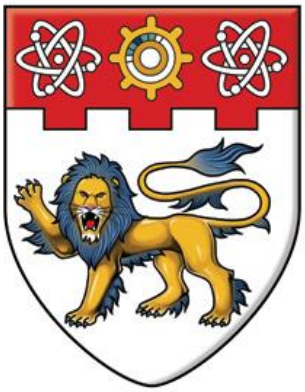

NANYANG TECHNOLOGICAL UNIVERSITY

Soccer Video Analysis

Submitted by: Muhd Haziq Bin Razali

Matriculation Number: U1322810E

Supervisor: Dr. Stefan Winkler

Co-supervisor: Asst/Prof Wang Gang

School of Electrical & Electronic Engineering

A final year project report presented to the Nanyang Technological University
In partial fulfilment of the requirements of the degree of
Bachelor of Engineering

2015

Abstract

In a soccer match, the ball is invariably the focus of attention. While there have been many works dedicated to object tracking, the tracking of soccer ball remains a challenge due to a lack of features, its relatively small size and the occlusions that occur during player-ball interaction. In this thesis, we present an automated real time tracking system for soccer videos utilizing multiple cameras for complete coverage of the soccer field. A background subtraction algorithm serves as the first stage in the object detection framework from which the detected objects are classified as either a ball or a player through a combination of multiple techniques. The player tracker is based on a Kalman filtering framework with an adaptive template for occlusion handling. The ball tracking algorithm takes advantage of prior knowledge of soccer for occlusion and event handling. The results from each camera are then registered onto a model of the soccer field for analysis in 3D. Experiments on the ISSIA soccer dataset show that the system is effective with promising precision and recall measures.

Acknowledgments

I would like to express my sincere gratitude to both Dr. Stefan Winkler and Professor Wang Gang for their continuous support during this project. Their patience, motivation, and immeasurable knowledge have helped steer me in the right direction. I could not have imagined having better advisors and mentors during this last leg of my undergraduate journey.

Contents

Chapter 1

Introduction

Video processing has found many applications in sports such as soccer. Being an incredibly competitive field, clubs around the globe are incorporating video analysis methods as training tools in the development of the team. In the context of team development, playbacks provide unparalleled coverage of key events, enabling teams to understand their own strengths and weaknesses, facilitating strategy development. Having said that, there exists an unmet need to automate and improve analysis of soccer videos as all related works require some sort of manual input to annotate important events and to perform statistical analysis.

In this thesis, drawing from previous work, we present an automated real time tracking system for soccer videos. The system utilizes multiple static cameras for complete coverage of the soccer field. The objectives are to develop robust methods for tracking players and ball across multiple cameras. In addition, since the ball is often flying across the air, a robust and efficient method to localize the ball in 3D must be investigated.

The organization of the remainder of this thesis is as follows. We start off by presenting a review of related works on multi and single camera ball tracking in chapter 2. Their approach will be summarized and their strengths and weaknesses discussed. In chapter 3, the outline of the system will be presented diagrammatically. Chapter 4 begins describing the system by covering the object detection framework. This contains a comprehensive description of the algorithms used to locate all objects of interest. Following that, the object tracking architecture will be delineated in chapter 5 where we explain the algorithms used and the occlusion handling techniques used. Given all the tracked objects from each camera, we must then localize the positions of the players and the ball with respect to world

coordinates and ensure that only valid correspondences are generated. This will be covered in chapter 6. Finally chapter 7 will present the performance of the designed tracker before we conclude in chapter 8.

Chapter 2

Related Works

In the soccer domain, various methods for the detection, tracking and localization of a soccer ball in 3D have been proposed. Ohno et als [1] multi camera setup incorporated frame differencing and trajectory mining to identify ball candidates in each separate view. Objects from consecutive frames are labelled as candidates if found within close proximity. This process is repeated until one candidate is selected as a ball.

Similarly, J. Ren et al [2] modelled a background image via the Gaussian Mixture Model before identifying candidates based on their size and velocity over multiple frames. The 2D ball positions from different camera views are then integrated to obtain a 3D position using motion models [1] or geometry [2].

The downside of such approach for ball detection and tracking is that the number of frames required until a candidate is selected is dependent on the severity of noise. Additionally, trajectory mining or motion analysis is impossible in the presence of heavy occlusion especially during player ball interaction as the output of frame differencing in [1] and background subtraction in [2] would give objects that merged.

Kim et al [3] and Reid et al [4] presented different techniques to estimate the position of the ball in 3D with a single camera by utilizing reference players and shadows. These techniques are unlikely robust however as the shadow positions are influenced by light source rather than on camera projections.

Matsumoto et al [5] utilized 4 cameras employing background subtraction and template matching framework to identify ball candidates in each view although

their research focused on finding an optimized viewpoint given the tracks and did not include any analysis in 3D.

For the detection and tracking of soccer ball without any analysis in 3D, the techniques presented in [5] to [11] all follow a similar approach in which multiple candidates are generated over consecutive frames before declaring one the real ball. To conclude, while there have been many related works on ball detection and tracking in the context of soccer videos, they all require the object to be detected over multiple frames before it is declared as a ball. Additionally, none of them have used multiple cameras as a means to enhance the performance of the detector.

Chapter 3

Object Detection

Every tracking problem requires an object detection technique as a precursor. In fact, the ability of the system to detect the desired object and reject false alarms is pivotal to the success of the tracker. In this chapter, we will provide a detailed description of our object detection framework. The first section covers our background subtraction algorithm, aimed to eliminate all pixels that are non-green in colour. As we will see, the result of the subtraction algorithm produces a binary image whose pixels need to be grouped into separate objects (section 4.2). Then from section 4.3 onwards, we will classify all these objects as either a ball or a player through various techniques.

3.1 Background Subtraction

Given an image, we want to identify regions containing the object of interest. Background subtraction is an image processing technique wherein an images regions of interests (foreground) are extracted for further processing. The approach covered here take advantage of prior knowledge that an image of the soccer field will be predominantly green in colour. More specifically, the histogram of the green channel of such an image is expected to contain multiple strong spikes. This approach thus aims to label every pixel as either a foreground or background depending on the value of its green channel. Given an input image, we first generate a histogram of its green channel with 256 bins.

Through the histogram, the dominant pixel intensities which constitute the soccer field become noticeable. We then generate a binary mask of the input image via back-projection i.e. at each location (u,v) of the input image, we collect the value from the selected channel (green in this case), find the corresponding

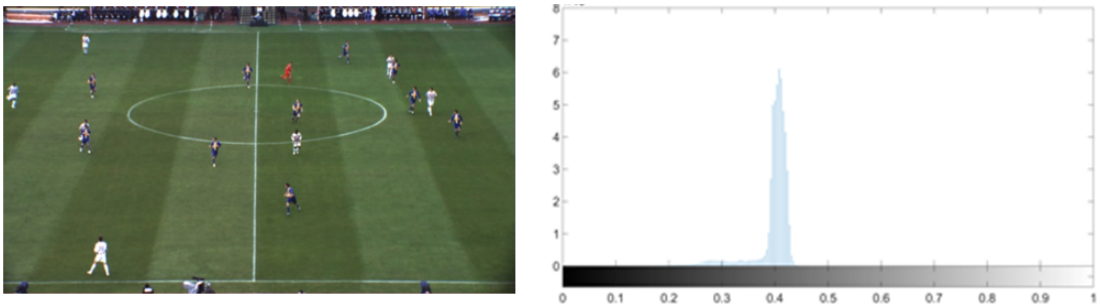

Figure 3.1: Input frame (left) and histogram of green channel (right)

histogram bin and either label it as a background or foreground by setting the mask at location (u,v) to 0 or 1 respectively. We label all pixels whose number of occurrence lies within 50% of the peak of the distribution as background pixels. All other pixels are labelled as foreground.

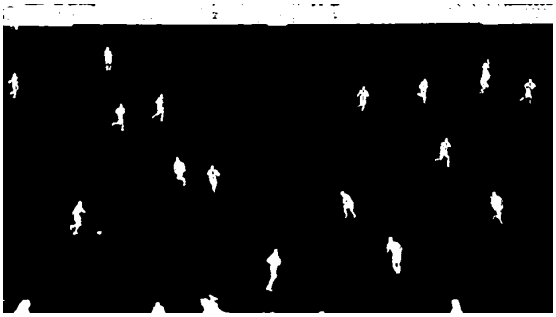

Figure 3.2: Result of background subtraction

3.2 Connected Components Labelling

As the current output is just a binary image filled with 1s and 0s, the next step involves extracting and labelling the various disjoint and connected components in an image, allowing for further analysis and is central to many automated image analysis applications.

We use an approach called connected components labelling. For this, we label the foreground pixels as a single entity if they are 8-connected. By definition, a set of pixels, P, is a connected component if, for every pair of pixels pi and pj P, there exists a sequence of pixels pi, . . . , pj such that all pixels in the sequence are 1 and every 2 pixels are adjacent. The figure below illustrates the output where the number on the object indicates its assigned identity.

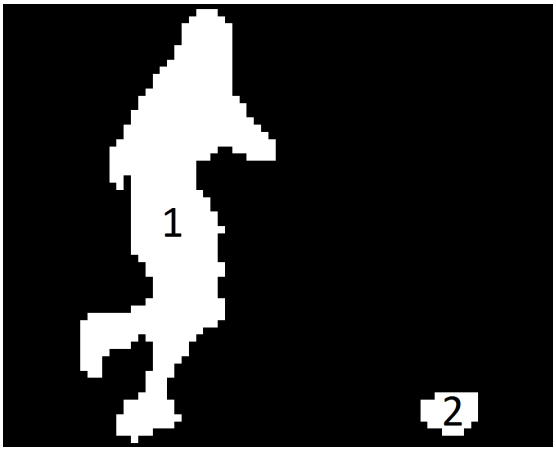

Figure 3.3: Connected components labelling

3.3 Contour Analysis

Given these components, including noise, the next task is to distinguish the different objects (ball or player). For this, we utilize object properties such as size and roundness. An objects roundness is determined from the equation below.

$$\text{Roundness} = \frac{Perimeter}{4\pi Area} \tag{3.1}$$

An object is labelled as a ball candidate if its roundness metric is within 0.5 1.5 and exhibits certain minimum area. For illustration, sample contours with their corresponding roundness and area are shown in the Table below.

3.4 Automatic Player-Team Identification

Given the player labelled objects (including noise), our next task is to identify the team (cluster) he belongs to (Blue/White team or Referee). To this end, we have employed the K-means clustering algorithm to learn a set of features that describe the mean of all 3 clusters. Assuming we have a set of player labelled objects accumulated from each camera with feature vector x, K-means aims to segregate the data into k clusters by minimizing the within cluster sum of squares and is defined as follows.

Our feature vector is generated as follows. First, we crop a rectangular region around the player before resizing it to a resolution of 20 x 45. The images are then separated into its individual RGB channel before being concatenated to form a 2700 dimensional vector (20 x 45 x 3). Thus, given a total of N points

(images) of the players and referees in a 2700 dimensional space, we partition the points into 3 clusters for the referee and 2 teams. This process is only run once upon initialization. Players in the subsequent frames are then assigned into their respective clusters via the nearest neighbour classifier.

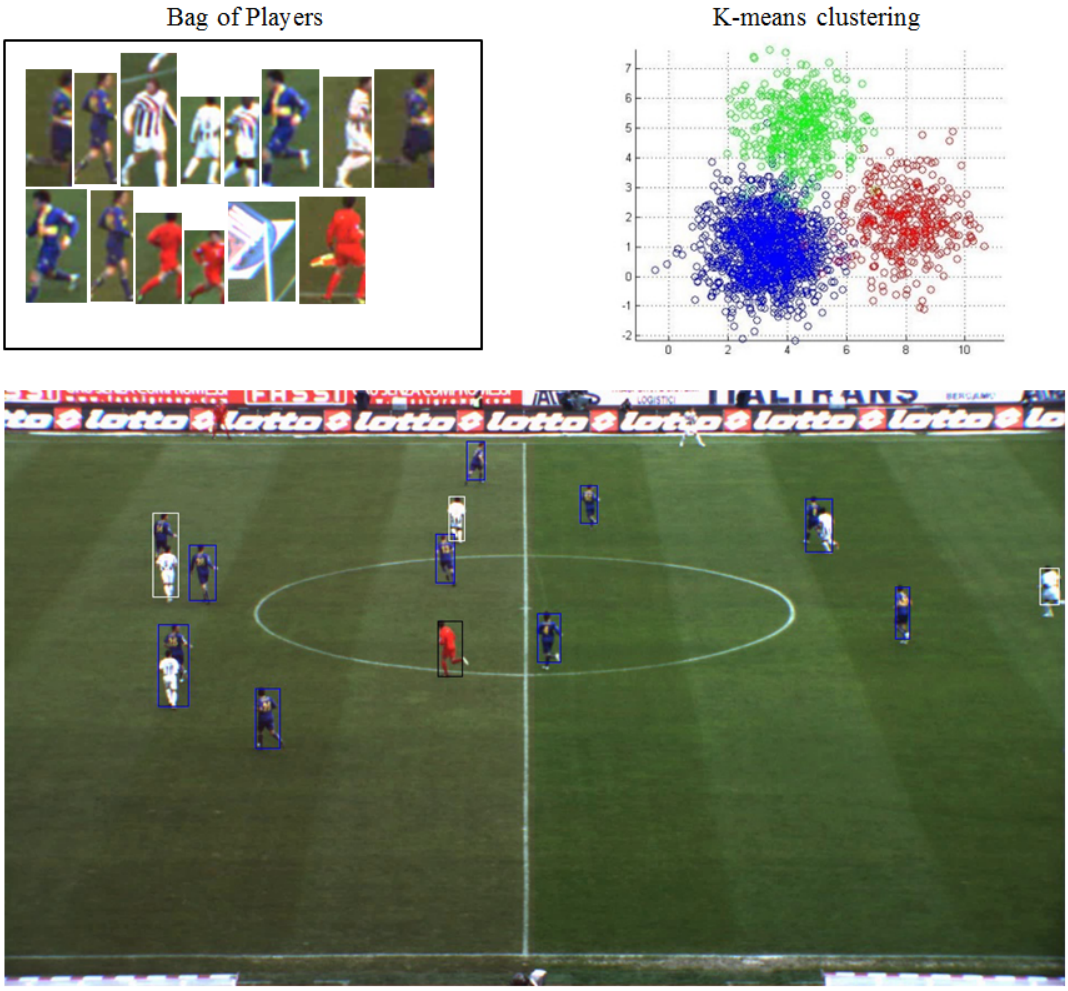

Figure 3.4: Player-Team Identification via K-means clustering

3.5 Template Matching for Ball Filtering

As there can be several ball-labelled components returned from 4.2 and 4.3 due to noise and artefacts, we employ the template matching framework via normalized cross correlation (NCC) with an offline generated template as the final means to identify the best ball candidate. In the template matching framework, the template is correlated with the image with the goal of identifying similar patches

in the image. At the image location (u, v), the normalized cross correlation is defined as

$$R(u,v) = \frac{\sum_{x',y'} T(x',y') \cdot I(u+x', v+y')}{\sqrt[2]{\sum_{x',y'} T(x',y')^2 \cdot I(u+x', v+y')^2}} \tag{3.2}$$

Where an output of 1 indicates a perfect match and -1 the worst match. We have manually generated 2 templates of the ball to be used interchangeably depending on the location of the object in the frame. A figure illustrating the output of the NCC is shown below.

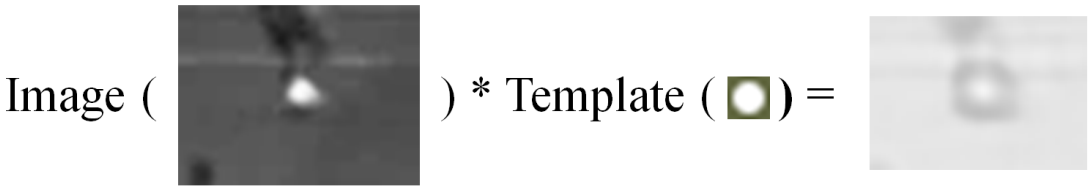

Figure 3.5: Template Matching via Normalized Cross Correlation

Before we end of the chapter on object detection, note that framework described for the ball detection up till now is not robust against noise as it is highly likely that the camera assigns any object as a ball if its area and roundness satisfies the threshold. In addition, the system treats each of the 6 input videos separately and will thus continue to search for the ball even if it has already been detected. These issues however, will be resolved in chapter 6 on multiple camera analysis when we integrate the information from multiple cameras. With that, we conclude the chapter on object detection.

Chapter 4

Object Tracking

Throughout the video, after a successful detection, we would like to establish the trajectory of the objects as it travels over consecutive frames. In this chapter, we discuss the tracking algorithms for both the ball and the players. Sections 5.1 will discuss the ball tracking algorithm while section 5.2 the player tracking algorithm.

4.1 Ball Tracking

Due to the unpredictable motion of the ball, we implement a simple tracking algorithm utilizing prior knowledge of soccer while working in conjunction with the player tracker for improved accuracy. The technique employs the template matching framework from section 4.5 with the only difference being where and how the search is performed depending on the status of the ball. In the following sections, we go through the different scenarios that occur during soccer and explain the minor changes in the template matching framework that we use to successfully track the ball throughout the video.

4.1.1 Ball in Play

Suppose a ball has been detected, we construct a window, manually designed to be large enough such that it contains the ball at the next frame even at its maximum speed.

We update two variables for every frame. A tracking lifetime metric, which is incremented for every successful track and a ball lifetime metric, which is incremented for every frame after a successful detection. The usage of these

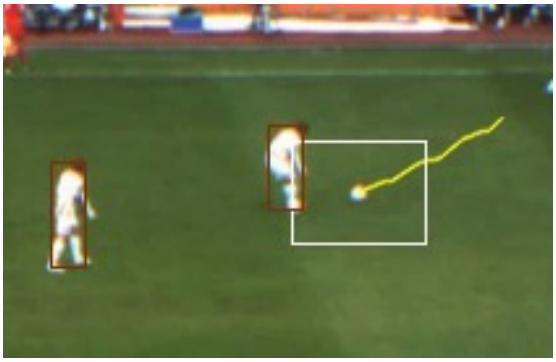

Figure 4.1: Default window size 45 x 35

variables will be explained in the next section.

4.1.2 Player-Ball Occlusion

A common scenario that occurs in soccer is when the ball goes under occlusion as the player dribbles it. To handle such types of occlusion, when the distance of the ball to the nearest player reaches below a threshold (15 pixels in our implementation), we construct a larger window (100x60) and attach the center at the feet of the player and assume that the player is dribbling the ball. Then, for all other bounding boxes enveloped within the window, a mask is generated so as to prevent misdetection.

For every frame it remains attached to a player, however, we do not increment the tracking lifetime by 1. This to prevent scenarios in which the system fails to locate the ball as it leaves the player and hence tracks the player for an indefinite duration. The track ratio, defined below, ensures that the window will remain attached to the players feet until a match has been found or its track ratio reaches below a threshold.

$$\text{Track Ratio} = \frac{\text{tracking lifetime}}{\text{ball lifetime}} \tag{4.1}$$

The figure below shows the system tracking the ball under player-ball occlusion till it escapes.

4.1.3 Ball out of play

When the ball goes out of bounds, a throw-in is awarded to the opposing team. During the throw-in, up till the ball enters the field of play, the player tasked to throw the ball in is the only player allowed to stand beyond the side-line. There-

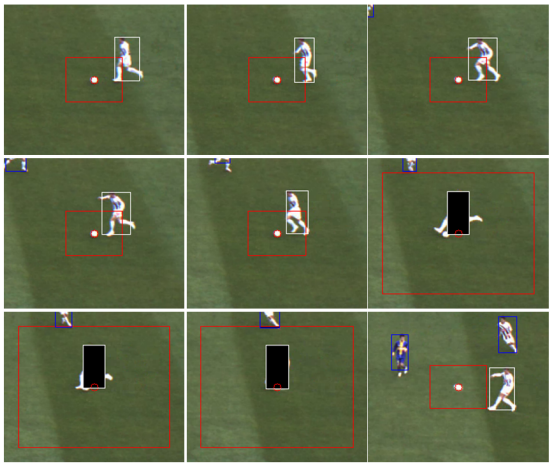

Figure 4.2: The frames 62, 64, 66, 68, 72, 74, 76, 68, 80 are shown

fore, when such an event occurs, we restrict the detector to perform its search only on the regions and the player whose position lies outside the boundaries of the soccer field.

In addition, a slight change to the algorithm described in 3.1 if the ball is located within the bounding box of the player is that no mask will be generated. Instead, we will assume the location of the ball at the top of the bounding box if no detections are found. As the ball re-enters the field of play, the tracker will revert back to either the algorithm described in 3.1 or 3.2, depending on the status of the ball (Figure below).

Figure 4.3: The frames 52, 58, 83, 108, 115, 125, 134, 190, 195 are shown

Note that the algorithm described above only applies to the side-line situated closer to the camera as it is much more challenging to detect a ball going out of play for the side-line at the opposite end. The placement of the cameras however,

ensures that the side-lines for both sides of the field are covered.

4.2 Player Tracking

The player tracker is based on a Kalman filtering framework which uses the equations of kinematics while accounting for statistical variations in the measurements and the model itself. We will simply state the Kalman filtering algorithm without delving into the derivation of the equations.

4.2.1 Kalman Filter

The algorithm consists of two stages, a predict stage which uses the learned motion model and the past state to estimate the next state and an update stage which uses the measurements corresponding to the next state and uses it to correct the motion model as well as to give a better estimate of the state vector given the prediction and measurements. In our implementation, the measurements for the update stage are given by the centroid of the nearest player (output of connected components analysis) if the distance is below 20 pixels. The update stage will not be invoked if the distance condition is not satisfied i.e. the player will take on its predicted coordinate.

4.2.2 Occlusion Handling

In the figure below, a single bounding box is generated for multiple players if they lie in close proximity. This is the result of the background subtraction and connected components algorithm returning a single connected object consisting of 2 players.

We resolve occlusion by taking advantage of the fact that in soccer videos, occlusions only occur when players move past each other. More specifically, each bounding box can only contain the coordinate of a single player; any more would indicate the presence of occlusion. Our occlusion handling algorithm is thus designed to segment players under occlusion via template matching. Given multiple players under occlusion, we employ a template matching framework using an image from the previous frame. For each bounding box in question, we convolve with it a template. The template with a higher score will then be selected to mask out the region. This process is repeated until all occluded players are segmented.

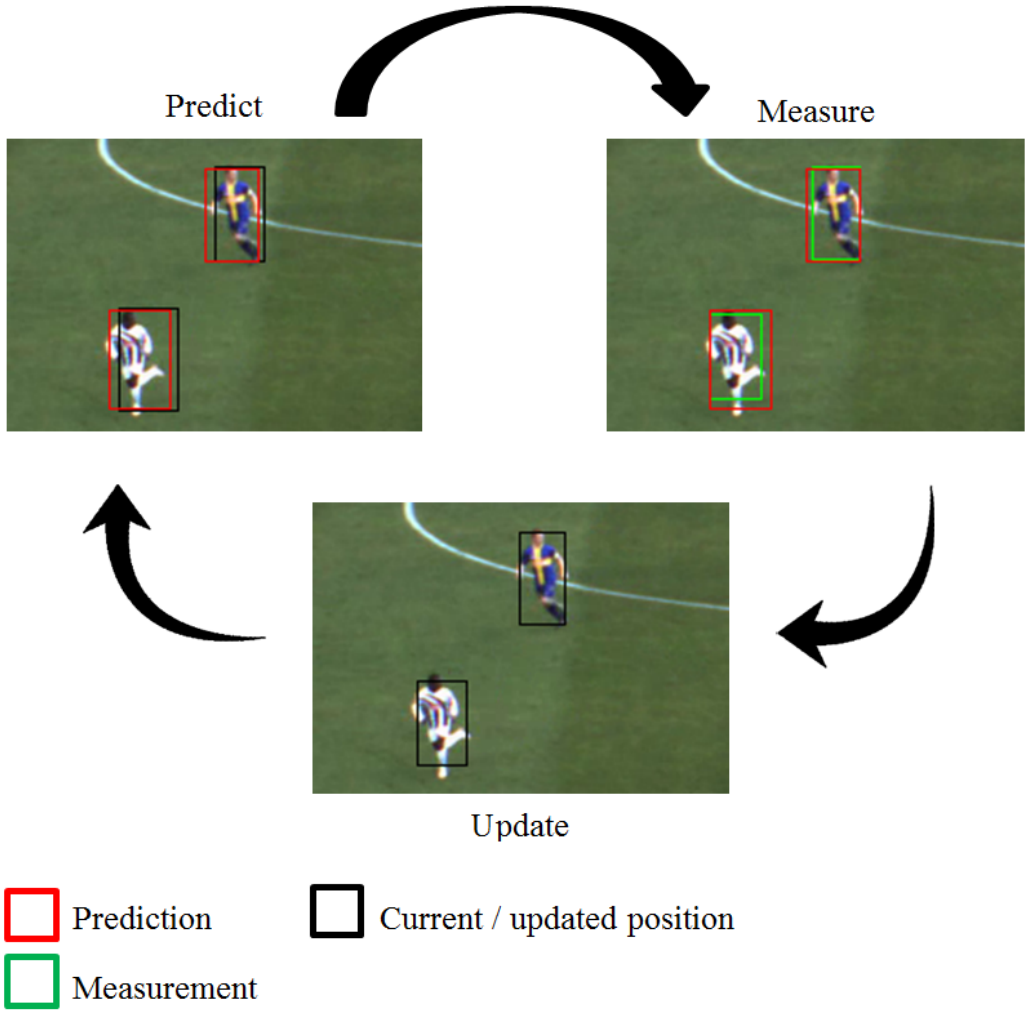

Figure 4.4: Kalman filtering framework for the player tracker

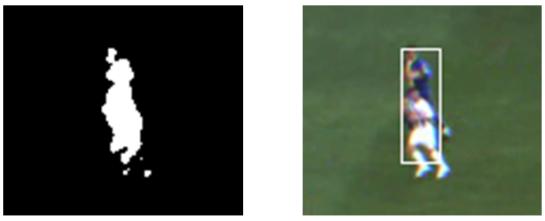

Figure 4.5: Occlusion in player tracking

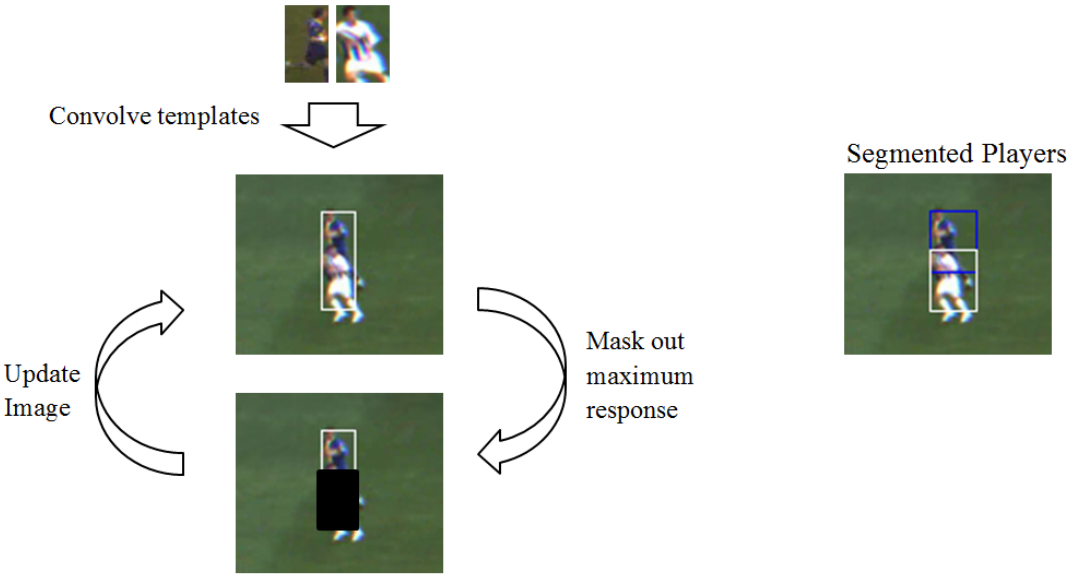

Figure 4.6: Occlusion handling framework

Chapter 5

Multi Camera Analysis

In the previous chapters, we are able to detect and track the ball in 2D. The location of the ball and players are however, in terms of image pixels. Also, recall that the system detects and tracks the best ball candidate (if there is one) in each view. Hence at any point of time, we can have up to 6 tracked balls in the field. What we would like to achieve is the ability to identify candidates that correspond to the real ball. Furthermore, since the ball often flies across the air, we would like to have a good estimate of its actual position in 3D.

In this chapter, we will finally integrate the information from all the cameras to present the results in terms of world coordinates. We will start with a theoretical derivation of the homography matrix in section 6.1. This matrix relates points between 2 image planes and is responsible for mapping the players and the ball from each camera view onto a common model of the soccer field. The problem of establishing correspondence between detected objects across cameras then follows in section 6.2, i.e., given the projections of all players on the field model, we need to identify the pairs that belong to the same object in the real world. In the case of the ball, we need to identify the pair that corresponds to the real ball. In section 6.4, we present the algorithms used to localize the ball in 3D.

5.1 Object Registration

With the image coordinates of a ball and the players and the homography matrix of the respective camera, we project all objects onto the field model, shown in the figure below. The figure below shows an example of registering all tracked objects onto a model of the soccer field. The colour of the circle indicates the players team while the number inside the circle the camera that was tracking it.

The ball candidates are indicated by a black circle with a red outline and are being tracked by cameras 1 and 3 as indicated. Notice the presence of noise on being projected near the left goalpost. As mentioned at the end of chapter 4, such results occur due to the inaccuracy of the detection framework.

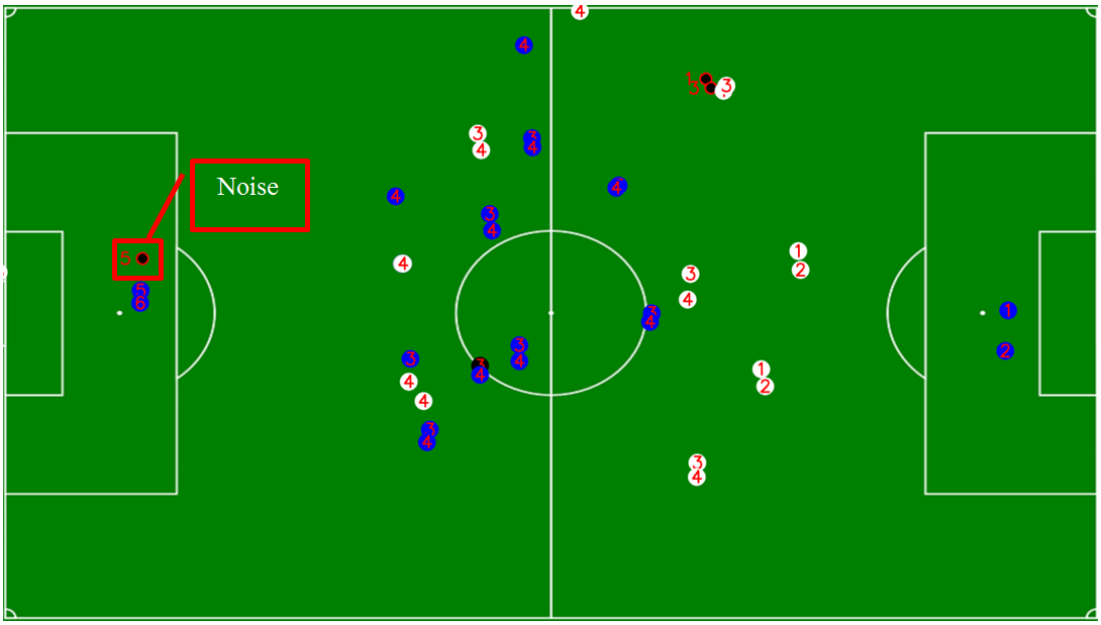

Figure 5.1: Registering all objects onto the field model (Frame 339)

Since the dimensions of the model are proportional to the actual field, we can apply a scaling operation to convert the coordinates of all projections from pixels to meters.

5.2 Fusion from Multiple Cameras

As seen in the previous section, the effect of registering the objects from multiple cameras produces multiple objects on the field model. The next step would then be to establish correspondence between all pairs of players and the ball. Different methods are being used to generate correspondence for the players and the ball as can

5.2.1 Player Fusion

The nearest neighbour method is used for determining correspondence between pairs of i.e., for each player i, we compute the Euclidean distance between every other player j from the same team and declare the pair with the smallest Euclidean

distance as the same object if their distance does not exceed a certain threshold i.e.

$$argmin\mathbf{p} < threshold \tag{5.1}$$

Players i who do not meet the criteria above will still remain in the field albeit without a pair. In this case, it is assumed that the player is being tracked by a single camera and thus does not have a closest pair. Such a scenario also occurs when the occlusion handling mechanism fails.

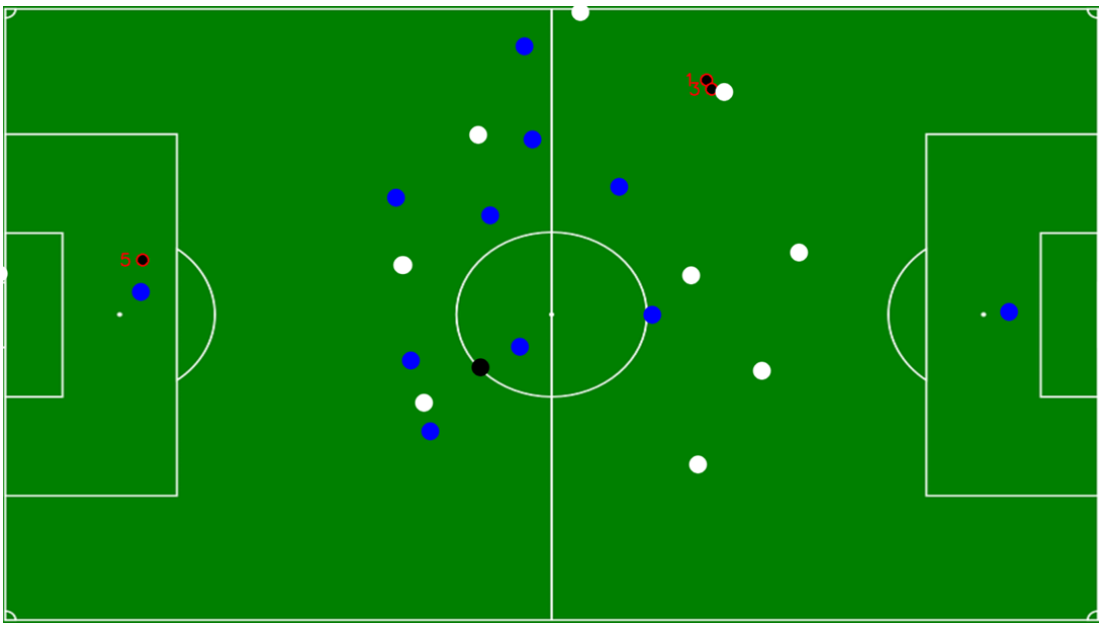

Figure 5.2: Player Correspondence (Frame 339)

5.2.2 Ball Fusion via Epipolar Constraints

A different method must be used for the ball as it can be easily observed in the figure below (where ball 5 and 6 belongs to the same object) that the nearest neighbour method would not work. Such a result can be reasoned by the fact that the homography transformation assumes all points in the scene lie on a plane. The image coordinate of a ball flying across the air would be higher. It is extremely important to establish the correct correspondence for the ball as identifying a false candidate would ruin localization and the remaining sections on localization in 3D. In order to achieve proper correspondence, we employ the Epipolar geometry of stereo vision in which a point viewed by 2 cameras generates a set of geometric constraints that relates their 2D position in both cameras

[13].

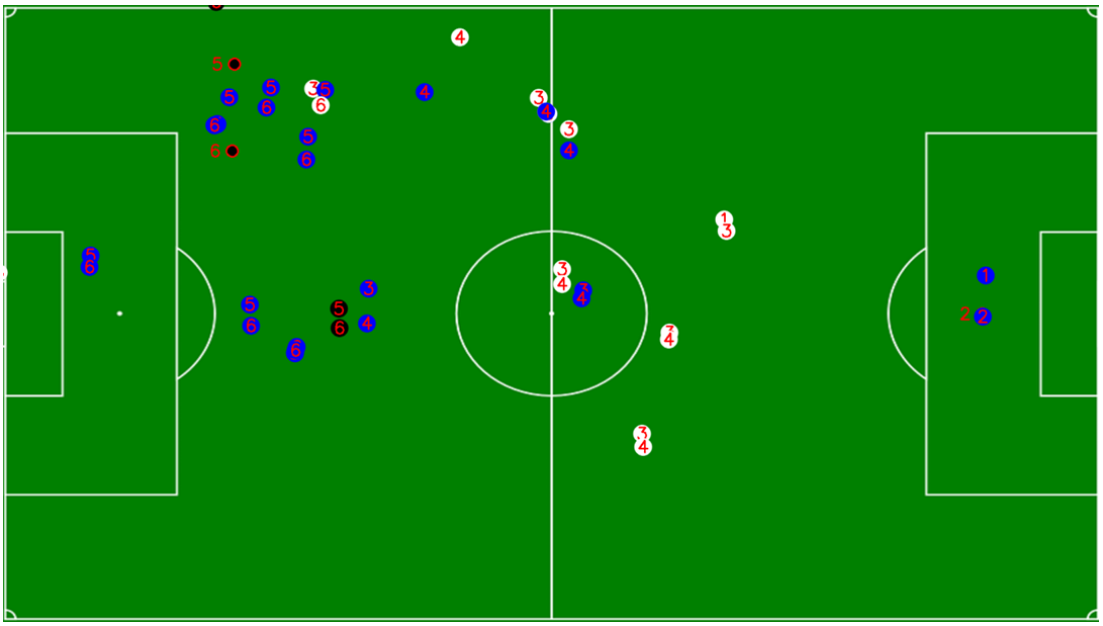

Figure 5.3: Registering all objects onto the field model (Frame 433)

With reference to figure 6.4 below, the Epipolar geometry can be explained as follows. Suppose that a point in the real world (denoted by the red circle), observed by 2 cameras and is projected onto the field model (denoted as and respectively) via the homography transformation. Then, the back-projected lines and will always intersect at the location of the object in 3D. It is this property that is of most significance in searching for a correspondence

These lines however, do not intersect in practice due to errors caused by camera calibration so instead, we will iterate through all possible pairs and compute the shortest distance between the skewed lines to determine if they belong to the same object, namely the ball

Note that the method used to extract the camera coordinates follows [14] with the exception of the usage of maximum likelihood estimation to minimize the re-projection error. Appendix C lists the data and results of camera calibration.

5.3 Height Estimation

Upon designating the pair of projections as the true ball, we can localize its position in 3D by assigning the midpoint of the perpendicular line as the estimated 3D position of the ball, assuming that the errors from both cameras are equal in magnitude. In the figure below where and denote the camera location, and the

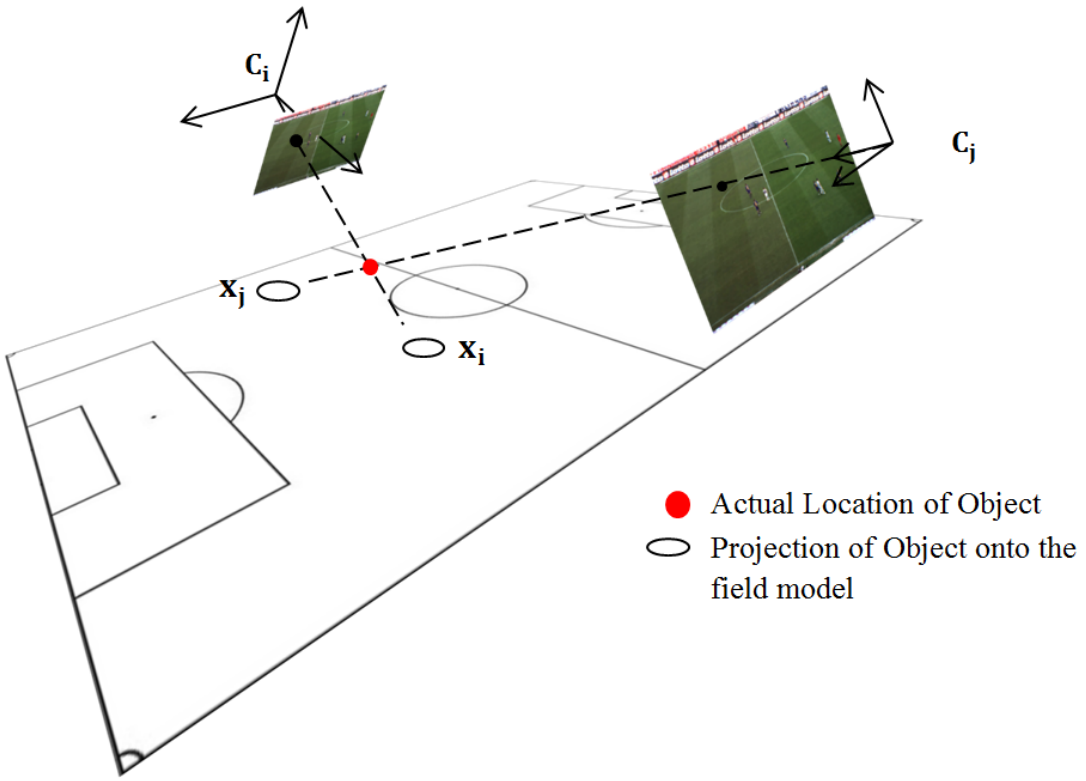

Figure 5.4: An illustration of the Epipolar geometry in multi view soccer videos.

projection of the ball onto the field model, the height of the ball is assumed to be b.

When the number of detections is reduced from 2 to 1, cues about the 3D position of the ball can be obtained by establishing a plane from last known 3D positions. In this scenario, it is assumed that the trajectory of the ball travels along a vertical plane, perpendicular to the ground plane (Figure 6.7 below).

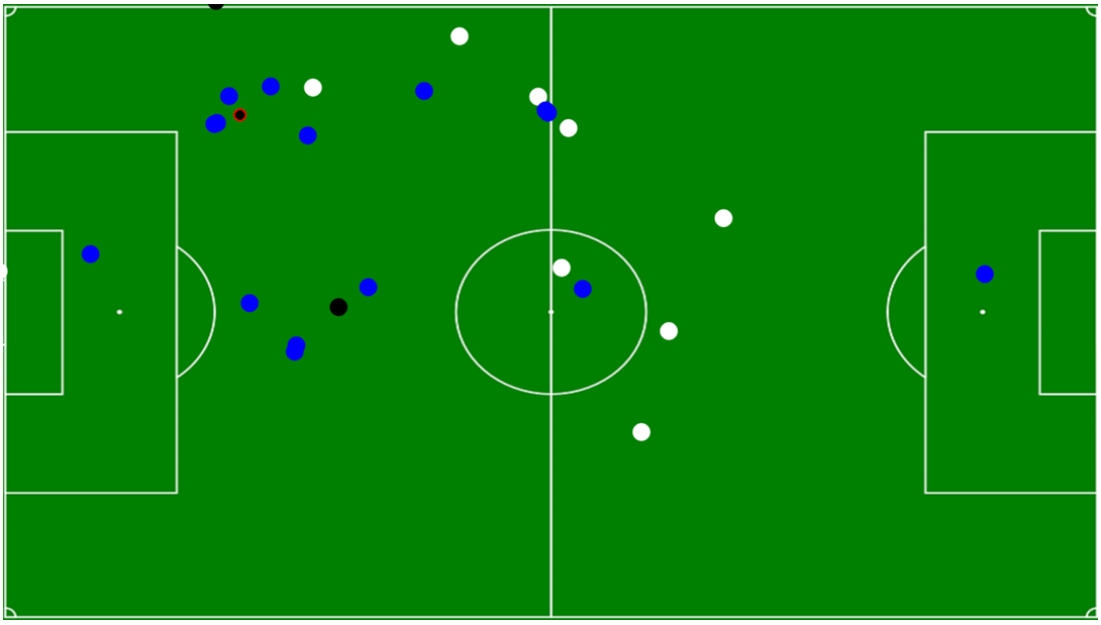

Figure 5.5: An illustration of the Epipolar geometry in multi view soccer videos.

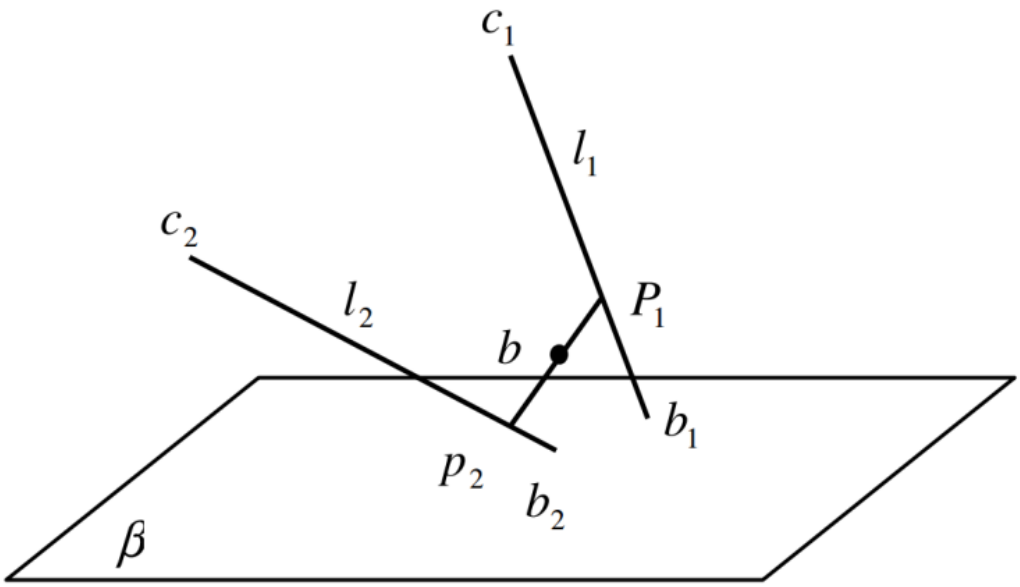

Figure 5.6: Estimating the 3D position of the ball using multiple cameras

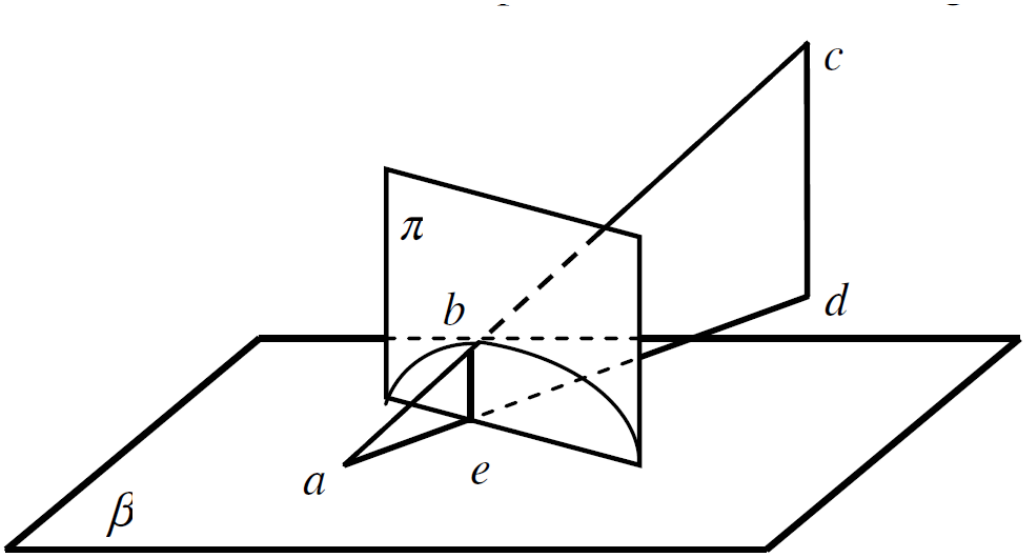

Figure 5.7: Estimating the 3D position of the ball using a single cameras

Figure 5.8: Height Estimation. The frames 610, 615, 620, 625 of camera 5 from top left to bottom right are shown. The ball is out camera 5s field of view for frame 620 and 625.

Figure 5.9: Height Estimation. The frames 610, 615, 620, 625 of camera 6 from top left to bottom right are shown. The height of the ball in frame 620 and 625 is estimated using a single camera.

Chapter 6

Results

An important aspect of tracking algorithms is its performance. Being able to assess the reliability of a tracking system via defined metrics provides a useful way to gauge its performance against the state of the art algorithms. More importantly, the quantitative analysis of the designed tracker helps measure progress as it provides feedback for future improvements. In this chapter, we present the results of the ball tracking algorithm and discuss some of the limitations of the system.

6.1 Results for Ball Tracking

For tracking in 2D, the tracker was evaluated based on the commonly used precision and recall scores. Their definitions, along with all relevant quantities are all defined below.

True Positive: Total number of frames where both ground truth and system results concur on the presence of the ball with the centroid of their bounding boxes lying within a specified distance. **False Positive**: Number of frames where the ground truth does not contain the ball but the system detects an object. **True Negative**: Total number of frames where both ground truth and system results concur on the absence of the ball. **False Negative**: Number of frames where the ground truth contains the ball, while the system either does detect the ball or the detected ball does not lie within a specified distance.

For multi camera tracking, the performance was assessed by computing the coverage of the ball, i.e. the percentage of frames for which the ball was tracked by at least 1 or 2 cameras and the triangulation error, i.e. the distance between the skewed lines. Figure 7.1 displays the classification of tracks for each camera over 3000 frames with their individual precision and recall scores tabulated in the

table below. The true positive and false negatives are computed with an error threshold of 10 pixels. From the figure, it can be seen that cameras 2 and 3 each suffers from high false negative rate compared to other cameras.

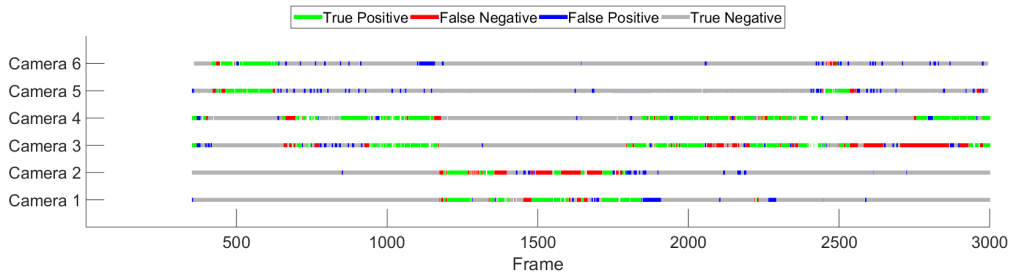

Figure 6.1: Classification of object trajectory over 3000 frames

This can be attributed to the fact that the ball for the ISSIA soccer video is under frequent occlusion with respect to said cameras. For example, the figures below explain the high false negative rate for camera 3 from frame 2500 to 2800. From frame 2530 onwards, camera 3 fails to track the ball as it goes out of play. It also remains unseen by the opposite camera 4 until frame 2749 when a player appears in camera 4s field of view to throw the ball in. The distance of the ball from camera 3 and the surrounding region makes re-detection a challenging task.

Figure 6.2: The ball goes out of play in frame 2530 (left) of camera 3 and remains undetected. Frame 2749 is shown on the right

Despite the low precision and recall measures for cameras 2 and 3, the capability of the system in tracking the ball over 3000 frames is rather sufficient as the placement of cameras on opposite sides of the soccer field ensures that the ball remains visible most of the time. The results below display the coverage of the soccer ball when taking all 6 cameras into consideration. The results are promising with the ball being tracked over 80% of the frames.

Lastly, the triangulation error over 3000 frames have been summarized by its minimum, maximum and average error. With an average error of 0.4 meters, the results are much better than the sensor based systems that rely on radio signals.

Figure 6.3: The ball goes out of play in frame 2530 and is only re-detected when it appears in camera 4s field of view in frame 2749

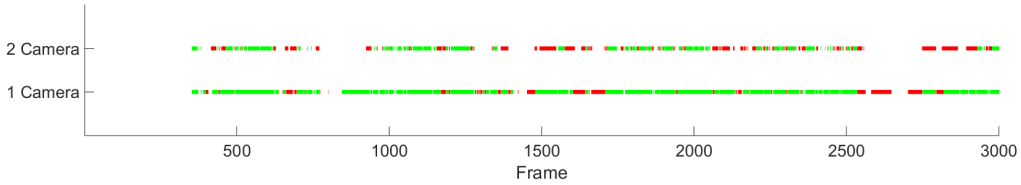

Figure 6.4: Ball coverage over 3000 frames. The top row indicates frames for which the ball is tracked by at least 1 camera while the bottom frames at which the ball is tracked by at least 2 cameras

To summarize, the figures below indicate the ball tracking precision and recall scores with an error threshold from 0 to 30 meters. The plot in red indicate the performance of the tracker at the beginning of this project and in blue the improved tracker. With an error threshold of 10 pixels (table 7.1) we have increased both precision and recall scores by an average of 18.2 and 9.8 percentage points respectively. The significant increase in precision can be attributed to the 3D analytics framework. By localizing the ball in 3D, we are able to activate / deactivate the ball tracker for each camera depending on the location of the ball thereby reducing the false positive rate.

6.2 Limitation of the Occlusion Handling Algorithm

The occlusion handling algorithm described in 5.2.2 is only invoked when a bounding box contains the coordinates of at least 2 players. As such, it is unable to segment players who are occluded at the start of the video as their pixels will be connected and thus will be declared as a single object. This is shown in the figure below when the tracker was initialized at frame 340. As can be seen, the occlusion handling algorithm fails to segment the pair of players who were already

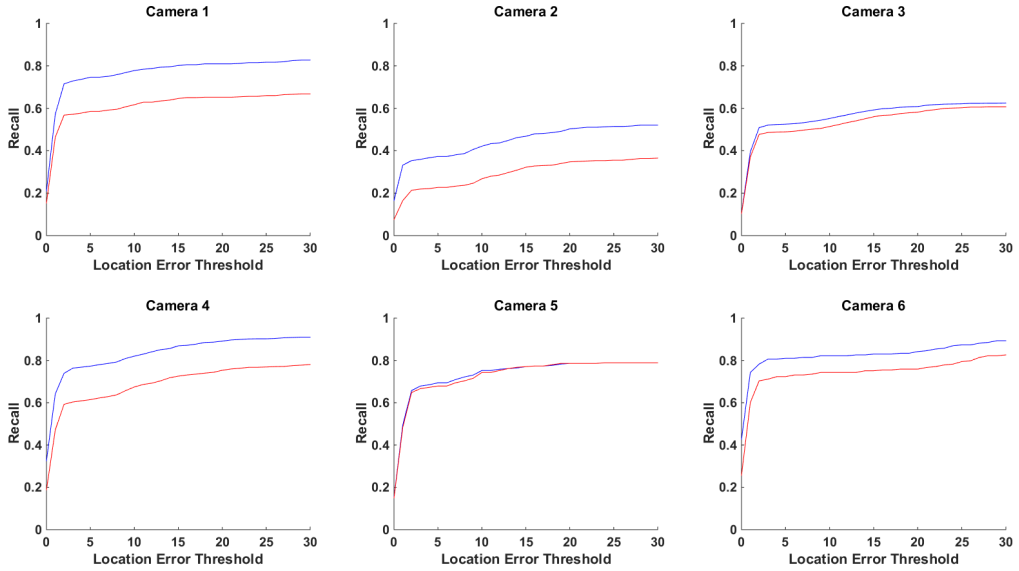

Figure 6.5: Recall plots for each camera

connected at the beginning.

6.3 Limitation of K-Means for Team Identification

As was previously indicated in section 4.4, the features that describe the 2 teams and the referee are generated via K-Means clustering upon initialization. The players are then assigned into their respective teams in subsequent frames based on the Euclidean distance to the generated cluster centers. An inherent limitation of the traditional K-Means for clustering is that the generated centers are affected by the composition and structure of the data. As there is an in imbalance of class samples as the players on each team far outnumber the referees, the cluster center for the referee may not be in the ideal position. Additionally, the method used to generate the feature vector plays an important role in the success of the team identification algorithm. In the figure below, the referee and some of the players have been assigned a wrong label.

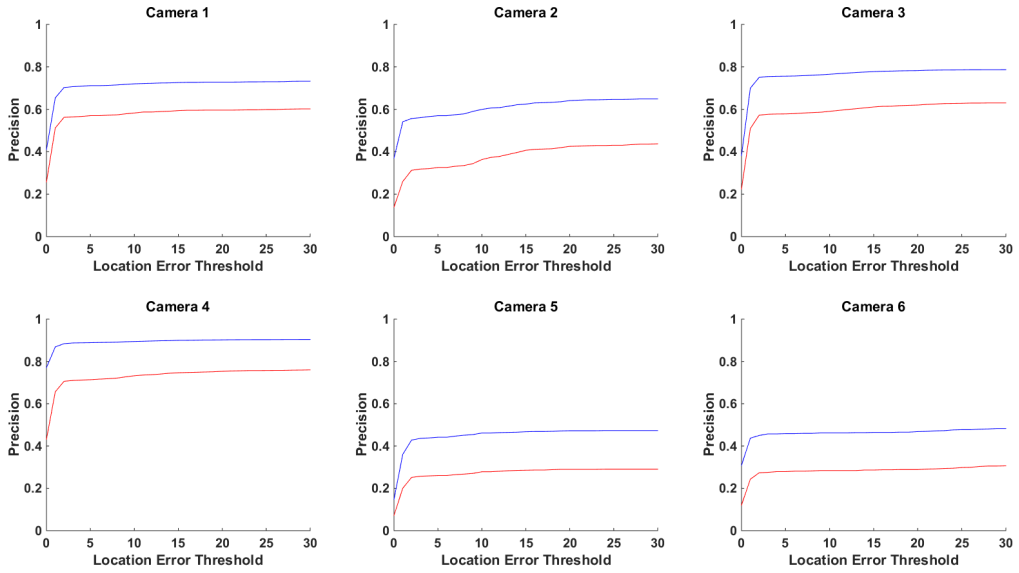

Figure 6.6: Precision plots for each camera

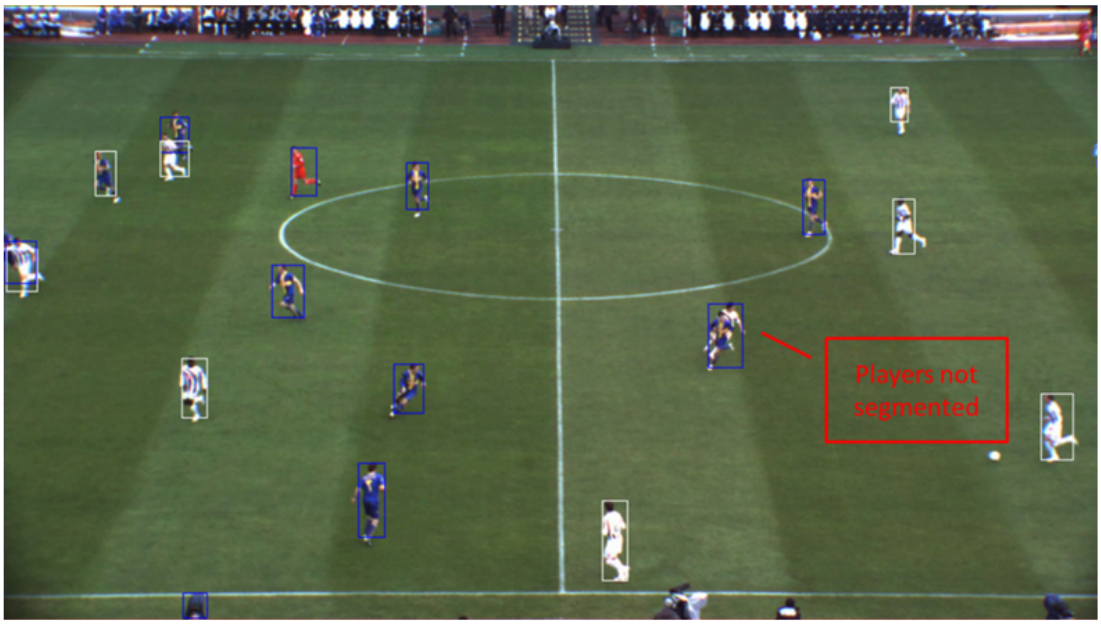

Figure 6.7: Frame 355 of Camera 4. The players highlighted by the red box are not segmented

Figure 6.8: Frame 2325 of Camera 4. The referee and some of the players have been labelled wrongly

Chapter 7

Conclusion

An automated real time tracking system for soccer videos has been presented in this thesis. The contributions are summarized as follows; an automatic player team identification algorithm utilizing K means (Section 4.4), an improved ball tracking algorithm (Sections 5.1.2 and 5.1.3) with precision and recall improvements of 18.2 and 9.8 percentage points respectively, an occlusion handling algorithm for player tracking via an adaptive template (Section 5.2.2) and multiple camera analysis in 3D (Chapter 6).

Its distinction between other existing vision based methods for ball tracking is (1) the utilization of Epipolar constraints for fast object detection, (2) the handling of out of game situations and (3) the coordination with the player tracker for occlusion handling. Experiments on the ISSIA soccer dataset verify the efficacy of the proposed methods with the potential to do even better. Although the precision and recall measures of the prototype are not as competitive as the state of the art trackers, the simplicity of the designed algorithms allows ease of integration with other state of the art trackers. We therefore conclude that the system has the potential to outperform other trackers.

Possible improvements would be to change the subtraction algorithm as it was noted that the algorithm does not work well on a different dataset with varied illumination. The weakness of the current approach for ball tracking is the usage of templates for ball detection. Extensions could therefore be the integration of a trajectory analysis based algorithm to learn and generate a template online which could then be reused for the future soccer matches provided the camera setup is similar. An algorithm that extrapolates missing or eliminates false trajectories could also be explored. Additionally, tracking of the ball via physics would be

helpful if the camera calibration errors are miniscule. It was not successfully done in this thesis due to said errors.

For player tracking, a robust correspondence and occlusion handling algorithm that takes into account the total number of players in the field would work well provided it is able to handle a reduction in the number of players (red card). Additionally, the limitation of the automatic player team identification described in section 4.4 is its inability to identify the separate goalkeepers. Furthermore, the performance of the clustering algorithm is suboptimal due to the fact that the input to the clustering algorithm comes from every single camera. As such, there will be more player samples than referees or goalkeepers.

Bibliography