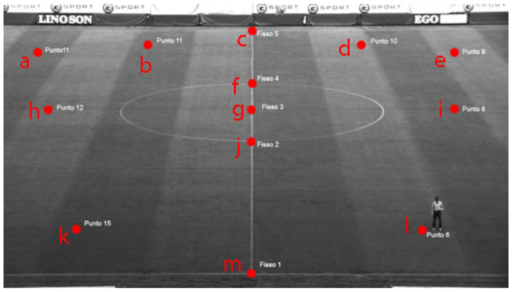

Figure 3. Input image with calibration points, part of the dataset.

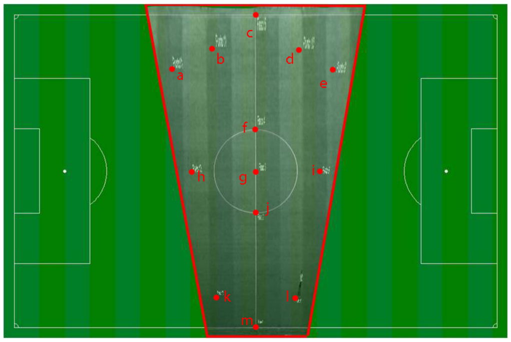

Figure 4. Effect of the homography transformation. As can be seen the different points are well-situated.

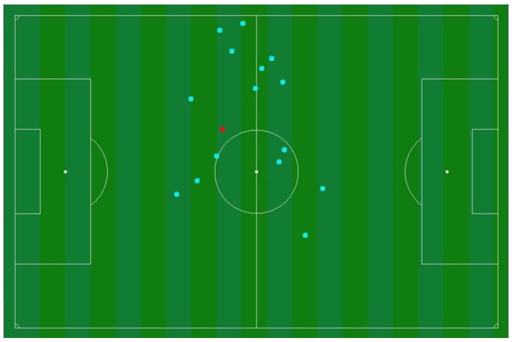

Figure 5. Positions in the top-view of players and ball detected in a single camera view.

The output of the analysis on the individual cameras is stored as XML-files. This metadata contains for each frame the projected location of the players and ball, the class of the object, and the amount of votes (from the hough transform) for the ball. For the players, the lowest point (y-axis) in the centre (x-axis) of the bounding box in the 2D view is transformed with the homography matrices. Figure 5 shows the positions of the players and ball as seen by a single camera. An example of the created XML-file is given in Listing1. Note that we could have used the metadata scheme of ViPER (Video Performance Evaluation Resource) a system for evaluating video analysis algorithms [8], as was also used in the SOCCER dataset. However, ViPER would arrange the metadata according to players, while in our metadata-listing the ordening is by frame. This allows to faster retrieve the objects that are present in a specific frame, which is needed during the integration of the different camera views.

```
<root>
  <numberOfFrames> 2811 </numberOfFrames>
  <frame>
    <ball>
      <x> 964.089 </x>
      <y> 157.628 </y>
      <votes> 17 </votes>
    </ball>
    <player>
      <x> 609.201 </x>
      <y> 118.749 </y>
      <team> 1 </team>
    </player>
    ...
  </frame>
  ...
</ root>
```

Listing 1. Example of metadata in XML format. It expresses for each frame the location of players and ball.

The different XML-files are parsed during the integration. The different positions of the players and ball make up a *scene*. Within this scene, different camera views result in different positions for the players and balls. For each of the camera views, the trajectories of the objects are determined. For consecutive frames the objects are added to a trajectory based on proximity. If an object is within 15 pixels from the last seen object (a buffer of 10 frames is used) of a trajectory, it is added to this trajectory. At this point, we have one point for each detected object in a camera view, which makes this a fast process. Using additional information like object size, average colors and so on is not necessary, since this info is represented by the class of the object. Due to classification errors, the class of the object might temporarily be wrong. Therefore, the trajectories for the objects are analyzed. The dominant class over time is used as the actual class, so outliers can be removed. Secondly, the trajectories of the different camera views are merged to create the final trajectory of an object. The single-view trajectories are compared based on the average distance of the intermediate positions. If this is smaller than a predefined threshold, a final trajectory is created by averaging the positions of the intermediate points. If different trajectories exist for the ball (each camera view finds the best ball candidate), trajectories with a low amount of votes are discarded. If multiple trajectories still exist, the trajectory with the most votes is chosen as the final ball trajectory.

4. Evaluation

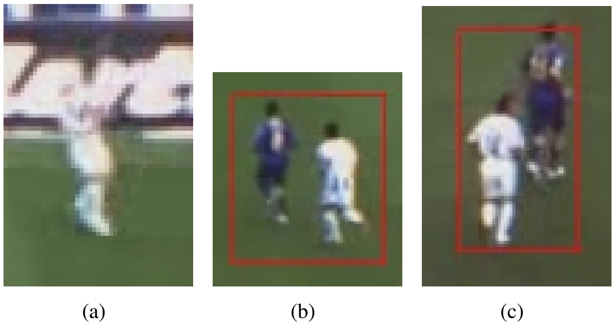
(a) (b) (c)

Figure 6. Examples of missed detection (a) and merged detections (b) and (c).

To evaluate the system we calculate precision and recall values for the positions of the players on the top-view. For this purpose, we transformed the ground-truth annotation of the SOCCER dataset using the homography matrices. Next, for each position detected by the system we search for the closest position in the ground-truth. If the distance between these is less than 10 pixels we count the position as a *true positive*. If no correspondence is found it is a *false positive*. A *false negative* is counted for each ground-truth object that was not detected. This way we can calculate the precision (the ratio of positions that our system outputs which actually correspond to real objects):

$$precision = \frac{TruePositives}{TruePositives + FalsePositives}. \quad (1)$$

The recall is the ratio of real objects that our system succesfully detected:

$$recall = \frac{TruePositives}{TruePositives + FalseNegatives}. \quad (2)$$

Table 1 shows the averaged precision and recall values for different camera views. Note that the SOCCER dataset contains sequences of cameras which are positioned in pairs facing each other. As a consequence, only within the camera pairs the views will overlap. We evaluate two camera pairs (camera 3 and 4, and camera 5 and 6) and present precision and recall for the individual camera views and their combinations. Note that the recall of the individual camera views is low due to occlusions of players. The background subtraction that we use delivers bounding boxes for each blob. As such, in single camera view we cannot distinguish different players when these are very close. The result is that only one position is calculated, while the ground-truth yields two or more positions. This influences the recall. Additionally, when two or more players occlude each other, the center of the resulting blob can be very distant from the centers of the represented players. This causes some false positives which makes the precision go down. Figure 6 shows typical examples of hard situations for the player detections.

Sequence	precision	recall
Film Role-0 ID-3	94.09	76.98
Film Role-0 ID-4	95.07	72.17
ID-3 & ID-4	93.49	86.77
Film Role-0 ID-5	89.49	70.67
Film Role-0 ID-6	89.6	68.58
ID-5 & ID-6	88.6	90.29

Table 1. Precision and recall values for player positioning.

Sequence	precision	recall
Film Role-0 ID-3	18.66	18.66
Film Role-0 ID-4	14.42	14.42
ID-3 & ID-4	21.08	21.08
Film Role-0 ID-5	19.07	19.07
Film Role-0 ID-6	20.55	20.55
ID-5 & ID-6	25.42	25.42

Table 2. Precision and recall values for ball positioning.

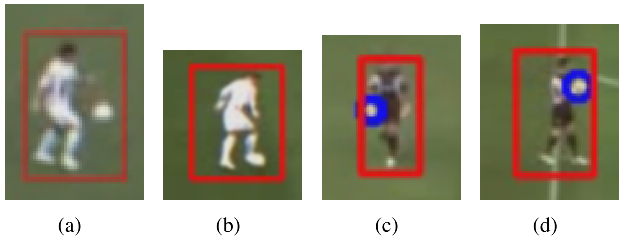
(a) (b) (c) (d)

Figure 7. Examples of missed balls (a) and (b), and wrongly detected balls (c) and (d).

Lastly, we include the results after combining the information of the two camera view-points. During the integration phase the positions of the players are refined. If the object detection in the single camera views resulted in incomplete detections, this results in positions on the top-view that are not accurate. Moreover, the detection of objects has a higher chance of failing when the objects are further away from the camera. This is due to the smaller size of the objects and the fact that a cluttered background is seen for these positions. Indeed, for regions close to the camera the background consists of the static playing field, the regions further away will have a cluttered background consisting of seating places, billboards, and so on. By combining the information of both cameras, these errors are removed since the players are close to at least one of the cameras. Additionally, if two players occlude each other, the point where they touch the ground-plane will be seen differently by the different cameras. This results, after integration, in the separate detection of both players which is beneficial for the recall. There is a slight drop in the precision due to the fact that sometimes the positions of a player, viewed by the two cameras, are to distinct. As a result, the system detects two players of which only one (at most) matches with the ground-truth.

Table 2 shows the results for the detection of the ball. The system searches for the best ball candidate in each camera view, as such, using all frames for evaluation would give very low precision values. Hence, for the evaluation we only take into account those frames in which a ball is actually present in the camera view. Since we restrict ourselves to the frames with a ball, the precision and recall values are the same. For each frame, the precision and recall is either zero or one, corresponding to a misdetected or correctly detected ball, respectively. To get the results in the table, the results are averaged over all frames that actually contain a ball. From the table it is clear that the ball detection is not perfect, approximately one of five balls is detected correctly. Examples of errors when detecting the ball are shown in Figure 7. As shown, the ball is hard to detect when it is close to a player. Additionally, the algorithm sometimes yields false detections due to parts of the image that ressemble the ball. Lastly, when the ball does not touch the ground plane, the projection results in mislocalization of the ball, which further decreases the results. During the integration, the trajectories of the candidate balls are created and merged (like with the players) to get the final position of the ball (on the top-view). The combination of the different camera views results in slightly better detection rates. However, it is clear that more sophisticated manners are needed to improve the ball detection, as presented in [11].

5. Discussions

Our system allows to retrieve the real-life positions of the ball and players in each frame on the playing field. Based on this information meaningfull events can be extracted. Simple events like an attack can be expressed by stating that a certain number of players of a team are running towards the goal of the opposite team. Additionally, detection of corner kicks and throw-ins do not require detailed knowledge of each player on the field. For evaluating such high-level events, it might be clear that a more high-level ground truth annotation would be beneficial for the SOCCER dataset. Current annotations in the dataset are restricted to position of ball and players in the individual camera views. However, ground truth annotation of the actual events would be of greater interest since these are what users are interested in. Research questions that arise are how to describe such an event. For instance, when does an attack start, when does an error occur, and what is a pass. Moreover, the dataset is rather restricted for event analysis due to the short sequences. It would be interesting to have different sequences that correspond to some interesting events (like goals, corner kicks, and so on). Sequences of different matches (different teams, weather conditions) would also be beneficial for analysis of soccer games. Lastly, ground-truth annotations of real-world positions of ball and players would be interesting.

6. Conclusions

In this paper we proposed a multi-camera video analysis system for soccer sequences. We assume that the playing field is viewed by different static cameras. On each camera, object detection and classification occurs. Players are detected using code-book background subtraction algorithm. The ball is detected by applying Hough transformations on the detected foreground image. The information of the objects in different camera view-points is combined by projection on a synthetic top-view playing field. Consequently, the different projections are merged to obtain the trajectories of the players and the ball. This representation allows to position the objects in the real world and to deduce meaningfull events. Lastly, we evaluated the system against a public available SOCCER dataset and present possible improvements for these sequences and annotations.

7. Acknowledgments

The research activities that have been described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT-Flanders), the Fund for Scientific Research-Flanders (FWOFlanders), and the European Union. Lastly, the authors would like to thank Steven Goossens for his valuable contributions.

References

[1] T. D’Orazio, N. Ancona, G. Cicirelli, and M. Nitti. A ball detection algorithm for real soccer image sequences. In *Proceedings of the International Conference on Pattern Recognition*, volume 1, pages 210–213, 2002. 1

[2] T. DOrazio, M.Leo, N. Mosca, and P. P.L.Mazzeo. A semi-automatic system for ground truth generation of soccer video sequences. In *Proceedings of the 6th IEEE International Conference on Advanced Video and Signal Surveillance*, 2009. 2

[3] L. Hua-Yong and H. Tingting. Semantic event mining in soccer video based on multiple feature fusion. In *Proceedings of the international conference on Information Technology and Computer Science*, pages 297–300, 2009. 2

[4] S. M. Khan and M. Shah. Tracking multiple occluding people by localizing on multiple scene planes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 505–519, 2009. 2

[5] K. Kim, T. H. Chalidabhongse, D. Harwood, and L. Davis. Real-time foreground-background segmentation using codebook model. *Real-Time Imaging*, 11:172 – 185, 2005. 2

[6] D. Lang, Y. Liu, Q. Huang, and W. Gao. A scheme for ball detection and tracking in broadcast soccer video. In *Proceedings of Advances in Mulitmedia Information Processing - PCM 2005*, pages 864–875, 2005. 1

[7] M. Leo, T. DOrazio, P. Spagnolo, P. L. Mazzeo, and A. Distante. Multi-view player action recognition in soccer games. In *Proceedings of MIRAGE 2009*, pages 46–57, 2009. 2

[8] V. Mariano, J. Min, J.-H. Park, R. Kasturi, D. Mihalcik, D. Doermann, and T. Drayer. Performance Evaluation of Object Detection Algorithms. In *Proceedings of the International Conference on Pattern Recognition*, pages 965–969, 2002. 3

[9] V. Pallavi, J. Mukherjee, A. K. Majumdar, and S. Sural. Ball detection from broadcast soccer videos using static and dynamic features. *Journal of Visual Communication and Image Representation*, 19(7):426 – 436, 2008. 2

[10] S. Park and M. Trivedi. Homography-based analysis of people and vehicle activities in crowded scenes. In *Proceedings of International Workshop on Applications of Computer Vision*, pages 51 – 56, 2007. 2

[11] J. Ren, M. Xu, J. Orwell, and A. Jones. Multi-camera video surveillance for real-time analysis and reconstruction of soccer games. *Journal of Machine Visions and Applications*, 2009. 2, 5

[12] S. Verstockt, S. D. Bruyne, C. Poppe, P. Lambert, and R. V. de Walle. Multi-view object localization in h.264/avc compressed domain. In *Proceedigns of the 6th International Conference on Advanced Video and Signal Based Surveillance*, pages 370–374, 2009. 2

[13] X. Yu and D. Farin. Current and emerging topics in sports video processing. In *Proceedings of International Conference on Multimedia and Expo*, pages 526–529, 2005. 1

[14] G. Zhu, Q. Huang, C. Xu, Y. Rui, S. Jiang, W. Gao, and H. Yao. Trajectory based event tactics analysis in broadcast sports video. In *Proceedings of the 15th international conference on Multimedia*, pages 58–76, 2007. 2