A robust method for tracking sports players for web or mobile display

Andreas Aristidou

A Dissertation submitted to the King's College London in Partial Fulfilment of the Requirement for the Master of Science

Department of Electronic Engineering
King's College London

2005-2006

Declaration

I, Andreas Aristidou of King's College hereby declare that, except where specifically indicated, the work submitted herein is my own original work.

Date: September 08, 2006

Signature: ……
Andreas Aristidou

Abstract

An architecture system and a method for tracking people are presented for sports applications. The system's input is video data from static camera of a stadium and the output is the real world, real-time positions of the players during a sport event. This output can be used for low-bandwidth match play animations for web or wireless display, and for analysis of fitness and tactics of the teams and players. Firstly, in this study, an efficient real-time background modelling and maintenance is described based on the segmentation of input images into stationary and non-stationary blocks. In order to detect foreground objects a method based on background subtraction is implemented, using chromaticity and lightness, with the intention of avoiding shadows and jitters. Object tracking is achieved using a cost function of blob information such as the centroid coordinates, the covered area, the velocity and the colour. A condensation filter is also implemented in order to cope with complex cases such as when regions enter/exit the scene and when they can be occluded by other regions. Thereafter, a field estimation algorithm, which utilizes the colour information of the ground is proposed and a new robust projective transformation from real image coordinates to the upper-view, virtual ground image is applied. The results of the proposed methods, the players' location and trajectories that provide enriched mobile services are presented and discussed.

Acknowledgements

I would like to thank Dr. Paul Pangalos, for making the study possible, by providing me his regular supervision and support to conduct this project. I am also very grateful to my parents, who supported me during my postgraduate studies and for giving me the opportunity to follow my ambition. I also owe an enormous dept to my friends Themis, Margarita, Pavlos, Evagoras and Christina, who helped me to accomplish my intentions. Furthermore, I would like to thank my friends Melios, Andreas, Vassilis and Modestos for their help in the video recording of my samples. Additionally, I would like to thank the owner of Mini Football "Fitness-land" Mr. Periklis Perikleous as well as "Famagusta Tennis courts" owners in Limassol, Cyprus, for providing me their football and tennis courts disinterested.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 Aims and Motivations

The increased public interest on sports led to the investment of huge amounts for their maintenance, empowerment and transmission via television. Many organisations, such as the FIFA World Cup, the Champions League, Formula 1, and the tennis grand slams like Wimbledon and Roland Garros, are in the centre of interest for many sports funs and businessmen. All these, together with the oncoming Olympic Games in London, motivated us to begin searching efficient ways to present sports events as play animations for web or wireless display.

The aim of this dissertation is to produce a novel object detection and tracking method, using a single fixed camera, which automatically tracks sports players, and identifies their real positions in order to describe the contents of an athletic event. The first goal to be achieved is to detect the playfield of a stadium where the game takes place. In this way, the moving objects of the playing grass court can be detected, and the players' positions will appear in a virtual ground image as a function of time. This can be used as an alternative way of playing low-bandwidth match animations providing enriched mobile services. This can also be used for analysis of fitness and tactics of teams and players during the match.

Background estimation is a basic requirement for the achievement of the primary goal of this study. Indeed, various individual problems should be overcome. For instance, moving foreground objects in the field of view, such as walking people, moving cars, and animals make background estimation very difficult. Sudden global illumination changes or a background that is changing its appearance (like a clock on the wall, clouds etc) are further problems. With regard to computer vision, the variation of the sport players' shapes and motion speed, their rapidly changing direction and the fact that many people interact, occlude, and make abrupt body movements, are dominating problems that render tracking even more difficult.

Another basic problem is the identification of the exact size of the pitch necessary for the resolution of an image of the game. Aiming to implement a projective transformation, this domain exhibits several challenging aspects having crucial role in the procedure. This happens because the nearest and the furthest parts of the field vary greatly.

An attractive research area is the analysis of positional data, which include identification of team game interactions. Multiple players' sports games provide a rich environment for modelling the actions of individual players as well as the behaviour of the teams as a whole. A large number of people and businessmen as well as sport clubs owners and coaches seek ways where they might watch at any time the match and be possible to have individual performance analysis.

1.2 Literature Review

Many research projects, published on human detection and tracking, focused on football player tracking [7], [10], [14], [31], [32], [33], [34], [35], [37] and [38]. Some used information such as texture, colour and shape for tracking [28]. Colour-based template matching is used in several approaches since it deals with occlusion problems [37], [38]. In contrast, other studies estimate motion and player position by Kalman [10] or condensation filters [14]. Kalman filter [11] has been used in many tracking applications due to its computational efficiency and its ability to predict future states.

Currently, there are several popular methods for tracking moving targets. Some of these achieve multiple tracking using a single camera. This could lead to the problem of occlusion. Some more effective methods used multiple static cameras. In this way, the problems of players' occlusion are solved, and their positions are showed in the 3D space [10], [34], [36]. This method increases the overall field of view and provides 3D estimates of ball location. Moreover, it improves the accuracy and robustness of estimation due to information fusion [10], [13]. However, it requires dedicated static cameras. Sports related works have been stimulated by several different aspirations; including action recognition, match reconstruction, and evaluation of a game. In [33], authors used multiple cameras to track players in the broadcast TV footage of American football games and aimed to identify actions based on visual evidence. The players' trajectory in panoramic view is also the output of a TV sequence in [34]. In [32], a 3D virtual animated view has been generated using two synchronised video sequences. The players' image resolution has been improved using zoom cameras. The correspondence between frames has been achieved based on matching field lines or arcs. Such studies provide an effective method of tracking and player positioning, but the implementation complexity is high. In addition to that, it is difficult to convince each association to approve the placement of all these static cameras in the field. Moreover, multiple cameras are not able to process old video files, but only the new files captured by these specific cameras.

The combination of a method that uses blob's information as well as the prediction of the next possible position of a player, using Kalman or condensation algorithm, could be an effective approach for players tracking. A powerful condensation technique for tracking an object through cluttered scene, which allows the propagation of conditional densities over time, is proposed in [12]. However, tracking multiple targets is an initial drawback of this scheme. In [8] a probabilistic exclusion principle that allows multiple trackers is introduced. It is based on the assumption that if several one-body trackers are employed, each with the same tracking algorithm, then two or more can coalesce onto the same target for which their model best fits [14]. Thus, this is an excellent approach and using this method, we can avoid the use of multiple static cameras, without the problem of occlusion.

According to the related bibliography, most reports focused on positional analysis, behaviour modelling, and silhouette processing. Minimal reports exist in ways of exporting the athlete's position and their transformation in a new virtual ground image, for display on web or wireless devices. This could be a very good application of this process, which has not been developed yet. Furthermore, we realise that, most of the literature focused on a small spectrum of sports, such as football, and have not paid attention to other indoor or outdoor sports such as tennis.

1.3 Thesis Overview

This dissertation is organized into seven chapters. The next chapter provides a real time robust background estimation model based on the segmentation of input images into stationary and non-stationary blocks. The processing time as well as the effectiveness of the proposed method is also analysed.

Chapter three presents a foreground detection method, which is based on background subtraction using chromaticity and lightness. This method attempts to detect moving regions in an image by differencing the current image and a reference background image in a pixel-by-pixel fashion. The next step is the blob processing, followed by the export of blob information, which is an essential condition for the tracker operation.

Chapter four introduces a multiple object tracker. An efficient tracking algorithm is developed, which establishes correspondence between objects across frames. In addition, the way that this method can cope with multiple cases such as when regions enter/exit the scene and when they can be occluded by other regions, is explained.

A new method of field region estimation and projective transformation of real image coordinates to a virtual platform is obtained in chapter five. Exploiting the characteristics of the playing field, such as the colour and dimensions, the sports ground boundary is detected, and feet positions of all players are projected on a virtual ground image from the upper view.

The evaluation of the results for football and tennis, the technical characteristics and some further applications in wireless technology are discussed and analysed in chapter six. Finally, contributions, future research and export conclusions are presented in chapter seven.

Chapter 2
Background Estimation

2.1 Introduction

In almost all systems of vision-based human analysis, detection of moving people is an important first step. The form of image blobs is used for later processes, such as tracking and activity analysis, because only the pixels of the moving region need to be analysed. There are three conventional approaches to moving object detection:

a. Temporal differencing (two or three frames) [2], [19], [20] is very adaptive in dynamic environments but does not effectively extract all relevant pixels.
b. Optical flow is a complex method that can be used to detect independent moving targets. However, it is inapplicable to real time algorithms without specialised hardware [16].
c. Background subtraction is the most popular method for motion segmentation, especially with a relatively stable background [15], [21], [22], [23]. It attempts to detect moving regions in an image by differencing the background from foreground regions. However, it is extremely sensitive to changes of dynamic scenes due to lighting and extraneous events. This is the implemented method used in the study.

One basic assumption of several existing tracking algorithms is that the background of the scene is known. This is attributed to the fact that in many cases and especially in real time processes it is practically difficult to record a separate background image without any objects in a video progression. An added difficulty is that the background might change slowly and therefore this process has to become accustomed. Indeed, it is imperative to develop an algorithm, which reconstructs the background image even when moving objects are apparent in each frame.

There many problems concerning background estimation, such as abrupt global illumination changes or a background that it is changing its appearance (waves, clouds etc). Therefore, the background initialisation involves extracting the background image of a scene with many arbitrary moving objects and when the background is visible for short periods of time.

In the context of making an algorithm for identifying the background in real time processes, it is also important to consider the algorithms processing time and the initialising time. In this chapter, we review the advantages and disadvantages of previous research focused on background estimation. Afterwards, we analyse our proposal method and present the corresponding algorithm and results. The Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR) between the real and the estimated background are also reported.

2.2 Related Work

Background estimation has been widely used in previous research on moving object detection in video. The background has been adaptively estimated from previous image frames. In order to estimate the pixels of the moving objects the current image frame of the video sequence was subtracted from the estimated background. After each frame, most of the existing algorithms update their current background estimation iteratively. The update factor can be regulated using a Kalman filter [24], or it can be a constant aging factor. The main disadvantage of this approach is that if there are any errors in the background estimate are removed slowly. Thus, this class of algorithms only works reliably when foreground objects move fast and background is visible during most of the time. Otherwise, the reconstruction fails to converge to the background only and it is constrained to an average of foreground and background colour.

Another methodology is to adopt a pixel-based temporal median filter over a large number of frames [27]. In this approach, each pixel has to contain background content for at least half of the input frames. An advantage of the median-filter algorithm over the weighted update algorithms is that the effect of blurring is not as severe. Nevertheless, the background image content can diverge from the correct background colour for this algorithm as well. This bias from the correct background colour is not always noticeable, but it can cause difficulties in automatic segmentation algorithms. LmedS (Least Median of Squares) is a similar method that is analysed in [18].

Another approach that mixtures the Gaussian, Nonparametric Kernel, and codebook is presented in [3]. This approach can perform better, but it needs extra computation and more memories. Bearing in mind that we have to work in real time, this approach appears to be very complex.

2.3 Background Subtraction

In this dissertation, a robust background estimation algorithm has been applied, based on the segmentation of input images into stationary and non-stationary blocks, as reported in [1].

2.3.1 Introduction

In this algorithm, the main idea is to apply a rough segmentation of the input images into stationary and non-stationary regions. The median-filter algorithm [27] is utilised to synthesise the background image, however, the non-stationary regions are excluded from the synthesis process. Therefore, no bias towards the foreground colour will occur in the reconstructed background. The classification takes place on NxN blocks (usually $N=8$) to make the background/foreground decision more robust. The subset of frames that show stable content in the block identifies parts of background content. The similarity of block contents over time is collected into a matrix, which contains the difference between the image content at the block position for each pair of frames. This matrix is decomposed to obtain the foreground/background classification. Low values correspond to the background regions, whereas high values correspond to the foreground regions. The subset of frames for which the sum of stationary matrix elements is minimised determines the background periods. In contrast, the subset of frames for which the sum of non-stationary elements is maximised is considered as foreground. A detailed description is provided in the following sections.

2.3.2 Block Similarity Matrix

Let us assume that each frame is divided in *N x N* blocks. This will help us separate the picture in many small blocks for better performance of the algorithm. The application of the following algorithm for each pixel solely, not in blocks, is not possible since it makes it more complex and dramatically increases the time needed for implementing the method. For the background estimation, a sequence of frames with total length L has been used.

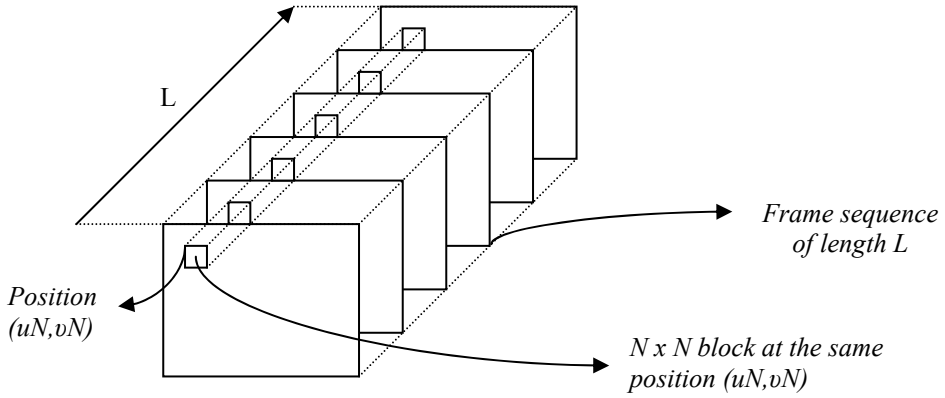


Figure 2.1: To obtain initial background model. Estimation of the history map on an *N x N* block, during a frame sequence of *L* frames.

Moreover, we assume that $f_i(x,y)$ is the luminance of pixel (x,y) in input frame *i*. We also assume that $f_i \in [0;1]$. Hence, for each block (u,υ) with top left pixel at position $(uN,\upsilon N)$, we calculate a symmetric similarity matrix $M^{(u,\upsilon)}$ of size *L x L* using the below equation:

$$M_{a;b}^{(u,\upsilon)} = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\left|f_a(uN+i,\upsilon N+j) - f_b(uN+i,\upsilon N+j)\right| \tag{2.1}$$

This states that each matrix element $M_{a;b}$ is set to the *sum of absolute differences (SAD)* measure and contains similarity of two blocks at the same spatial position, but at different timing, at frame *a* and frame *b*.

Figure 2.2 represents an example of a sequence of frames. Observing a small image block *N x N* over time, three classes of content can be distinguished:

- Static Background
- Moving Foreground
- Foreground, which appears static within a small time range.

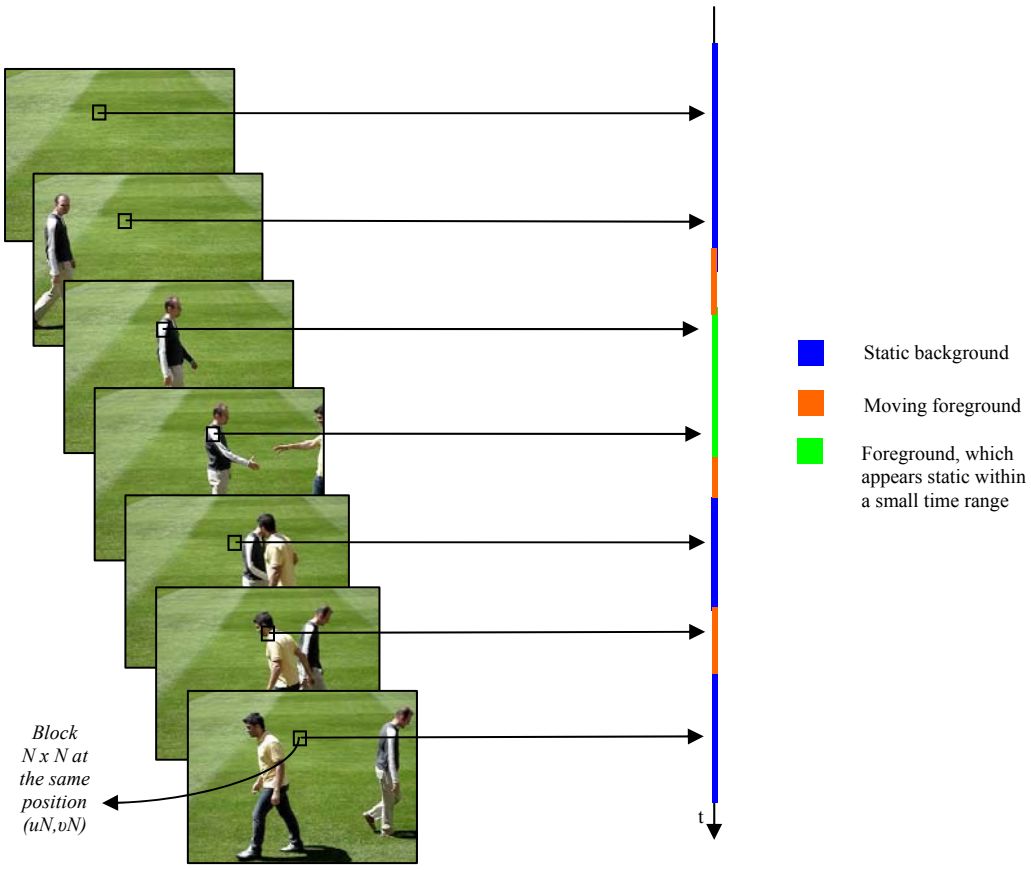


Figure 2.2: The three of content in a small image block over time.

It is obvious that the moving areas cannot be part of the background. For this reason, frames with moving contents can be ignored, as reported in figure 2.3.

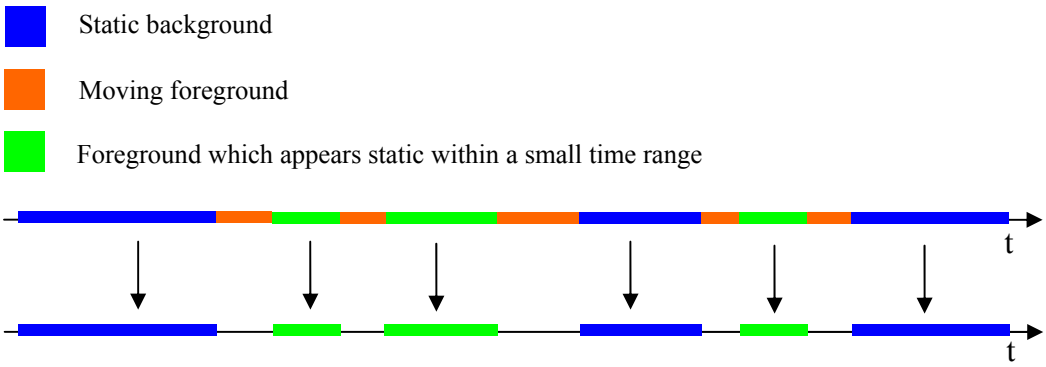


Figure 2.3: Moving Content Exclusion: *Ignore frames with moving content because moving areas cannot be part of the background.*

The result of this process is the development of a similarity matrix $M_{a;b}$ with dimensions *L x L* for each block, comprising values which belong to the subset *[0;1]* (see figure 2.4). For the same spatial position of each frame and different timing, the similarity matrix $M_{a;b}$ is roughly equal to 1, when the similarity of the block is low. On the other hand, when the

similarity is high, $M_{a;b}$ is nearly equal to 0. It is easily observable that the diagonal of the table will present low values, since in each point the sum of the absolute differences (SAD) will be zero. Herein, the matrix is separated into two parts: the stationary elements (small difference values), and the non-stationary elements (large difference values). For more details, see paragraph 2.3.4.

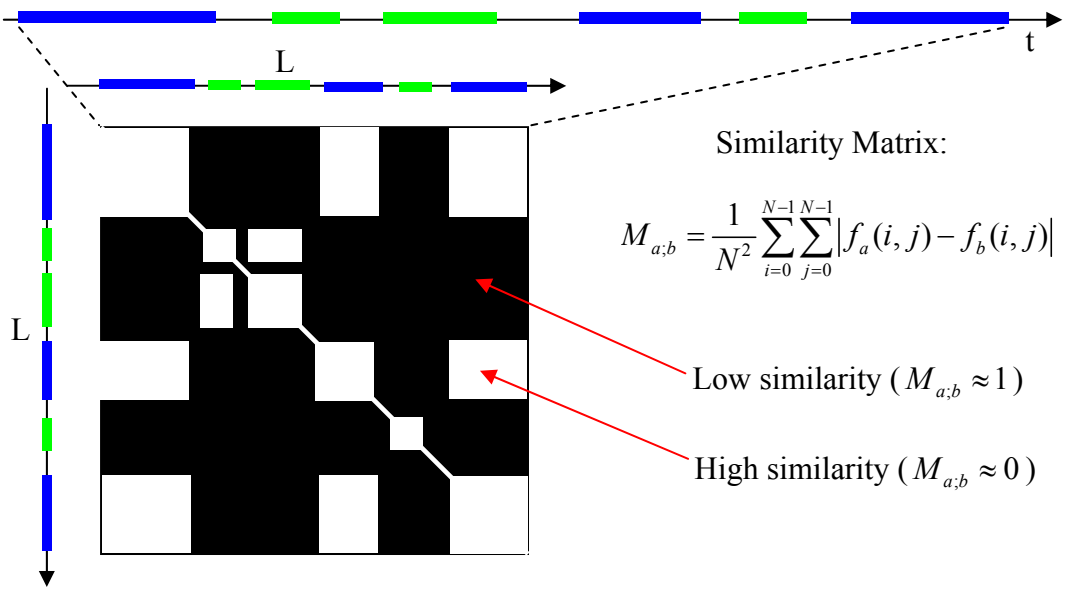


Figure 2.4: Structure of a block similarity matrix. Elements with high values of similarity are shown in white, low values in black areas in the matrix.

2.3.3 Matrix Decomposition

To identify the periods in which only the background is visible in a block, the matrix is separated, as already mentioned, into two parts: the stationary elements (small difference values), and the non-stationary elements (large difference values). The next step is to define a new set $T^{(u,v)}$, which is a subset of set $\{1,...,L\}$, and represents the frames in which block includes only contents of the background. The way in which the background contents and moving elements are separated is based on a cost function that is analysed below. Therefore, an element of the table $M_{a;b}$ is considered as stationary if $a,b \in T$.

A subset T is chosen such that the stationary elements are as small as possible while the non-stationary elements are as large as possible. The cost function C is optimised using the equation below:

$$\min_T C = \min_T \left[\sum_{a,b \in T} M_{a;b} + \sum_{a \notin T \cup b \notin T} (1 - M_{a;b}) \right] \tag{2.2}$$

An iterative process is implemented during the optimization process. Firstly, T is initialised (see paragraph 2.3.5 for more details). Thereafter, the cost value C is calculated as the result of the addition or removal each of the input frames in the set of T. If the cost value C decreases by adding or removing the frame, then T is modified accordingly. In the opposite

case, where the cost function C increases or in the marginal case where it remains stable, T also remains immutable. Optimization is stopped when none of these changes can further decrease the cost.

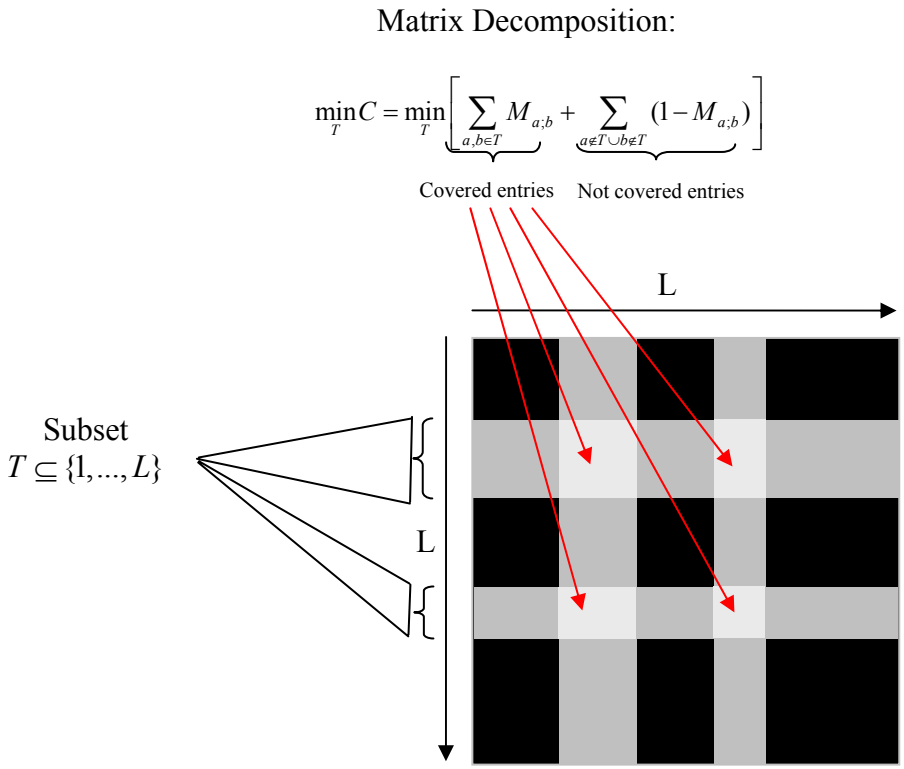


Figure 2.5: Matrix Decomposition example

Instead of the naive way of computing the cost by summing over the complete matrix, the cost difference is computed by summing only over a single matrix row. In the case of adding a frame k into subset T, the cost difference is given by:

$$\Delta C_{+k} = 2\left(\underbrace{\sum_{a \in T} M_{a;k} + \sum_{a \notin T} \left(1 - M_{a;k}\right)}_{\text{new costs}} - \underbrace{\sum_{a \in \{1,\ldots,L\}} \left(1 - M_{a;k}\right)}_{\text{old costs}} \right) \tag{2.3}$$

$$= 2\left(\sum_{a \in T} \left(2M_{a;k} - 1\right) \right) \tag{2.4}$$

Apparently, in the case of removing a frame k from T, it is the negative value: $\Delta C_{-k} = -\Delta C_{+k}$.

The above matrix decomposition process converges to a local minimum close to the initialisation. Therefore, an initialisation near to the global minimum must be chosen. However, the correct background periods do not necessarily correspond to the global minimum. For example, if the sequence contains many foreground objects, which are visible during most of the time, the global minimum may correspond to those periods in which foreground objects are visible. Two additional steps are inserted, before the optimisation one,

in order to solve this problem. Firstly, periods from T are excluded for which motion in the block is observed. Since it is assumed that the camera is still, camera motion has already been compensated. Hence, moving regions do not belong to the background. Secondly, the correlation of background periods between neighbouring blocks is exploited. Both steps are described in the following two sections.

2.3.4 Integration of Motion Information

It is obvious that the moving areas cannot be part of the background. Therefore, it is necessary to determine a criterion for dividing blocks into moving and stationary elements. A block-matching algorithm is used to carried out motion estimation for each block. Stationary elements are defined as those elements with minimum SAD matching error less than 0.1 (10% of the maximum possible value) and are equalised with 0. Similarly, the elements with SAD more than 0.1 are defined as non-stationary elements and the corresponding matrix row and column entries are artificially set to 1.

- We assume that block content is non-moving if:

$$SAD(dx, dy) < 10\% \cdot SAD(0,0) \quad where\ dx\ or\ dy \neq 0 \tag{2.5}$$

- And it is moving if:

$$SAD(dx, dy) > 10\% \cdot SAD(0,0) \quad where\ dx\ or\ dy \neq 0 \tag{2.6}$$

This prevents the optimization algorithm from selecting block in this frame as a background block.

2.3.5 Background Periods Prediction

Observing carefully a sequence of frames, we can infer some important properties. One of these is based on the movement of an object. More specifically, if an object moves across a block, it will most probably also move across a neighbouring block as well during a certain period of time. Thus, the previously calculated $T^{(u-1,\upsilon)}$ and $T^{(u,\upsilon-1)}$ is used to calculate $T^{(u,\upsilon)}$, while initialising the optimisation process. If an input frame ξ is contained in $T^{(u-1,\upsilon)}$ and $T^{(u,\upsilon-1)}$, it is also included in $T^{(u,\upsilon)}$. If it is only contained in one of them, it is included randomly. The solutions of the block above or to the left are used to derive predictions at the left and top border respectively. The first left-bounded block is initialised with all input frames active in $T^{(0,0)}$. Usually, image activity is concentrated in the centre of the image and hence it is reasonable to assume that the border contains mainly background content.

The spatial prediction scheme has two advantages:

- Provide an accurate initialization of the optimisation, leading to fast convergence.
- The prediction helps to select the correct local minimum, even when the object is visible for more time than the background.

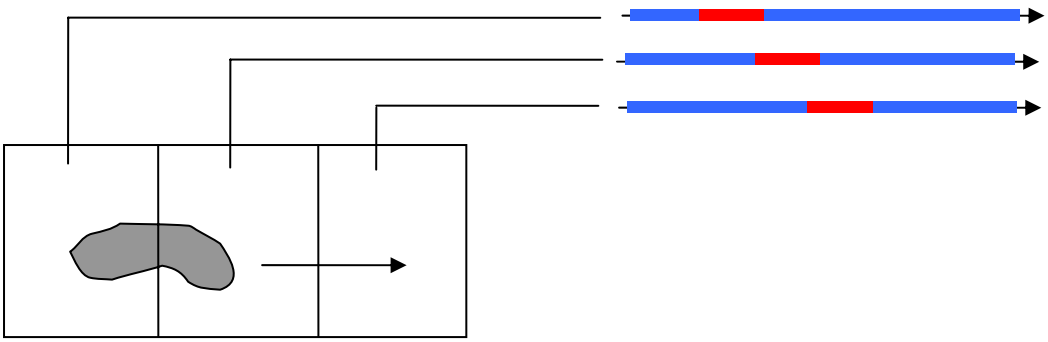


Figure 2.6: Spatial Background Period Prediction: there is a strong correlation of background times across neighbouring blocks. Therefore, we use solution of neighbouring block as initialization of optimization

2.3.6 Required Processing Time

A basic consideration in real time applications is the processing time. The most important parameter on the time consuming for the background modelling is the number of frames in the training sequence. As the training sequence includes more frames, the processing time increases. Therefore, we should be very careful in the number of frames used for the background estimation. In this project, *L=300 frames* have been used, which correspond to *10 sec* of a movie. Obviously, this time leads to delays. Bearing in mind that the video recording can be started earlier and the background can be estimated after 10 sec, the goal for applications in real time can easily be accomplished.

2.4 Maintenance Background Model

The maintenance of a background is the most difficult part of background subtraction. Real-time background maintenance is of vital importance and a very challenging issue for surveillance applications, since the monitoring area may change over time. For example, during the sunset there are illumination changes in both colour and brightness, or other kind of changes, such as a car that comes into the scene and parks. Even though these pixels should not be considered as a part of the scene, the stationary pixels should play the role of background for detecting motion of a silhouette. Several studies focused on this area such as [3], [4], [6], [15]. In this study, an ideal background maintenance approach for surveillance systems, proposed in [15], has been implemented. This approach is divided in three levels:

1. *Pixel-level* processing makes the preliminary classifications of foreground versus background and also handles adaptation to changing backgrounds. It avoids many of the common problems immediately: moved objects, time of day, waving trees, camouflage, and bootstrapping. All pixel-level processing takes place at each pixel independently and ignores information observed at other pixels.
2. The *region-level* considers inter-pixel relationships that might help to refine the raw classification of the pixel level; the foreground aperture problem is thus avoided.
3. The *frame level* addresses the light switch problem: it watches for sudden changes in large parts of the image and swaps in alternate background models that explain as much of the new background as possible.

2.5 Results and Discussion

We have applied our algorithm in a variety of video sequences. When one or more objects enter the scene and continue moving, the background is recovered without error. The algorithm has been applied to special sequences containing very difficult background objects and when more than one people were in the scene without moving. This made background estimation even more difficult because these objects were visible for more time than the background. Part of the background is impossible to reconstruct since it is never visible during the whole sequence. Beside all these difficult cases, the proposal algorithm implements background estimation in a video sequence without any blurring.

In order to evaluate the quality of the reconstructed background image and to compare the proposed method with others, the following distortion measures have been introduced: Mean Square Error (MSE) and Peak Signal to Noise Ratio (PSNR). We choose specific video files, in which we could estimate the real background because they had frames without an object in the scene. Therefore, we compared the produced background with the real background using MSE and PSNR. These measures were calculated for several reconstruction algorithms and various videos. In the proposed algorithm, the reconstructed background has performed better results with MSE close to 14 and PSNR up to 36.55 dB. The average values of MSE and PSNR of each method are listed in Table 2.1.

Table 2.1: Calculated MSE and PSNR between reconstructed background and real background

	MSE	PSNR
Average	62.9405	30.1415 dB
Median	49.8001	31.1585 dB
Proposed Algorithm	14.3694	36.5564 dB

2.6 Summary and Conclusions

Background estimation comprises a useful first step in our attempt to detect moving objects. The proposal algorithm represents an important component for the implementation of the next step that is the detection of moving objects, using background subtraction (see Chapter 3). The various moving objects will be detected by differencing the current frame with a reference background, and then track them.

An effective algorithm for background estimation is very important since an eventual faulty evaluation will not permit the correct detection of the moving objects. The proposal method is obtainable even for complex video sequences, providing better results than any existing method. The limited demands in processing time and the low complexity permit us use this method in real time, something that was precondition for this particular implementation.

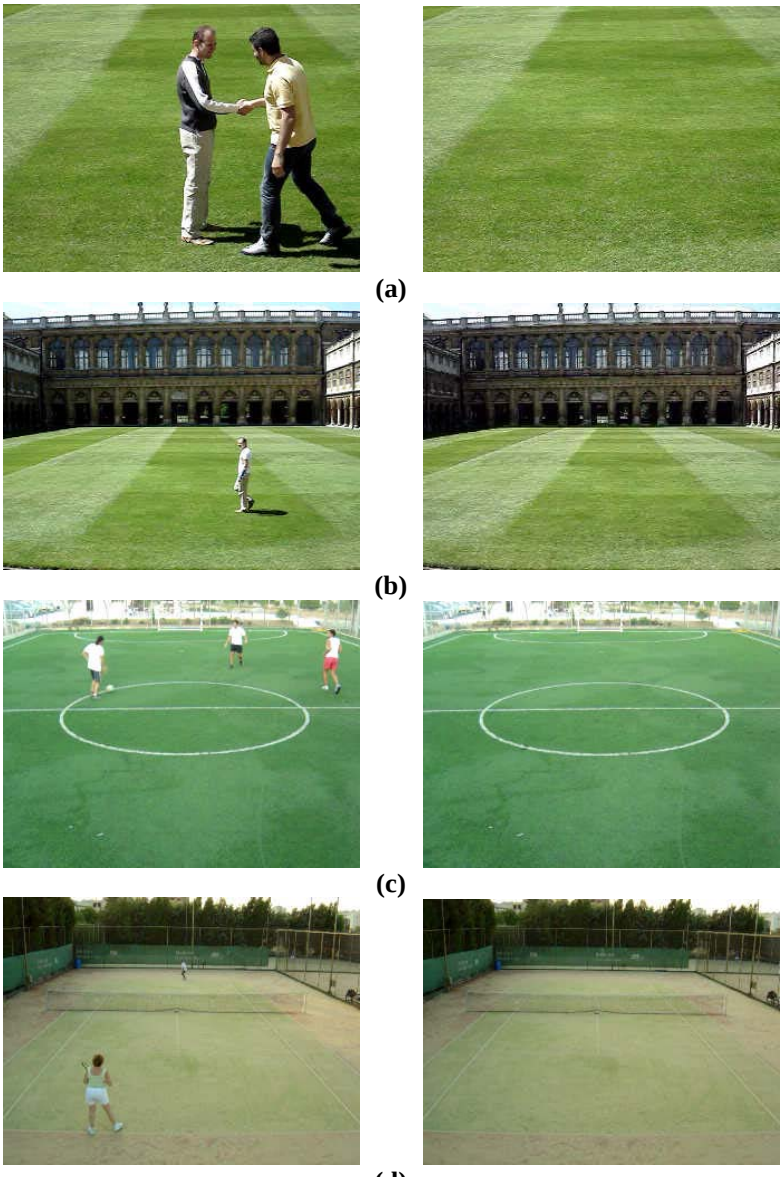


Figure 2.7: Background model initialization results produced by the proposed method: **(a)** and **(b)** Trinity College, Cambridge, UK, **(c)** Fitness-Land, mini-football campus, Limassol, Cyprus, **(d)** Famagusta tennis court, Limassol, Cyprus.

Chapter 3
Foreground Region Detection

3.1 Approach

Based on adaptive background subtraction, a running statistical background value of the intensity of each pixel is maintained for the foreground region detection. The background can be reconstructed as explained in Chapter 2. Background subtraction is a very popular approach for foreground region detection, especially with static backgrounds. It attempts to detect moving regions in an image by differencing between current image and a reference background image in a pixel-by-pixel fashion. To track moving silhouettes from the image in each frame, the background subtraction method and a tracking algorithm (based on silhouette correlation) are adopted. The main assumption made here is that the camera is static, therefore the background is static, and so the 'differencing' between current frames and the background is the moving object.

3.2 Previous Work

Occasionally, several studies used the background subtraction method that effectively estimates the foreground. Most proposed studies paid attention on the difference of the current pixel, with reference a background pixel, using 'differencing' techniques, such as the luminance or brightness difference, or the colour difference in RGB, YUV space. A foreground model in HSI space is proposed in [14], because the separation between the foreground and background clusters is high. Nevertheless, this method creates a noise image with player regions, and especially when the player's legs are being fragmented. This causes serious problems in the localisation of the real position of people, since the real position of the player is assumed the position where feet are in contact with the floor.

These methods have to face certain important problems, starting from the difficulty to estimate the foreground when the moving object has the same colour as the background. Another important problem is the abrupt change in the intensity of light, which is mainly owed on indoor turning off or turning on, or in some other parameters such as clouds blocking the sun. Other important problems are the shade of the moving object, because it increases the size of the object and changes its coordinates. Moreover, the shade creates problems in the export of the outline of people, as well as in the motion analysis.

The properties of the camera should be considered. In this research, a camera based on RGB colour map is used. Camera's R, G and B channels are assumed to have Gaussian noise. However, jitter or small 'micro-motions' such as leaves moving on a tree or waves, can cause some background pixels to violate the Gaussian assumption. Actually, this motion adds 'salt

and pepper' noise, which is small areas of pixel in the image. Therefore, an application based exclusively on the difference in a pixel-by-pixel fashion on RGB space will not be efficient. The failure of this application is attributed to the various problems created by the shadows of the moving objects.

3.3 Foreground Detection

In this work, the non-parametric method for background subtraction proposed in [17] has been implemented.

3.3.1 Shadow Detection

As aforementioned, moving regions in an image are detected by differencing between the current image and a reference background image. The pixel is flagged as potentially containing a foreground region, when the value of a pixel in a new image differs significantly from the background value.

A method based in chromaticity change is implemented in this project. Chromaticity exploits the fact that an area cast into shadow that often results in a significant change in intensity without much change in chromaticity [17], [28]. Therefore, we assume that any significant intensity change without significant chromaticity change could have been caused by shadow. Chromaticity is computed as:

$$r_c = \frac{R}{R+G+B} \tag{3.1}$$

$$g_c = \frac{G}{R+G+B} \tag{3.2}$$

$$b_c = \frac{B}{R+G+B} \tag{3.3}$$

where $r_c + g_c + b_c = 1$ and r_c, g_c, b_c correspond in red, green and blue chromaticity, respectively. The use of chromaticity coordinates has the advantage of being more insensitive to small changes in illumination such as introduction of shadows. As described before, the background subtraction technique can be utilised with any colour space.

Next target is to find a suitable criterion in order to detect moving regions in an image, by differencing current image with a reference background image. This criterion must classify each pixel as a background pixel or a foreground pixel.

- If $|cr_c - br_c| > Th$ && $|cg_c - bg_c| > Th$ && $|cb_c - bb_c| > Th$. (3.4)
  - *Then the current pixel is foreground*
- *Otherwise*
  - *The current pixel is not foreground*

where *Th* is a threshold, cr_c is the red chromaticity of the current frame, and br_c is the red chromaticity of the background. Also, cg_c, cb_c, bg_c, bb_c represents the green and blue

chromaticity of the current frame and background, respectively. The selection of a suitable threshold is very important for the correct classification.

Even though the use of chromaticity coordinates helps suppressing shadows, it loses lightness information, which is its major drawback. Lightness is related to the difference in whiteness, blackness and greyness between different objects [29]. For example, consider the case where the moving silhouette is white coloured and moves against a gray background, then there is no colour information because both white and gray have the same chromaticity coordinates. Therefore, the moving object will not be detected.

To address this problem a lightness measure ($s = R + G + B$) is used at each pixel. Consider the case where the background is completely static, and let the expected value for a pixel be (r_c, g_c, s). Also, assume that this pixel is covered by shadow in frame t and let (r_{ct}, g_{ct}, s_t) be the observed value for this pixel at this frame. Then, $a \le \frac{s_t}{s} \le 1$. That is, it is expected that the observed value, s_t, will be smaller up to a certain limit, $a \cdot s \le s_t$ since the colour are darker. This corresponds to the intuition that at most *(1-a)%* of the light coming to this pixel can be reduced by a target shadow.

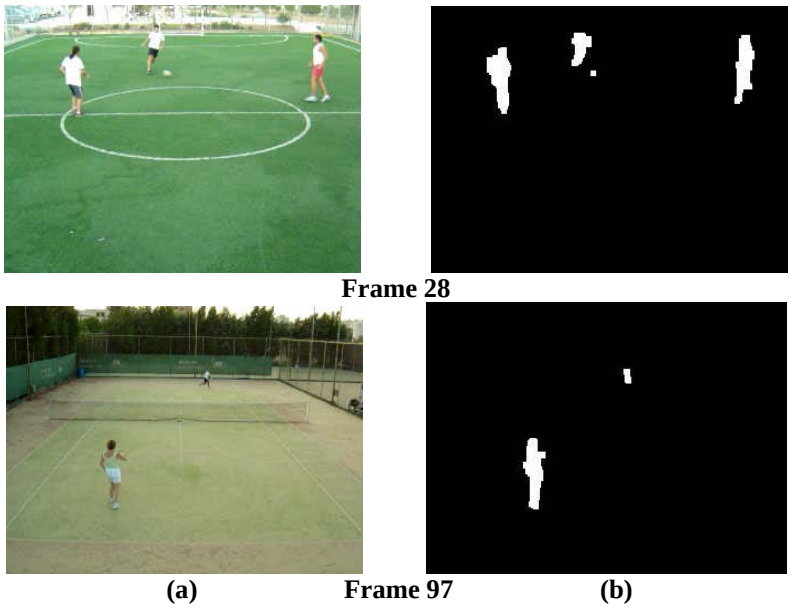


Figure 3.1: The differencing between current image and a reference background image. **(a)** Input Image, **(b)** detected foreground regions.

3.3.2 Processing the Shadow Properties

While detecting a foreground region, there may be erroneous pixels detected and holes in object features. Therefore, the first pre-processing step is to clean up anomalies in the detected regions. A morphological filter (opening and closing) is implemented in order to eliminate the noise. This process removes any small holes in the silhouette and smoothes out any interlacing anomalies. Then the outline of the silhouette region is extracted using a border following algorithm as shown in Figure 3.2. The outline of the human silhouette region can be used in various applications such as recognition of human shape compared to other shapes and the study of human movement. Further details are reported in [5] and [9].

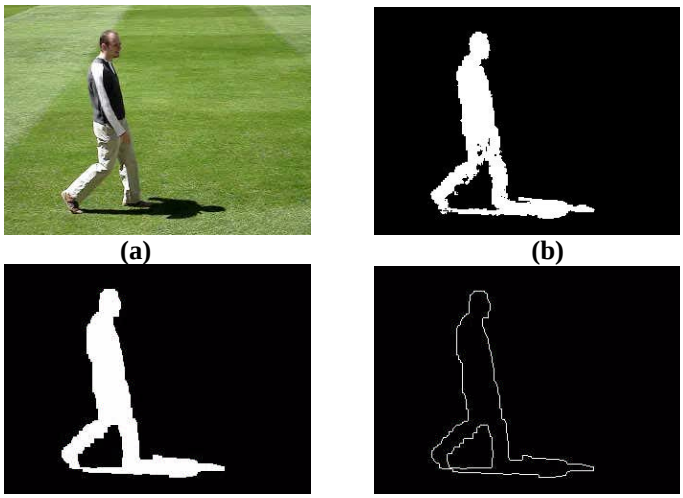


Figure 3.2: Foreground region pre-processing. **(a)** Current frame with moving object, **(b)** foreground detected region in binary pattern, **(c)** the moving foreground region after morphological filter, **(d)** foreground region's border extraction.

In order to achieve object tracking (see Chapter 4) it is essential to export some characteristics of each blob. These are the centroid coordinates, the bounding boxes, the covered areas and the ground plane coordinates. The term *'ground plane coordinates'* is used to refer to the centre point of the blob that is found in the lowest point and represents the reported place of human in the ground field (see figure 3.5).

The centroids of the target image boundary (x_c, y_c) are determined:

$$x_c = \frac{1}{N_b}\sum_{i-1}^{N_b} x_i \tag{3.5}$$

$$y_c = \frac{1}{N_b}\sum_{i-1}^{N_b} y_i \tag{3.6}$$

where (x_c, y_c) is the average boundary pixel position, N_b is the number of boundary pixels, and (x_i, y_i) is the pixel of the boundary of the target (see figure 3.4).

Here is some other important information that are used and reported in figure 3.4:

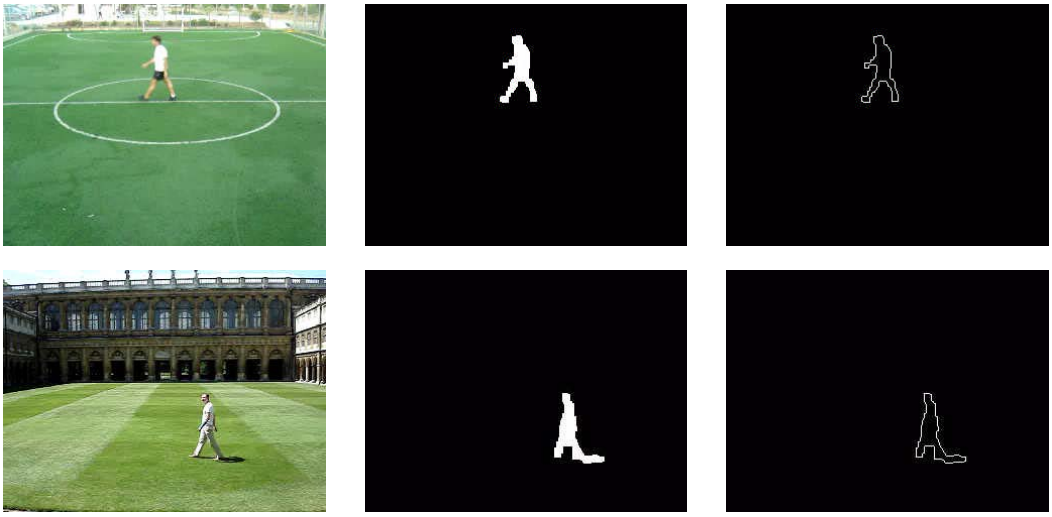
- (x_b, y_b), the coordinates of the blob highest left point,
- BW_x, the distance between the more left and more right point,
- BW_y, the distance between the highest and the lowest point,
- *Area*, the blob covered area.

An identification number, *id*, is also given in order to determine the player. Before object tracking, the *id* value is sorted according to the blob size. In section 4.3, the *id* correlation process, also known as tracking, is analysed in order to establish correspondence between

objects across frames. In section 4.3 an example of a three frames sequence is also reported, in which the characteristics of each blob are correlated across frames.

Hence, each object is presented as:

$$\mathrm{X} = \left(id, x_c, y_c, x_b, y_b, BW_x, BW_y, Area \right) \tag{3.7}$$



(a) **(b)** **(c)**

Figure 3.3: Foreground region pre-processing. **(a)** Current frame with moving object, **(b)** The moving foreground region after morphological filter, **(c)** Foreground region's border extraction.

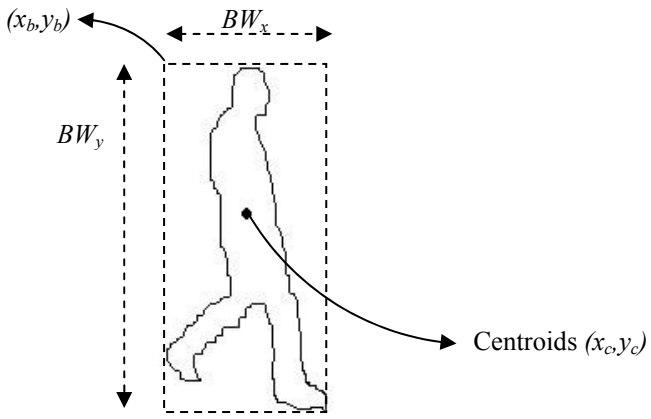


Figure 3.4: The silhouette boundary of a detected foreground region. Centroids (x_c,y_c), the coordinates of the blob highest left point (x_b,y_b), the distance between the more left and more right point BW_x, the distance between the highest and the lowest point BW_y are also presented.

The ground plane coordinates of a player represent the position of the feet that contacts the floor. These coordinates are preferred instead of the players' centroids, and wish to determine for use when projecting blobs from real image to a ground plane platform. Image points can be projected to ground plane coordinates by calibrating the image plane. However, when calculating players' world position from the image, it is assumed that their feet are in contact with the floor. It is important to be mentioned that, in our implementation, ground

plane coordinates are calculated from centroids after object tracking. This information is used in chapter 5. The coordinates of the lowest centroid point (x_g, y_g) can be calculated in the following way:

$$x_g = x_c \tag{3.8}$$

$$y_g = y_b + BW_y \tag{3.9}$$

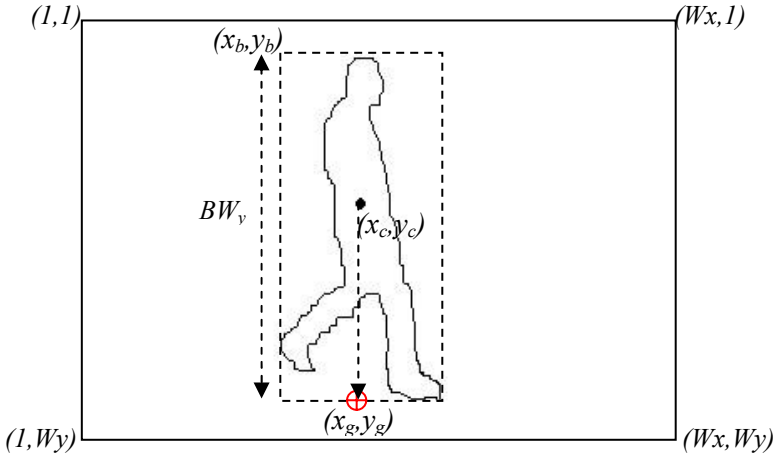


Figure 3.5: Sample representation of ground plane coordinates *(xg,yg)*, the contact point of the players' feet with the floor.

In order to reduce the effects of the *'salt and pepper'* noise, blobs with covered area *(Area)* less than a threshold (in this project threshold is equal to 100 pixels) are not further considered in the work.

3.4 Summary

The proposed method, which is based on the chromaticity change of the corresponding foreground and background pixels, succeeded in finding an effective way of estimating moving objects. Specific problems, such as suddenly light changing, shadow detection and foregrounds with the same colour as the background, were faced. Thereafter, a pre-processing step has been applied, in order to cope with cases such as *'salt and pepper'* noise, spurious pixels and anomalies in the detected areas, improving the image using a morphological filter. For each moving blob, the centroid's coordinates, the covered areas, the bounding boxes and the ground plane coordinates have been calculated. These foreground characteristics are very useful in order to accomplish tracking for each moving object, as analysed in chapter 4.

Chapter 4
Object Tracking

4.1 Introduction

The main objective of object tracking is the establishment of correspondence between objects across consecutive frames. It is assumed that the regions can either enter or exit the scene or can be occluded by other regions. Regions carry information like shape and size of the silhouette, while colours carry information on the location estimated for each object. The characteristics of each object are presented in the following equation:

$$\mathrm{X} = \left(id, x_c, y_c, x_b, y_b, BW_x, BW_y, Area \right) \tag{4.1}$$

where *id* is the identification number, (x_c, y_c) are the centroid coordinates, (x_b, y_b) are the coordinates of the blob highest left point, BW_x is the distance between the more left and more right point, BW_y is the distance between the highest and the lowest point, and *Area* is the blob covered area. In order to achieve correspondence, these blobs information is used in addition to the predicted positions returned by a condensation filter.

4.2 Previous Work

Currently there are several popular methods for tracking moving objects. The main difficulties that the tracking procedure has to face are focused on the temporal occlusions of the targets and the splitting of the blobs aiming at separating or isolating the players.

A number of research projects were published on human detection and tracking, were a sufficient amount focused on football player tracking [10], [14], [34], [35], [37], [38]. Usually, in tracking information such as texture, colour and shape are used [28]. Colour based template matching is one of the most popular methods to deal with the occlusion problems [37], [38]. Some other studies estimated motion and player position using Kalman [10] or condensation filters [14]. Kalman filter [11] has been used in many tracking applications due to its computational efficiency and its ability to predict future states.

Multiple tracking using a single camera is a very popular method, but suffers by the problem of occlusion. In team sports, it is difficult to track players using a single camera. That happens because in team sports, such as football and basketball, occlusion occurs in many cases and tracking often fails when players cover other players.

Some efficient, but very complex, studies used multiple cameras for human tracking, because they solve the problem of players' occlusion and show their positions in 3D space [10], [34], [36]. Inner-camera operation is performed independently in each camera to detect the features of the extracted player regions in each camera image. Although these methods require dedicated static cameras, it increases the overall field of view, minimises the effects of dynamic occlusion, provides 3D estimates of ball location, and improves the accuracy and robustness of estimation due to information fusion [10], [13]. All these studies provide an effective method of tracking and player positioning, but the implementation complexity is high. In addition to that, it is difficult to convince each association to approve the placement of the static cameras in the field. Furthermore, methods using multiple cameras are not able to process old video files, but only new files captured by these specific cameras.

Football related work has been inspired by several different ambitions, including annotation, action recognition, game reconstruction, and the need for game evaluation. In addition, a method for parallel tracking of all soccer players using multiple view images taken by static cameras is also referred in [35]. Each camera detects the players' positions. These positions are projected onto the virtual upper-view image via homography of the ground between the camera image and the upper-view image. By integrating the detected positions in all cameras, all players can be tracked.

The combination of a method that uses blob's information as well as the prediction of the next possible position of a player, using the Kalman or condensation algorithm, could be an efficient approach for players tracking. A new method of condensation, which is proposed in [12], allows the propagation of conditional densities over time. It has been used with contour tracking to track an object through cluttered scenes. However, it fails tracking multiple targets. A probabilistic exclusion principle that allows multiple trackers has been introduced in [8]. If several one-body trackers are employed, each with the same tracking algorithm, then two or more can coalesce onto the same target for which their model best fits [14]. It is an excellent approach where the use of multiple fixed cameras is avoided without considering the occlusion problem.

4.3 Object Tracking

The target of object tracking is the establishment of correspondence between moving blobs across frames. This can be achieved by correlating the identification numbers, *id*, of each blob during the frames sequence. Object tracking can be accomplished using a cost function of blob information such as the centroid coordinates, the covered area, the velocity and the colour. A condensation filter is also implemented in order to cope with complex cases such as when regions enter/exit the scene and when they are occluded by other regions. The tracking method used is analysed in section 4.3.1 and 4.3.2.

4.3.1 Tracking using cost function

Extra information is needed in order to create the cost function. This cost function can calculate the correlation between blobs across frames. Each blob is determined as $\mathrm{X} = \left(id, x_c, y_c, x_b, y_b, BW_x, BW_y, Area \right)$, and each region is defined by:

- the 2D coordinates of the centroid, P (P values refers to (x_c, y_c)) ,

- a ratio between the total number of foreground pixels *(Area)* and the size of the bounding box ($B = BW_x \cdot BW_y$), $R = \frac{Area}{B}$,
- The colour/gray level characteristic, $D = (R + G + B)$ in case that RGB space is used.
- The regions, for which correspondence has been established, have also an associated velocity, $V_t = P_t - P_{t-1}$, where *t* is the current frame and *t-1* is the previous frame.

In frame *t* of a sequence, there are *M* regions with centroids P_i^t (where *i* is the number of regions) whose correspondences to the previous frame are unknown. There are *K* regions with centroids P_L^{t-1} (where *L* is the label) in frame *t-1* whose correspondences have been established with the previous frames. The number of regions in frame *t* can be different to the number of regions in frame *t-1*. They might be either more due to entries or less due to exits or occlusion.

The task is to establish correspondence between regions in frame *t* and frame *t-1*. The minimum cost criteria are used to establish correspondence. The cost function between two regions is defined as:

$$C_{L_i} = \left| \frac{P_L^{t-1}}{P_i^t} + \frac{R_L^{t-1}}{R_i^t} + \left| \frac{Area_L^{t-1} - Area_i^t}{100} + \frac{V_L^{t-1}}{10} \right| + \frac{D_L^{t-1}}{D_i^t} \right| \tag{4.2}$$

where *L* is the labels of region in frame *t-1*, *i* is index of non-corresponded region in frame *t*.

The cost is calculated for all *(L,i)* pairs and correspondence is established between the pair with the lowest cost, being less than a certain threshold (see figure 4.1 and 4.2). All the parameters of each region are updated using linear low pass filter prediction models as illustrated in the following equations:

$$Predicted\ Position: PP_n = P_{n-1} + (t_n - t_{n-1}) \cdot V_{n-1} \tag{4.3}$$

$$Predicted\ Velocity:\ PV_n = a \cdot V_{n-1} + (1-a) \cdot \frac{(P_n - P_{n-1})}{t_n - t_{n-1}} \tag{4.4}$$

$$Predicted\ Ratio:\ PR_n = a \cdot R_{n-1} + (1-a) \cdot R_n \tag{4.5}$$

$$Predicted\ Colour:\ PD_n = a \cdot D_{n-1} + (1-a) \cdot D_n \tag{4.6}$$

The process on correspondence continues until no pairs left or the minimum cost rises above the threshold. Therefore, correspondences between all regions in frames *t-1* and *t* have been achieved apart from the cases due exits or occlusions, or new entries in the scene. The position and the predicted velocity of the exiting/entering region from/to the scene is used in order to determine whether a region has been exited/entered the scene. If this is not the case, then a check for occlusion is made.

At the tracking process, the centroid of the detected foreground region in the sequences is stored in a trajectory map. The stored data is used to find the trajectory of each foreground region in the scene. When any foreground object in tracking is occluded to any stationary region (such as passing behind a building), or temporarily occluded by other moving silhouettes as they cross, the detected data of that object may not be obtained at the low level processing. At these kind of situations, with the aid of the previous tracked data obtained from trajectory a high-level implementation procedure is triggered to estimate the possible position of that object. This high-level implementation procedure predicts the next state of the occluded player, based on condensation tracking algorithm. This is analysed in section 4.3.2. At the subsequent frames, its confidence is reduced if the low-level data about it is not obtained. An object is considered lost if its confidence drops below a selected threshold, and it is aborted from the tracking list stored in the trajectory map. High confidence objects are considered the ones that have been tracked for a reasonable time period. These will persist for several frames, if an object is occluded but then reappears and the foreground object tracker will reacquire it.

Tables 4.1 and 4.2 present an example, showing the cost between 3 blobs in two consecutively frames and the correlation of the identification numbers of each blob.

Table 4.1: An example presenting the cost functions' results of our data set in time *t, t-1* and the correlation between the *id* of each blob.

		Frame *t-1*		
		Id = 1	Id = 2	Id = 3
Frame *t*	1st (Id=1)	3.2541	10.3275	3.5414
	2nd (Id=3)	13.6990	4.5708	3.0112
	3rd (Id=2)	22.3150	3.3418	10.5655

Table 4.2: An example presenting the cost functions' results of our data set in time *t+1, t* and the correlation between the *id* of each blob.

		Frame *t*		
		Id = 1	Id = 3	Id = 2
Frame *t+1*	1st (Id=1)	3.1120	4.2581	10.5029
	2nd (Id=3)	13.8996	3.1151	4.0503
	3rd (Id=2)	22.4699	11.5047	3.0528

It is clearly observed that the cost value between two related blobs is marginally closed to 3. Therefore, we can define a cost value that could constitute a threshold in order to segregate whether a new blob enters the scene. This can be useful in order to avoid cases where new entries return minimum cost.

4.3.2 Tracking Optimisation using Condensation algorithm

A condensation tracking approach based on Kalman filter is used, in order to cope with complex cases where moving objects are occluded by other objects. In this dissertation, a method based in [14] is employed.

The Kalman filter for tracking has been widely used in previous research. It is an efficient method that predicts the next possible position, but fails on bouncing and when the object changes abruptly its movement direction. Given that players have sudden changes in velocity and in their motion direction, an alternative method is achieved called condensation algorithm.

Condensation algorithm comes out from **Con**ditional **Dens**ity Propag**ation.** The basic features of condensation tracking are: at each time step, it maintains multiple hypotheses of position (this number of hypotheses is the computational cost); it updates using new data; and it selects hypotheses probabilistically. These are the reasons of why, condensation algorithm can cope with very noisy data and process state changes.

Each frame might have many blobs. In the case of football, these blobs can be reached up to 25, 11 players a side and 3 referees. Below a simple model that predicts each blob's information of a current frame *t* from the previous frame *t-1* is presented:

$$x_{c,t}^{i} = x_{c,t-1}^{i} + \varepsilon_x \quad (4.7), \qquad y_{c,t}^{i} = y_{c,t-1}^{i} + \varepsilon_y \quad (4.8)$$

$$x_{b,t}^{i} = x_{b,t-1}^{i} + \varepsilon_{bx} \quad (4.9), \qquad y_{b,t}^{i} = y_{b,t-1}^{i} + \varepsilon_{by} \quad (4.10)$$

$$BW_{x,t}^{i} = BW_{x,t-1}^{i} + \varepsilon_{BWx} \quad (4.11), \qquad BW_{y,t}^{i} = BW_{y,t-1}^{i} + \varepsilon_{BWy} \quad (4.12)$$

$$Area_{t}^{i} = Area_{t-1}^{i} + \varepsilon_{Area} \quad (4.13)$$

for $i = \{1,...,N_j\}$, where N_j are the blobs of the current frame, but only those for which are considered occluded and the high level procedure is activated. ε_x , ε_y, ε_{bx} and $\varepsilon_{by} \sim N(0,\sigma_1)$ with σ_1 typically taken to be of the order of 0.1m, given that the maximum distance on the ground plane that sports player will move will be in the order of $3\sigma_1$ (0.3m per 1/25 of a second). This allows a player who moves at a speed of $0.3m \times 25s^{-1} = 7.5ms^{-1}$ to be tracked.

The height and width of the bounding box must be allowed to react quickly to the sudden change, due to the fast change in shape of a player (e.g. when they raise arms to attract attention, or open stride when running), thus adding Gaussian noise to the height and width. ε_{BWx}, ε_{BWy} and $\varepsilon_{Area} \sim N(0,\sigma_2)$ with σ_2 *=2 pixels* allows such a change. By doing so, the blobs may no longer correspond to different players. This could happen for example, when one sample locks to another target in close proximity, which is already being tracked.

Such situations are avoided by proposing an optimization method using Kalman filter to predict each sample's information for the next time step, given previous states. Herein, N_j Kalman filters are utilised, one for each player. These are updated using the observed value of the position of each player from the 'best' trajectory map.

Kalman filters have been applied because they address the problem of estimating the position $\mathrm{X}_t = (x, y) \in \Re^2$ of the player at the next discrete time step. A sample linear stochastic difference equation governs this process:

$$\mathrm{X}_t = \mathrm{X}_{t-1} + w_{t-1} \tag{4.14}$$

With a measurement $Z \in \Re^2$ which directly relates to X that is

$$Z_t = \mathrm{X}_k + v_{t-1} \tag{4.15}$$

The independent random variables w_t and v_t represents the process and measurement noise, and have normal probabilistic distributions.

$$p(w) \sim N(0, Q) \quad (4.16), \qquad p(v) \sim N(0, R) \quad (4.17)$$

Currently constant Q (process noise covariance) and R (measurement noise covariance) are used. However, in the future these may be used to assess the certainty of the estimates being made, which will improve the 'trust' in the estimate of the players position from the Kalman filter, compared to the observation Z from the image, when resolving occlusions.

At each time step, a Kalman estimate $\hat{\mathrm{X}}_t = (\hat{x}_t, \hat{y}_t)$ of the position of each player is calculated using information from frame *t-1*. Hence:

$$x^i_{c,t} = (\hat{x}_{c,t} + x^i_{c,t-1})/2 + \varepsilon_x \quad (4.18), \qquad y^i_{c,t} = (\hat{y}_{c,t} + y^i_{c,t-1})/2 + \varepsilon_y \quad (4.19)$$

$$x^i_{bx,t} = (\hat{x}_{bx,t} + x^i_{bx,t-1})/2 + \varepsilon_{bx} \quad (4.20), \qquad y^i_{by,t} = (\hat{y}_{by,t} + x^i_{by,t-1})/2 + \varepsilon_{by} \quad (4.21)$$

$$BW^i_{x,t} = BW^i_{x,t-1} + \varepsilon_{BWx} \quad (4.22), \qquad BW^i_{y,t} = BW^i_{y,t-1} + \varepsilon_{BWy} \quad (4.23)$$

$$Area^i_t = Area^i_{t-1} + \varepsilon_{Area} \quad (4.25)$$

for $i = \{1, ..., N_j\}$, where N_j are not all the blobs of the current frame, but only those for which are considered occluded and the high level procedure is activated. The observed player positions $Z \in \Re^2$ from the 'best' trajectory map are used to update each discrete Kalman filter. This has the effect of grouping the samples corresponding to each player within the condensation algorithm, since each sample is drawn towards the predicted $\hat{\mathrm{X}}_t$ for that player. This prevents the samples for a player splitting up into two or more groups, which might have allowed the 'best' sample for a player to jump between the groups, or lock onto a different player instead.

4.4 Results and Discussion

Tables 4.3 and 4.4 are examples of object tracking for a 3 frames sequence, with 3 moving objects each. Blobs information is also included. During initialisation, blobs were sorted according to the blob size *(Area)*. Thereafter, multiple object tracking correlates *id* across frames, based on the *id* of the previous frame.

Table 4.3: A tracking example presenting the relationship between the *id* and centroid coordinates of blob's position in the real image, during frames 56, 57 and 58. Blobs information is also included.

	Id	x_c	y_c	x_b	y_b	BW_x	BW_y	Area
frame 56:	3	133	40	125	22	16	36	468
	2	262	39	253	20	19	32	421
	1	213	36	205	22	17	24	387
frame 57:	2	260	39	251	19	18	41	490
	1	211	37	205	21	16	35	396
	3	131	40	123	23	17	29	471
frame 58:	2	260	39	251	19	18	41	502
	3	131	41	123	22	17	41	482
	1	211	37	205	22	15	34	383

Using this method, object tracking has been achieved in very complex samples. In table 4.3, the correlation between *ids* of the moving objects across frames, using the cost function, can be observed. The cost is related to the blob's position, the velocity, the covered area and the colour information. All these can be calculated using the equation 4.2, as presented in tables 4.1 and 4.2. Each blob, in the current frame, is correlated with the blob in the previous frame that returns minimum cost. For example, the 3rd blob *(with id=2)* returns *C=3.3414, C=10.5655* and *C=22.3150*. Therefore, the minimum cost is *C=3.3414* so this blob is correlated to the blob of the previous frame with *id=2*. In table 4.3, frame 56, we can see that the blob with *id=3* is localised at the position with centroids *(133, 40)*, blob with *id=2* at *(262, 39)* and blob with *id=1* at *(213, 36)*. In frame 57 the blobs with *id=3, id=2* and *id=1* are localised at the position with centroids *(131, 40)*, *(260, 39)* and *(211, 37)*, respectively.

The covered area of each blob changes due to three main reasons. Firstly, blobs move to the far end or near to the beginning of the pitch, so they appear smaller or larger respectively. Secondly, a possible failure in foreground detection can cause modification of the blobs size. Finally, the partial occlusion of various regions of the body from other parts, such as legs and arms is the third reason.

Figure 4.1 presents the multi-object tracking implementation in frame sequences of football and tennis games. In tennis, figure 4.1 (a), the object tracking can be easily estimated because regions are not occluded frequently, in contrast with football, where occlusion is a common phenomenon. Figure 4.1 (b) shows two players becoming merge (frame 102, 110, 118) and then split (frame 126). It can be clearly observed that the football player with the red colour is still estimated even when the player with the black colour occludes him. The correlation between moving players can be indicated from the fact that the player with the

blue football short is continuously surrounded from the blue bounding box. This pattern is obvious for all players. The segregation between blobs in frame 102 is achieved because an occluded region is determined (from 3 objects remained only 2), so the high-level implementation procedure is activated and the next players' possible position is estimated.

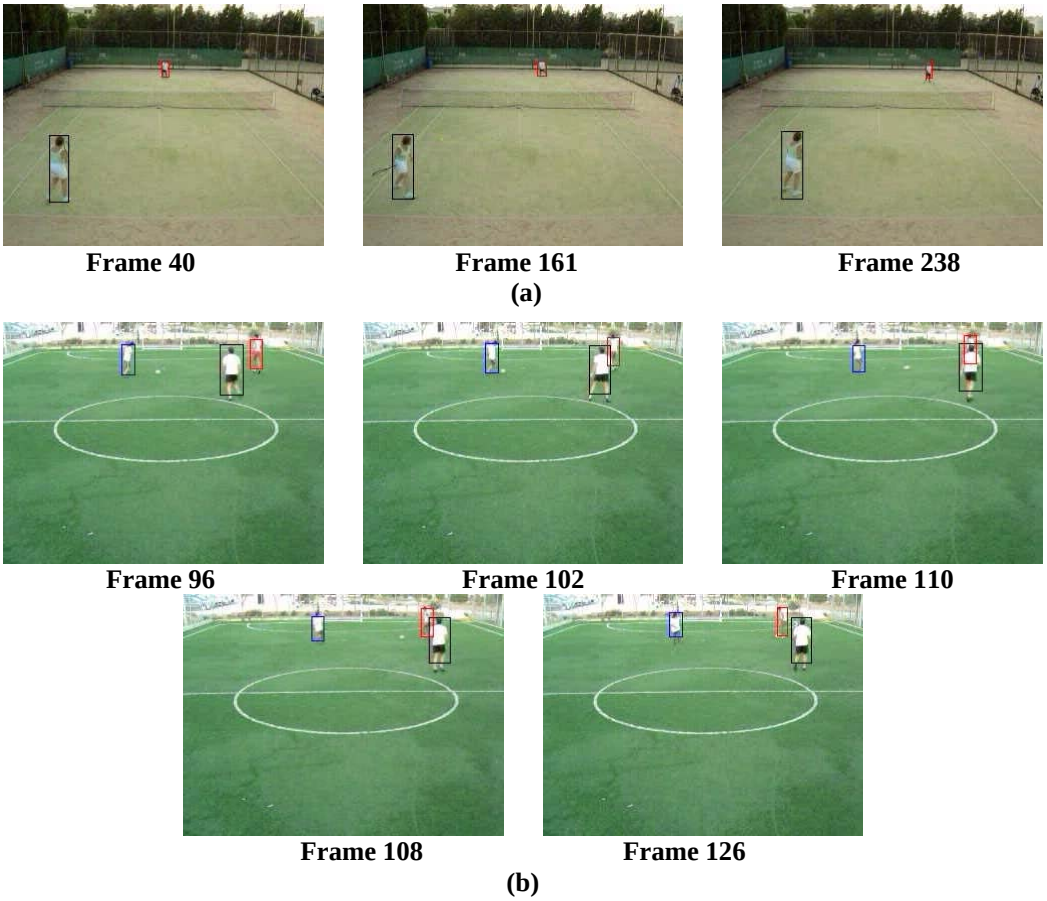


Figure 4.1: Tracking results across frames, using the proposed method on our data set.

4.5 Summary and Conclusions

This chapter has presented a novel framework for multi-object tracking. The aims have been succeeded and therefore the correspondence between objects across frames is established. The achievement of this project is that two effective methods have been combined, using the advantages of each method, in order to decrease the probability of error detection. A method that uses a cost function rewards constantly good scores, because it does not predict the position of the blob but actually calculates and correlates the real points. On the other hand, it suffers by occlusions from others regions, which it is a frequently observed phenomenon in sport matches. This can be faced by the prediction of the position of each occluded blob using an method based on the condensation algorithm. A set of Kalman filters are introduced to regroup the stochastically propagated samples, within the condensation framework, targeting to improve the tracker performance. These results are considered to be better and more accurate than in any other proposed method, because this method does not estimate all possible positions but only the cases where an occlusion appears.

All the necessary information is exported in order to transform the real image into a virtual ground image from the upper view, and display the players as an animation for web or wireless applications. The only step left in order to accomplish our target is the field region extraction and the projective transformation of the coordinates of each player, that is a process fully analysed in the next chapter.

Chapter 5
Field Region Extraction and Projective Transformation

5.1 Introduction

5.1.1 Approach

The aim of this dissertation is to automatically track sports players, using a single static camera, and identify their real world positions, in order to describe the contents of an athletic event. In order to reach this goal, we need to associate the coordinates of the real image's moving silhouette with those of the virtual ground image.

To illustrate the difficulty regarding the projective transformation between a real image and a ground platform from the upper view, one needs to observe the relationship between height and depth of the pitch. During a match, all the action of the game is constrained to the pitch; however, the perspective of the image highlights several issues:

- The size of the pitch depends on the sport (e.g. in tennis courts is *11x24* meters).
- The image's resolution varies greatly between the nearest and the furthest parts of the pitch. Specifically, in an image with resolution *320x240* pixels, the covered interval of two vertically adjacent pixels in the nearest part of the pitch are just about 3.4 cm apart. On the other hand, those pixels in the far goalmouth of the pitch are almost 9 cm apart.

5.1.2 Related Work

The transformation from the set of coordinates of an image plane to another set of a ground plane has been an area of interest in many studies. These studies mainly used fundamental matrices for transformations and had good performance but high complexity. In [7], [14] fundamental matrix in geometry was used to track the player. Computed fundamental matrices naturally contained some error, and produced negative effects on the intersection calculation of the lines which was needed to estimate the player-location. Additionally, only a single player could be tracked in studies which used fundamental matrices. Image projective using fundamental matrix is also used for wide-screen display. In [35], the authors used planar homography in projective geometry. Homography gives one-to-one relationship of the points between two images while fundamental matrix gives the relationship between the point of image and the line of the other image. In that way, the error

in estimating the players' location became smaller. A disadvantage of using the homography is the need to use more than one cameras. Methods using matrices as well as depth coverage of geometry for computer vision can be found in [30]. Indeed, using a method that transforms only the coordinates of the positions of the players, and not all the coordinates of the field, seems to be a more efficient solution. An algorithm based on bisection is proposed and implemented in this dissertation.

5.2 Field Estimation

The extract of the field region is basic requirement for the tracking process necessary for understanding a game. The regions that are not included in the terrain and the super-imposed captions should be excluded from the processing image. Only the objects inside the extracted field region should be considered in the procedure.

Looking at various football fields, we were able to observe distinct characteristics of the terrain. All the fields are flat, and only athletes and referees are on the pitches. The colour of the pitch varies depending on the sport that it is used for. Football and rugby fields as well as some tennis courts (like Wimbledon) are green-coloured. In other tennis courts like Roland Garros the fields are usually red/brown, whereas in indoor terrains like basketball, volleyball and handball the colours of the pitches vary (grey, brown etc). The method proposed in this project can be implemented successfully in any field regardless its colour.

After extracting the background of video sequences, we will implement the following procedure in order to find the outline of the pitch. This will help us to detect six pairs of corresponding points on the background image (four pairs corresponding to the four corners and two pairs of coordinating points correlated to the middle of the field), which are essential for the transformation.

In sports having a grass arena, the field region is roughly coloured green. Thus, firstly the green regions in the image were detected. According to the RGB colour-map a certain range were defined to detect green coloured pixels. The variable R, G, and B takes values in the range *0-255* and so, grass includes the following colours:

$$\begin{aligned} &Red \rightarrow 10 \ (\pm 10) \\ &Green \rightarrow 160 \ (\pm 50) \\ &Blue \rightarrow 10 \ (\pm 10) \end{aligned} \tag{5.1}$$

Additionally, Red and Blue values cannot exceed a threshold equal to 120 respectively, because in a different case the combination of these colours is not Green. Therefore, every pixel of the image is checked, and

- if the value of all colours are not in the proposed interval,
- AND, if the value of any of the Red, Blue or both of them exceed the proposed threshold (120)

→ then, this specific pixel does not belong to the playfield.

A problem that may occur using this method is the presence of spurious pixels and holes in the regions, because different objects in the background have the same colour. A continuous region with the largest area is considered as the field region, since there are occasionally small green coloured regions outside the field. Extra-field regions such as the gallery should be excluded. This problem can be circumvented using a morphological filter (closing and opening).

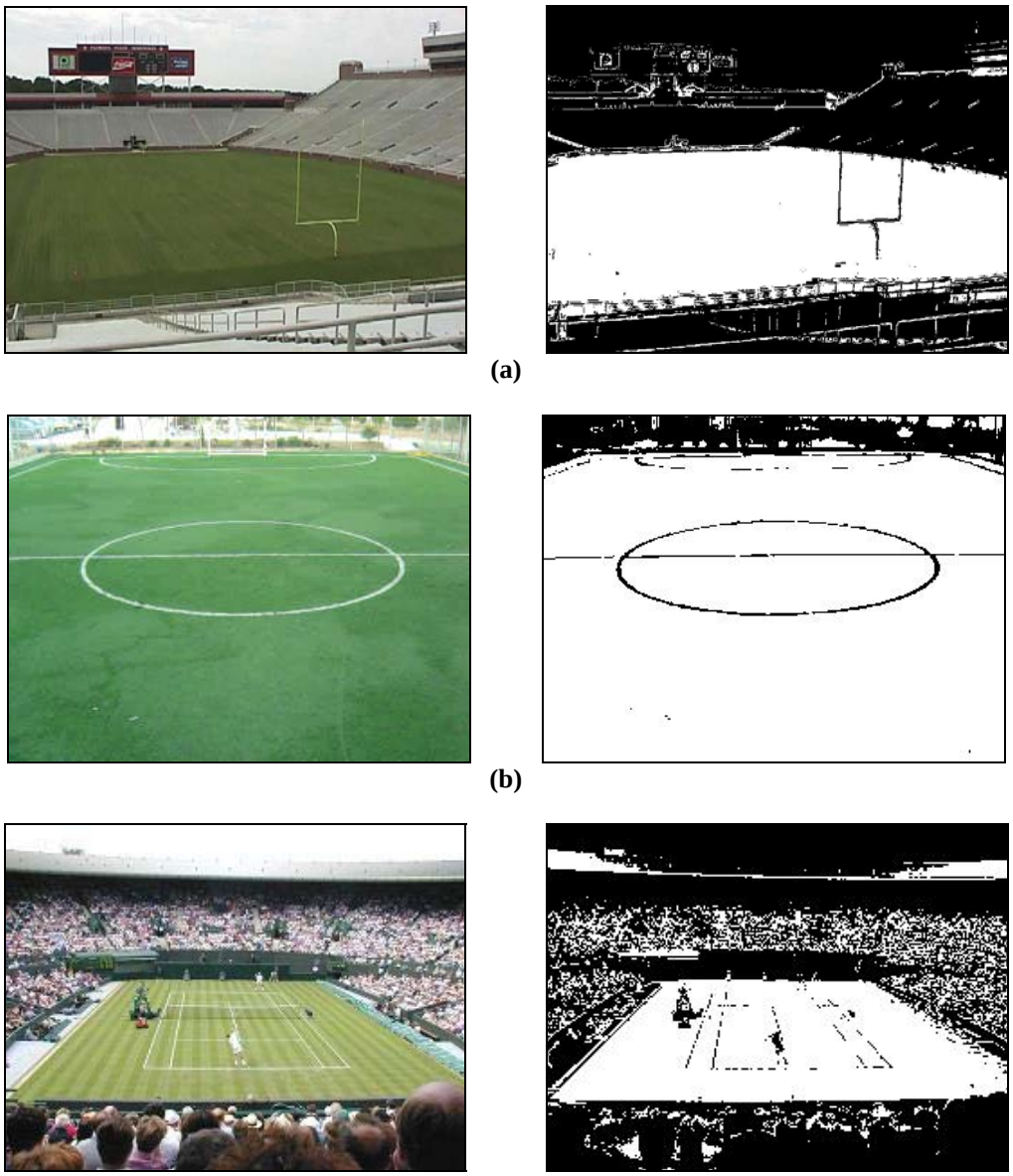
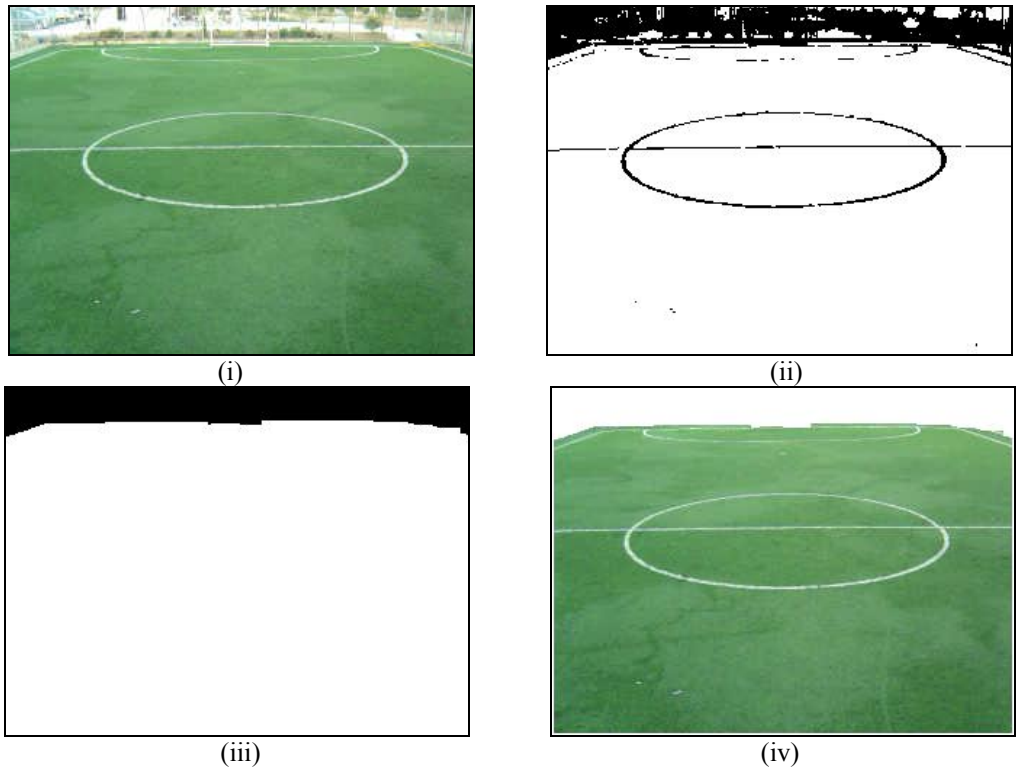


Figure 5.1: Field Estimation before image enhancement by morphological filter. **(a)** American Football, USA, **(b)** Mini football Fitness-Land Campus, Limassol, Cyprus, **(c)** Tennis, Wimbledon, UK

Looking at the examples of the detection and the extraction of the field in images 5.1 and 5.2, the effectiveness of the proposed scheme can be observed. Using this, the important points of the field, needed for the projective transformation were detected, and the blobs that are not included in the area of the field were removed, since these are not players participating in the game (all the players are inside the field, see figure 5.2(b)).

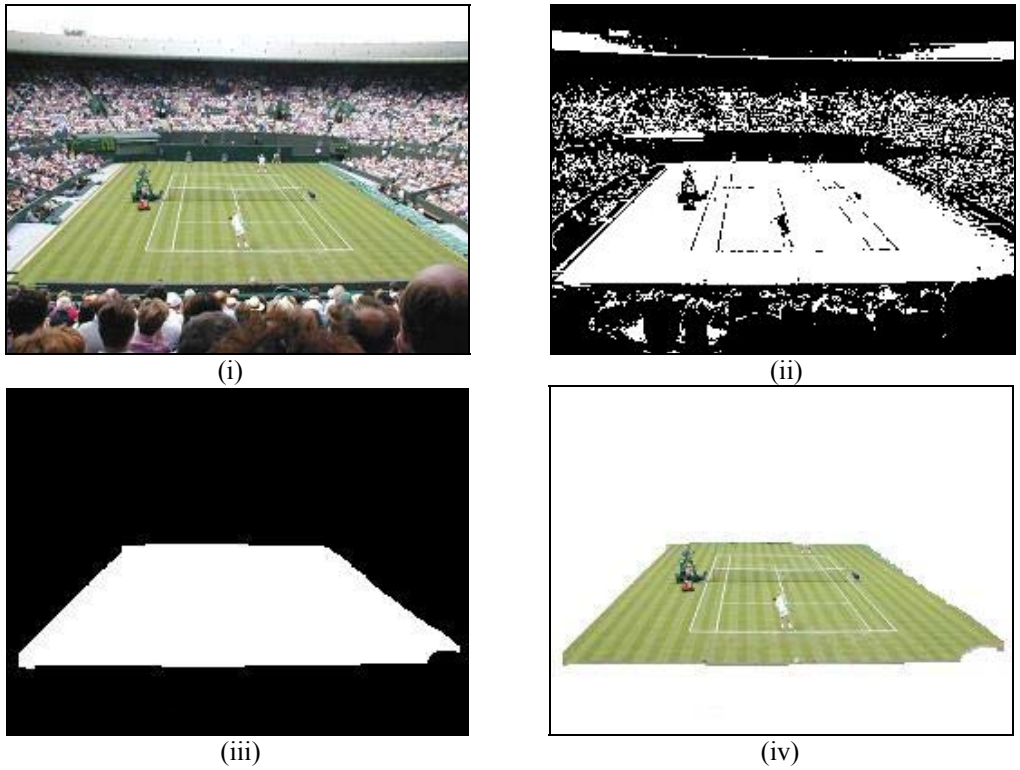
The following figures present some results of the above method:



(i) (ii)

(iii) (iv)

(a)

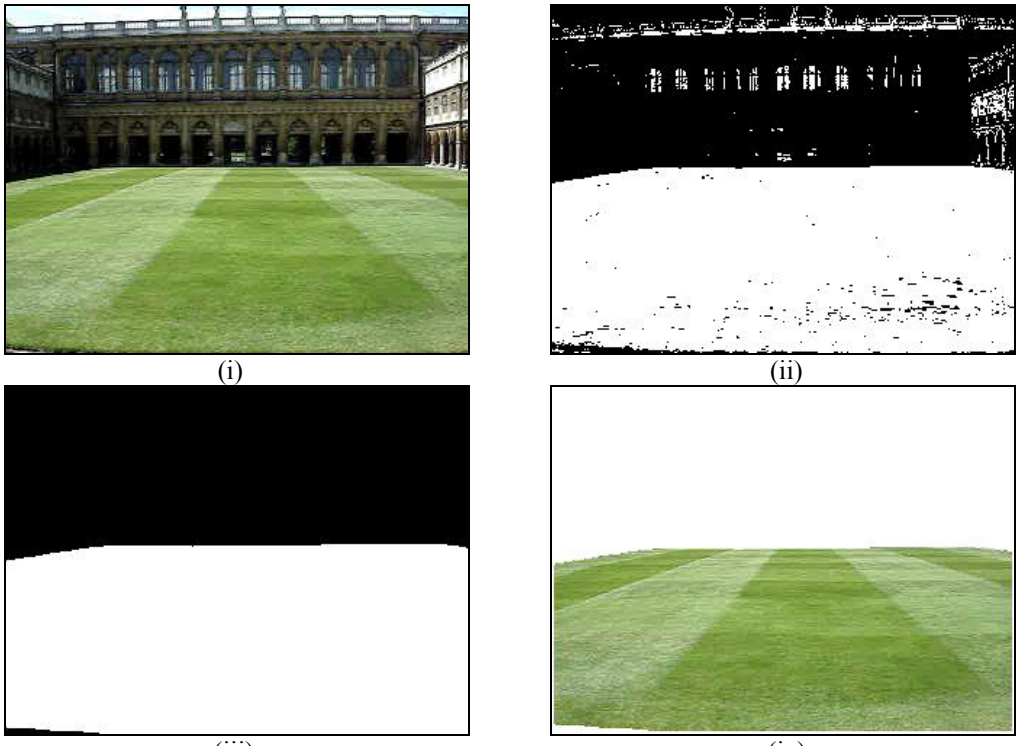


(i) (ii)

(iii) (iv)

(b)

Figure 5.2: Field Extraction: (i) input image, (ii) image before enhancement by morphological filter, (iii) extra field region exclusion, image after enhancement by morphological filter, (iv) Field region Extraction. **(a)** Mini football Fitness-Land Campus, Limassol, Cyprus, **(b)** Tennis, Wimbledon, UK.



(i) (ii)

(iii) (iv)

(c)

Figure 5.2 (continue): (c) Trinity College, Cambridge, UK

5.3 Projective Transformation

In this section, a straightforward image plane to ground plane transformation *(2D to 2D)* is used to determine where the players are on the ground plane from where they appear on the image. Six pairs of corresponding points on the background image (four pairs corresponding to the four corners of the playfield and two pairs of coordinating points correlated to the middle of the field), and known real world ground plane dimensions are used to construct transformation algorithm. A more sophisticated approach of camera calibration in order to calibrate the image, which allows transformations between *2D* images, coordinates and *3D* world coordinates, can be explained analytically in [31]. In the case of 3D transformation, which can be accomplished using two or more cameras, we are able to locate the exact position of the ball as well as the exact positions of the players especially when they jump. More cameras will improve the overall performance of the system but will also make the price higher as well as demanding more computational power.

To be able to transform from one set of coordinates to the other, the distance of the field W_x and W_y as well as the six pairs of corresponding points on the background image, are needed. The position of the points related to the near-left, near-right, half-left, half-right, far-left, and far-right can be calculated during the field region estimating process.

The true dimensions of the mini-football field and tennis court are presented in Figures 5.3 and 5.4 respectively:

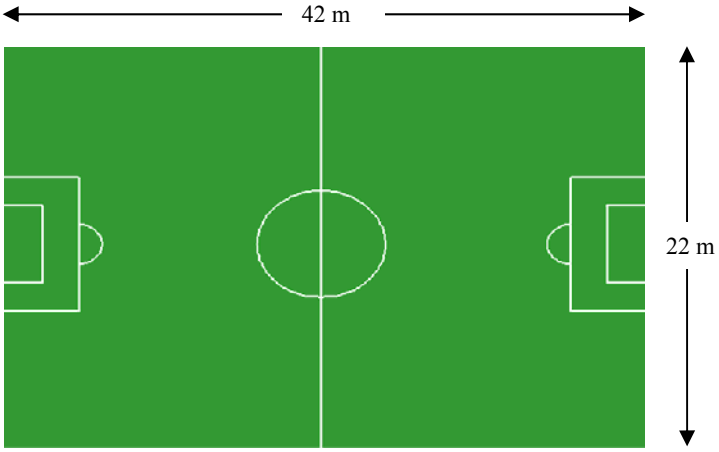


Figure 5.3: Real dimensions of a mini football field.

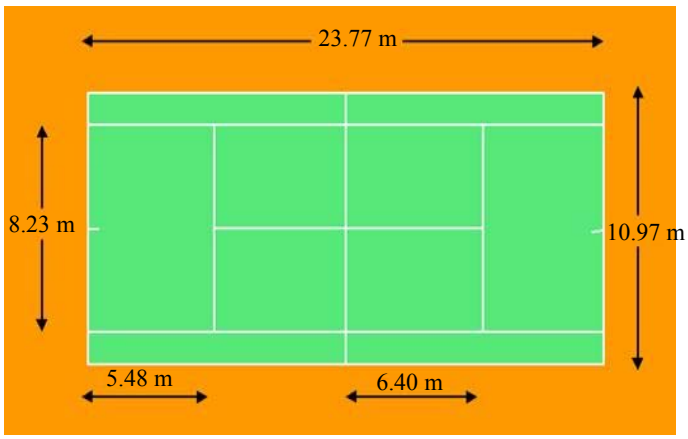


Figure 5.4: Real dimensions of a tennis court.

The purpose is to transform the coordinates of the players in the field, as they are shown in the real image, to a platform that is the ground plan of the terrain. Figure 5.6 presents this idea as well as the relationship between depth and size both on the X and Y-axes.

For the transformation to take place, the camera must be in the middle of the sideline of the pitch. The system environment is shown in figure 5.5. The height and the distance of the camera from the field do not have pre-defined values but they need to have such values in order to include the whole area we want to transform. The height and distance from the pitch, affect the performance of the transformation because the higher and closer we are to the field, the better the transformation will be.

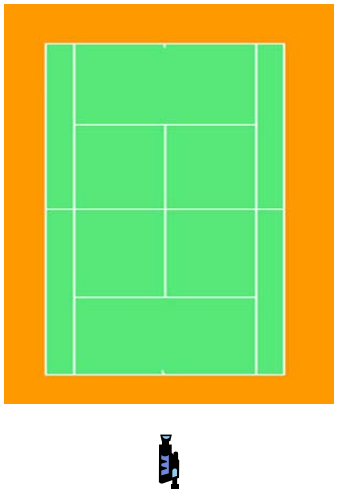


Figure 5.5: The location of the camera

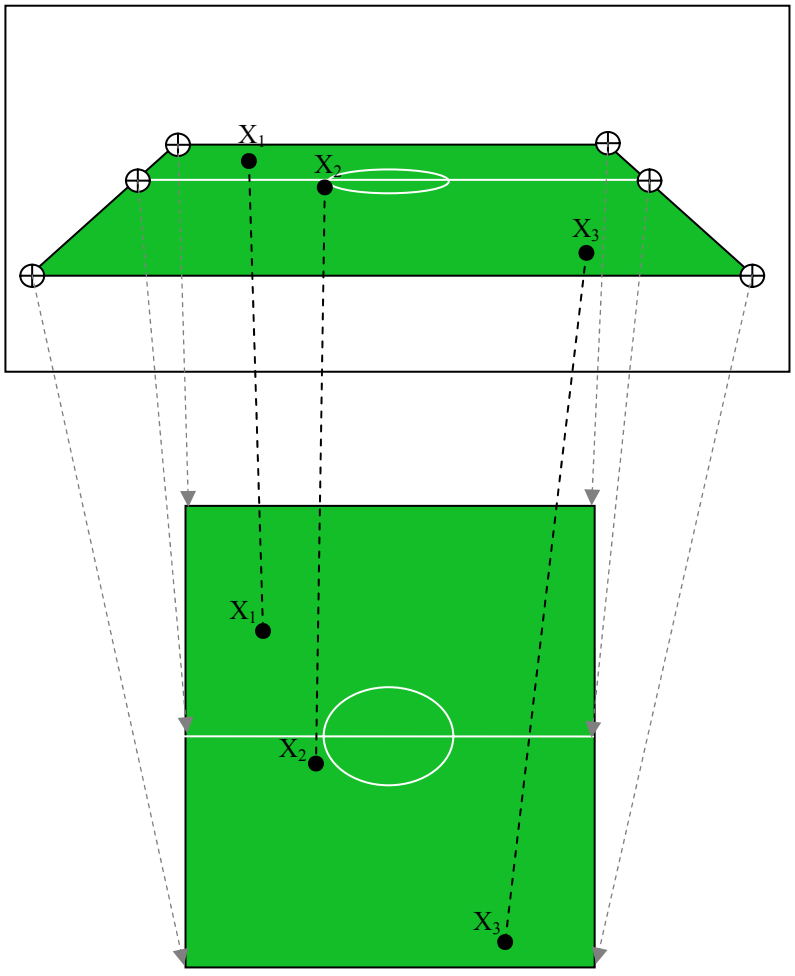


Figure 5.6: The projective transformation.

The desired image is taken as reference. Thereafter, the coordinates of the moving objects that are included in the playing terrain domain are extracted. Then these coordinates are transformed to the new platform space. This transformation is analysed later on.

For every moving object, the following information holds:

$$X = \left(id, x_g, y_g, Area \right) \tag{5.2}$$

where *id* is the identification number, x_g, y_g are the ground plane coordinates, and *Area* is a value associated with the blob's area.

As already mentioned, sports players are always on the pitch, and often (particularly in sports like netball) have a tactical position in which they stand for short periods of time. Hence, the x_g, y_g coordinates correlated with player location, are present inside the field. The projective transformation can be implemented as follows:

Y coordinates Transformation

Y_{near}, Y_{half} and Y_{far} are known (from chapter 4) and are illustrated in figure 5.7. The input is the ground plane coordinates, Y_g, of the moving object. The algorithm based on the Bisection method, is presented below in pseudocode:

```
// Y_far-Left = Y_far-Right
// Yn_ear-Left = Y_near-Right
// Y_half-Left = Y_half-Right

Range = (Y_half - Y_far) / (Y_near - Y_far);

y = Pwy;        // the Pwy length of the platform.
Y_output = Pwy;

// Updated Coordinates.
While y > 1  // loop.
    Y = y / 2;
    if Y_g > Y_half    // If Y_g greater than half, then Y_output remains stable.
                       // Assign Y_half to Y_far and calculate the new value of Y_half.
        Y_far = Y_half;
        Y_half = Y_far + ((Y_near - Y_far) * Range);
    Else               // Otherwise, η Y_output takes the new value of Y_output=(Y - y).
                       // Assign Y_near to Y_far and calculate the new value of Y_half.
        Y_output = round(Y - y);
        Y_near = Y_half;
        Y_half = Y_far + ((Y_near - Y_far) * Range);
    end
end
```

The above code returns Y_{output}, which is relative to the Y coordinate of the moving object in the virtual field.

To perform the transformation of the X coordinates, we first need to calculate the coordinates of the side positions. The closer the X coordinates get to the centre, the smaller the shifts. As the X coordinates move further away from the centre the move becomes larger. As we move downwards in the image and Y_g is large, the transformation will be less effective. Inversely, as Y_g gets smaller, the transformation becomes more effective. Therefore, the height of the blob has a great impact on the transformation of X because the positions of the bounding coordinates depend on it. Figure 5.8 presents this.

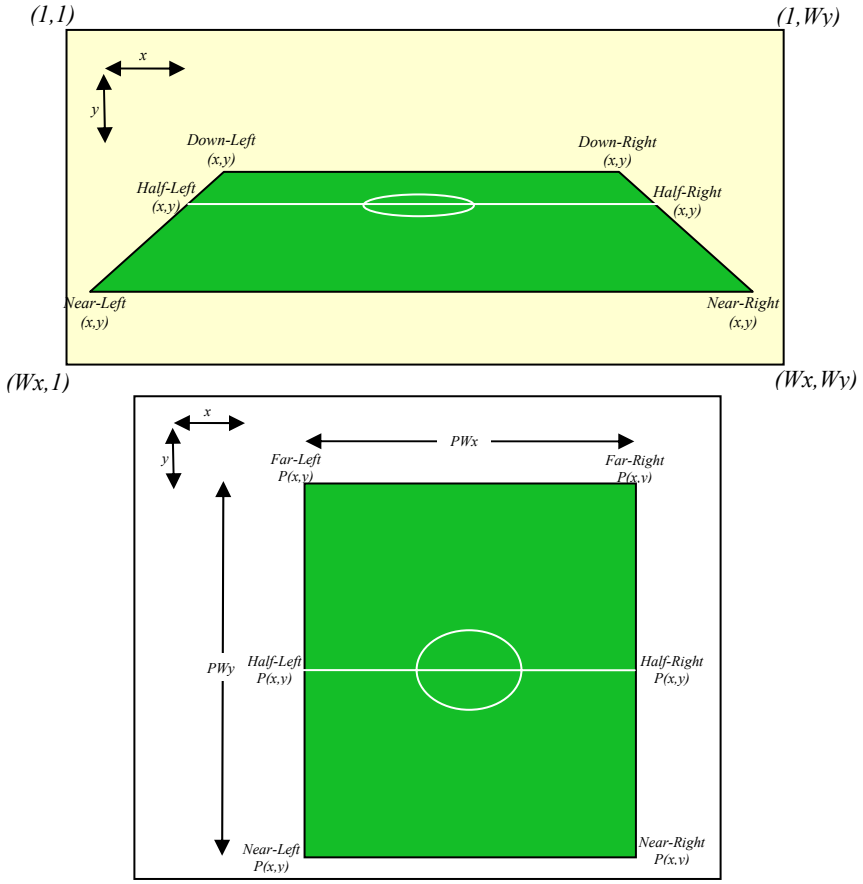


Figure 5.7: The projective transformation and all the necessary information for the implementation.

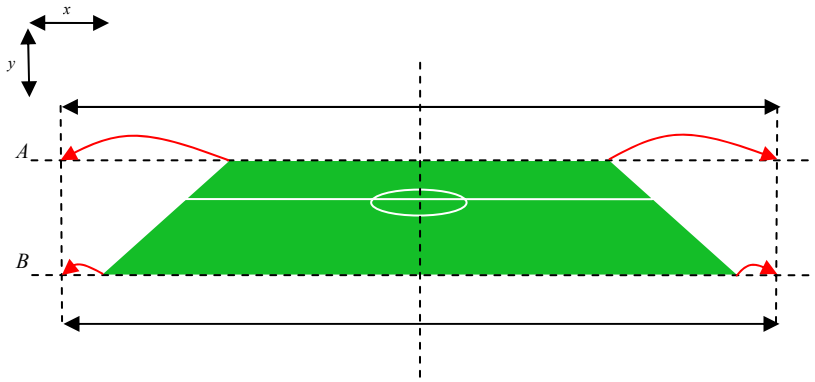


Figure 5.8: Bounding point coordinate's translation related to the distance from the "*far*" points of the terrain (The range between the point A and B). We can easily observe that the transformation of the *x* coordinates in the case when we are at the far end of the pitch (point A) is more effective, in contrast to the case where the location of the blob is near the beginning of the pitch (point B).

Additional information is needed to be able to calculate the X coordinate, X_{output}, which corresponds to the blob in the virtual field. Firstly, the leftmost point $X_{marginal\text{-}left}$, and the rightmost point $X_{marginal\text{-}right}$ of the real field are needed. This is calculated using the far left and right ($X_{far\text{-}left}$, $X_{far\text{-}right}$), as well as the location Y_g, where the corresponding blob with X_g is. The following pseudocode presents the calculation of the leftmost and rightmost points, respectively.

Estimation of the Bounding Points

```
N = (Y_near - Y_far) / 2;     // The centre value.
Mean_Y = Y_near - N;          // Initialization of the Y-coordinate centre.
Diff = ((W_x - X_near-right) + X_near-left)/2;  // A variable related to the distance value.

X_marginal-left = X_near-left      // Initialization.
X_marginal-right = X_right-left    // Initialization.

While diff > 1
    diff = diff / 2;
    if Y_g <= Mean_Y   // If Y_g less or equal to the centre, then the bounding points
                       // remains stable. Assign N to N/2 and calculate the new
                       // value of Mean_y.
        N = N / 2;
        Mean_Y = Mean_Y - N;
    else               // Otherwise, N is equal to N/2, the new value of Mean_y
                       // and the new bounding points are calculated.
        N = N / 2;
        Mean_Y = Mean_Y - N;
        X_marginal-left = X_marginal-left - diff;
        X_marginal-right = X_marginal-right + diff;
    end
end
```

Now, the X coordinate, X_{output}, can be calculated, since we know the coordinates of the marginal points, left and right. This calculation is presented in pseudocode below:

X coordinates Transformation

```
Mean = (X_marginal-left + X_marginal-right)/2;   // Initialization of the X-coordinate centre.

// Check if it is on the left-half or at the right-half bound of the field.

If X_g < Mean              // If it's at the left-half bound, X_output must be
                           // transformed moving it to the left.

    X_low = X_marginal-left;   // Assign X_low as X_marginal-left and X_high as Mean. The new value
                               //of X_half is calculated, and X_out is set to 0.
    X_high = Mean;
    X_half = (X_high + X_low)/2;
    x = (X_high - 1);
    X_out = 0;

    while x > 1  // loop.
        X = x / 2;
        if X_g >= X_half    // If X_g is greater or equal than X_half, the new value of X_out
                            // is calculated, X_low is defined as X_half, and a new value
                            // for X_half is calculated.
            X_out = round(X_out + x);
            X_low = X_half;
            X_half = (X_high + X_low)/2;
        else                // Otherwise, X_out remains stable, X_high is defined as X_half
                            // and a new value for X_half is calculated
            X_high = X_half;
            X_half = (X_high + X_low)/2;
        end
    end

else                        // If it's at the right-half bound, X_output must be
                            // transformed moving it to the right.
    X_low = Mean;           // Assign X_low as Mean and X_high as X_marginal-right. The new value
                            // of X_half is calculated, and X_out is set to Mean.
    X_high = X_marginal-right;
    X_half = (X_high + X_low)/2;
    x = W_x - X_low;
    X_out = Mean;

    while x > 1  // loop
        x = x / 2;
        if X_g >= X_half    // If X_g is greater or equal than X_half, the new value of X_out
                            // is calculated, X_low is defined as X_half, and a new value
                            // for X_half is calculated.
            X_out = round(X_out + x);
            X_low = X_half;
            X_half = (X_high + X_low)/2;
        else              // Otherwise, X_out remains stable, X_high is defined as X_half
                          // and a new value for X_half is calculated
            X_high = X_half;
            X_half = (X_high + X_low)/2;
        end
    end
end

X_output = round((X_out * PW_x) / W_x);  // X_output is defined as the X_out multiplied with the
                                         // length of the pitch of new image divided by the
                                         // length of pitch of the real image.
If X_output == 0
    X_output = 1;
End
```

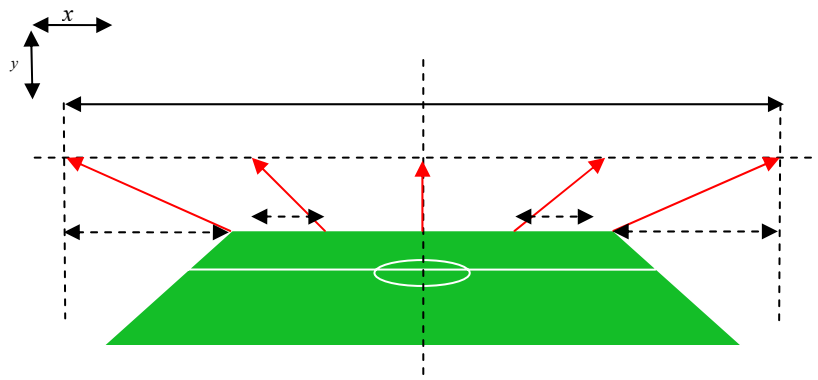


Figure 5.9: *x* coordinates' translation related with the distance from the terrain centre. It's obvious that the closer the X coordinates get to the centre, the smaller the shifts. As the X coordinates move further away from the centre the move becomes larger.

The following tables present an implemented example of the mapping from the real image to the platform, for a sequence of 3 frames with 3 blobs each.

Table 5.1: A tracking example presenting the relationship between *id* number and the blob's location (ground plane coordinates) correlated to the real image, for frames 56, 57, 58. Blob's *Area* is also included *(image resolution 320x240)*.

	Id	x_g	y_g	Area
frame 56:	3	133	58	7
	2	262	57	7
	1	213	47	5
frame 57:	2	260	60	7
	1	211	56	6
	3	131	52	6
frame 58:	2	260	60	7
	3	131	63	7
	1	211	56	6

Table 5.2: A tracking example presenting the relationship between *id* number and the blob's location (ground plane coordinates) correlated to the ground platform, for frames 56, 57, 58. Blob's *Area* is also included *(image resolution 525x600)*.

	Id	x_{output}	y_{output}	Area
frame 56:	3	208	289	7
	2	461	279	7
	1	365	271	6
frame 57:	2	459	281	7
	1	363	271	7
	3	206	290	6
frame 58:	2	459	281	7
	3	207	291	7
	1	363	271	6

Note: *New Area value corresponds in the root square divided by π of the previous Area value:* $Area = \frac{\sqrt{Area}}{\pi}$

It is easy to see that the transformation is very sensitive, especially in the cases when we are at the far end of the pitch of the real image. As expected, in a typical image, if two vertically adjacent pixels on the image plane are projected onto the ground plane, then pixels in the nearest part of the image are closer than those in the far part of the image. In this case, the pixels of the nearest part are just about 3.4 cm apart, whereas those in the far goalmouth are almost 9 cm apart (Figure 5.10). In the area of the image representing the nearest part of the pitch, 5 metres of ground plane covers 145 pixels, compared with only 28 pixels at the far end of the pitch (Image resolution *320x240*).

Figure 5.10: A typical image (Image resolution is 320x240 pixels). It is clear that, in the area of the image representing the nearest part of the pitch, a player covers more pixels, compared to the far end of the pitch.

5.4 Summary and Conclusions

This chapter showed the detection and extraction of the playfield in order to perform a space transformation from the real image to a platform that shows the ground floor of the field. This is also applicable when we want to remove all the blobs that do not belong in the field region, since they are not players of the particular game.

Observing the field, we managed to extract some useful information regarding the colours comprising the surface of the pitches and using these characteristics, we were able to extract the playfield region. The extraction of the playfield was successful, and the important points surrounding the pitch were identified. Using the information about the pitch, like the location of the four corners, the left and right points of the centre of the pitch, the real positions of the athletes in the real image were transformed to a new space that corresponds to the ground image from the upper-view. Overall, the suggested method provided excellent results, and required low computational power since it was not necessary to transform the whole pitch but only the coordinates corresponding to the athletes.

The extraction of the playfield terrain and the projective transformation are the last stages of this project. Now, we have all the necessary information to accomplish the goals set, which are tracking multiple players and identifying their real world positions, in order to describe the contents of an athletic event. In Chapter 6 we will evaluate the results and discuss different applications.

Chapter 6
Evaluation of the Results and further Applications

6.1 Implementation of the proposed method

In this study, a combination of individual processes has been applied, each one constituting a step in the attempt for real time tracking and players' localisation. Extensive experiments of the proposed system, evaluated in an outdoor 5 a side football match and a tennis game, verified the effectiveness of this model. In this chapter, the results will be evaluated with the aim of identifying suggested applications. The technical characteristics of the implemented system are also reported.

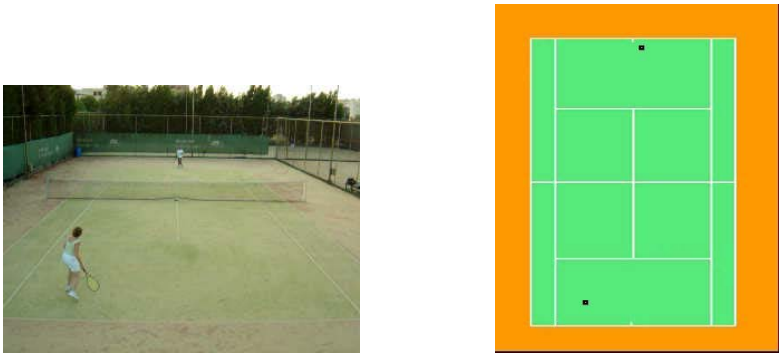
6.1.1 Experimental Results

The experimental results and one possible application are presented in figure 6.1. The location of the x-y coordinates of the player's feet, found inside the playfield, have been converted in a new set of coordinates of a ground platform. In the example below, the tracker can successfully correlate the players' positions across frames. For instance, in frame 140 and 290, figure 6.1, the players' positions with the related colours in the ground plane platform correspond to the silhouettes in real image. The use of better graphics and different colours for the players of each team could attribute better aesthetic results, something that is not part of this work.

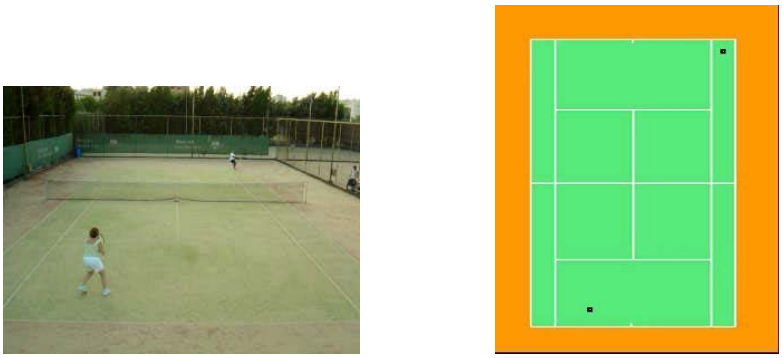
In any case, the *id* of each blob is known, as it appears in tables 5.1 and 5.2, with the corresponding ground plane coordinates and the covered area (*Area*). This gives us the opportunity to present the trajectory of each player that is stored in a trajectory map. Hence, a second implementation could be the one in figure 6.2, that is, the analysis of motions, fitness and tactics of teams and players. In this example, the motion of each player inside the playground can be observed and therefore, important information about the location and movement tactics during the match might be attained.

Frame 140

Frame 290



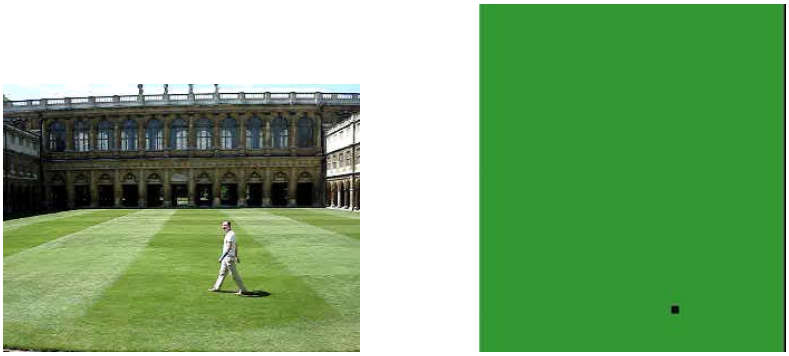
Frame 50



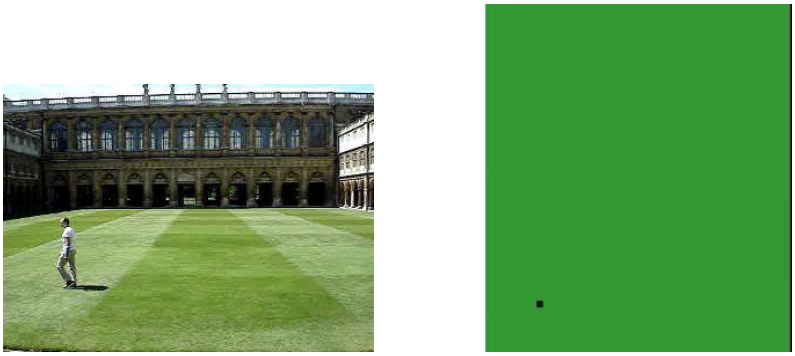
Frame 125

(a) **(b)**

Figure 6.1: Examples of implementation: **(a)** The real image, **(b)** The projected platform in Java Applet.



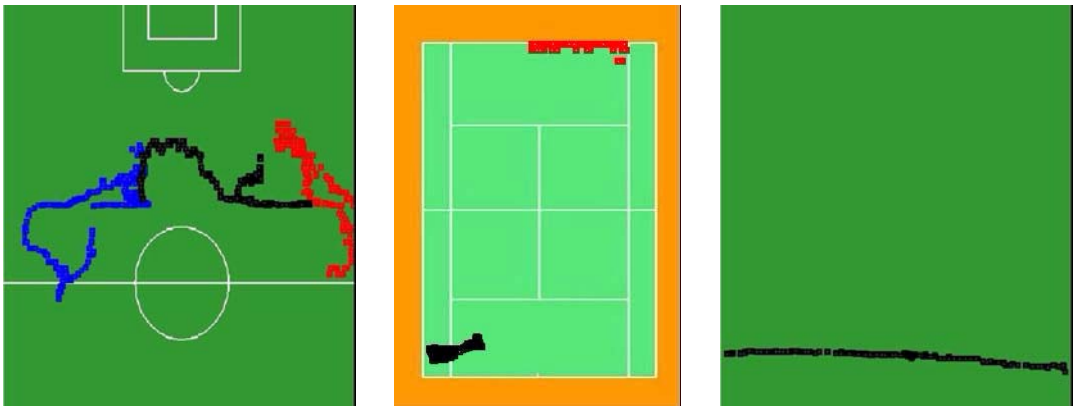
Frame 82



Frame 245

(a) **(b)**

Figure 6.1 (continue): Examples of implementation: **(a)** The real image, **(b)** The projected platform in Java Applet.



(a) **(b)** **(c)**

Figure 6.2: Trajectory of the moving objects

6.1.2 Results Discussion

The method outlined in this paper can be successfully demonstrated in several recorded matches, such as football, tennis, rugby. The designed predictions have been materialised, the players' positions have been located and converted in a new space that constitutes the ground plane of the playfield. This gave us the opportunity to develop a playing low-bandwidth match animator for web or wireless display, in mobile devices, which was the primary aim of this study. Each user has the opportunity to watch a match that interests him using his mobile phone at any time and place. The system works as planned and gives reasonably reliable and accurate results.

To provide a more efficient approach to the multi-object tracking sport players system for wireless display, several problems have to be encountered. Even if the system has high attribution of operation, such problems critically affect the performance of the system decreasing its validity. The main drawbacks of this work as well as possible ways to deal with are reported below:

- The first basic problem is focused on the accurate estimation and calculation of the real location of each silhouette. This involves jitter in the extraction of the y-coordinate of each athlete. There are many factors deteriorating this problem, starting from the erroneous estimation of the players' feet (ground plane coordinates). As already mentioned, feet's of the players is the position used as reference of the player's location in the playfield. Thus, when the athletes jump above or run, it is obvious that the down part of their legs does not correspond to their real place in the terrain. In addition, since some players are placed in the far end of the field, their legs seem to be very thinly. This affects the accuracy of the estimation of the players' location, because after the image improvement using the morphological filter these regions are erased. This drawback can be improved using more cameras, located in concrete places, returning better results but with the disadvantage of increasing the complexity and the computational cost. A more efficient foreground estimation procedure can also improve the performance of this process. Another way facing such problems could be achieved using higher resolution cameras. This happens because a pixel in the far end of the low-resolution image corresponds in bigger interval than in the case of high-resolution image. The higher the image resolution, the lower is the animated player's jitter.
- The second existing problem is the localisation of the position of the ball. The ball is not difficult to be exported, but the localisation is impossible to be estimated using a single camera. This happens because the ball moves in 3 dimensions, x-axes, y-axes and height. Essential condition for the localisation of the ball position is the use of more than 2 cameras, with the previously mentioned advantages and disadvantages.
- Problems in the validity of the tracker constitute a third drawback for this study. Match situations involving many tightly packed players give inaccurate estimates, and re-initialisation errors as the players re-disperse. Currently, if two players enter the field of view as an occluded group, the single-view tracker cannot recognise the correct number of players. This drawback can be faced using a multi-view tracker from several cameras.
- Another equally important problem of this work is detected on the field extraction. Even if a few pixels more or less than the required to define the field are utilized, the coordinates of the players on the platform will be declined from the corresponding coordinates in the image after the mapping.

6.1.3 Technical Characteristics

A video database has been created for the validation of our software implementation. The database mainly constitutes by video sequences on different outdoor environments, such as football fields and tennis courts.

All experiments were run on *320 x 240* pixel resolution images using MathWorks Release 14 including MATLAB 7.2 and Simulink 6.2, on a computer with a Pentium Centrino 2.0 GHz. The operating system used is Microsoft Windows XP service pack 2.0. For the video processing, AVD Video Processor 7.4 has been used, in order to create AVI files compatible with MATLAB. All videos were taken using a Sony^{TM} Digital Camera, DSC Cyber-shot 200.

All videos include 25 frames per second, in RGB space and without any video compression, in order to be compatible with MATLAB functions *avifile, aviread* etc. Inputs are the image sequences of sport games in the single-view points.

6.2 Further Applications

The main application of this study is for real time low-bandwidth JAVA animations. This could be used for displaying match animations for web or wireless display, such as football and tennis, showing the score display at the same time. Several clubs, athletes or organisations, as tennis grand slams of Wimbledon, Roland Garros, American and Australian Opens might display their games in real time via their official websites, gaining thus money from various advertise that will be included in their web page. In addition, because in many sports the TV presentation rights have been prepaid, display on demand is another one possible solution. Such an application does not exist in the market and gives the opportunity to people who are interested to watch sport events using theirs mobile appliance at any moment and place.

Another application is the analysis of fitness and tactics of players or teams during the match. Player's motion and position analysis could be a very useful tool in the hands of coaches in order to point out possible mistakes and omissions of their athletes. This application could be demanded in clubs of the F.A. Premiership and in commentators or analysts of matches in big television networks as the BBC.

Beside the athletic uses, another application is the surveillance in closed circuit security systems. A system having this operation, in combination with systems that can recognize movement could be an effective method to prevent terrorist or criminal energies.

Chapter 7
Concluding Remarks

7.1 Towards a model of tracking sports players for wireless display.

In this dissertation, an architecture system, that aims to track people for sports applications using a single static camera, has been presented, in order to identify their real world positions. In this way, the moving objects of the playing field can be detected, and the players' positions will appear in a virtual ground plane image as a function of time.

In Chapter 2, a real-time background modelling and maintenance, based on the segmentation of input images into stationary and non-stationary blocks, has been described. The background image is synthesised using the median value of the stationary regions. Therefore, no bias towards the foreground colour will occur in the reconstructed background. A similarity matrix has been created, which contains the difference between the image content at the block position for each pair of frames. High values correspond to non-stationary elements and low values to stationary elements. Background periods are obtained by searching for the subset of frames so that the sum of stationary matrix elements is minimised, while the sum of non-stationary matrix elements is maximised. This method is efficient even if many objects are simultaneously visible and the background can be seen only for a short time.

The main purpose of background estimation is to separate the background from foreground regions of motion and as a result to identify moving objects. This can be achieved using a method based on background subtraction that is developed in chapter 3. Chromaticity and lightness change are used as the differencing method between current image and a reference background image in a pixel-by-pixel fashion, so that the moving silhouettes can be detected. Thereafter, a pre-processing step is applied, in order to avoid erroneous pixels and anomalies in the detected areas, using a morphological filter. Blob's information such as the centroid's coordinates, the covered areas, the bounding boxes and the ground plane coordinates have been calculated for every moving object.

A novel multi-object tracking method has been proposed in chapter 4 in order to establish correlation between the moving silhouettes across frames. This method is implemented using a cost function of blob information such as the centroid's coordinates, the covered area, the velocity and the colour, in addition to a condensation filter. A set of Kalman filters improve the performance of the tracker, since they help regrouping the samples that are stochastically propagated within the condensation framework.

The development of a field estimator and a projective transformation have been presented in chapter 5. Colour information of the ground is used for field region extraction so that the required information for the mapping can be perceived. Objects that exist outside the field are excluded because they are not participating in the sport event. Nevertheless, a new robust

projective transformation is applied, in order to map the player's coordinates from real image to a new set of coordinates regarded the virtual ground image from the upper-view.

Finally, chapter 6 provided the results of the proposed architecture in addition to some recommended applications. For example, web applications such as a Java Applet displaying the players' locations and trajectories, were presented as an alternative way of playing low-bandwidth match animations for web or wireless display in mobile devices. The structure of this study can be more clearly understood in figure 7.1.

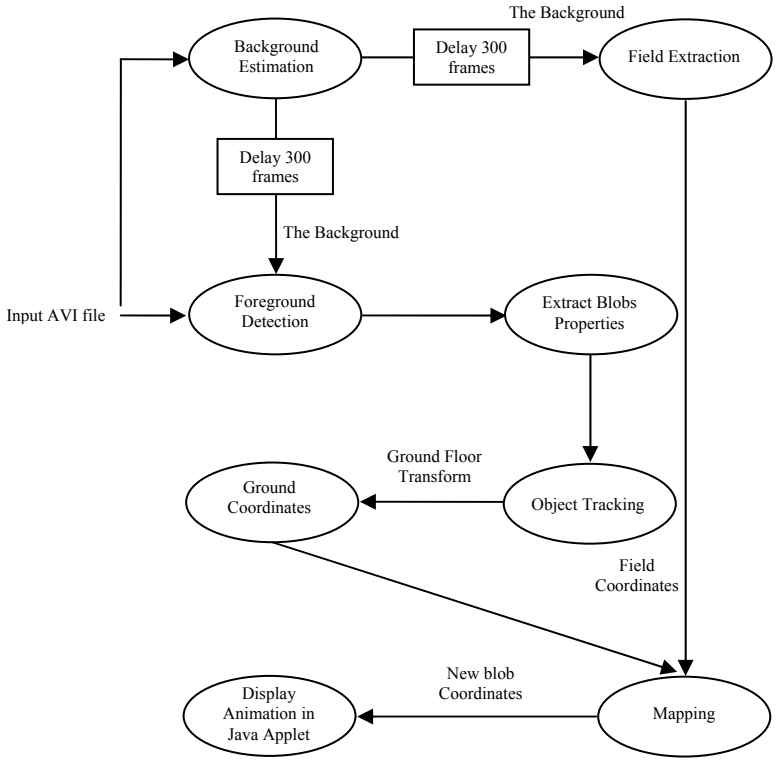


Figure 7.1: Towards a model of tracking sports players for wireless display.

7.2 Contributions

The main contributions of this dissertation are:

- An extension to the tracking algorithm that uses a cost function of blob information, such as the centroid's coordinates, the covered area, the velocity and the colour, in combination with a condensation filter, in order to cope with complex cases such as when regions enter/exit the scene and when they are occluded by other regions. These results are considered better and more accurate than in any other proposed method, because the player's locations are not predicted but actually calculated using the foreground detection method. The next possible players' position is predicted only in cases of occlusion.
- A new robust method for projective transformation of the players in a playfield from a real world image into a reference ground plane image, using bisection method. The suggested method provides excellent results, and requires low computational power since it does not transform the whole pitch but only the coordinates corresponding to the athletes.
- A new, simple but very efficient, method for detecting and extracting the playing terrain in grass grounds or other flat surfaces using only colour information.

7.3 Conclusions and Future Research

Tracking sport players for wireless display is a very important application for all. Computer vision is a large unexplored space, which occupies a huge spectrum of applications. Mobile telephony, and more generally the wireless communication, entered in our life offering us essential services, such as the direct communication. Taking into consideration the human facilitation and the increased interest for athletic events, this work produced a novel object detection and tracking method, using a single fixed camera, which automatically tracks sports players, and identifies their real world positions for describing the contents of an athletic event.

The results of this dissertation are extremely encouraging and provide a powerful first step for future research in this direction. The use of multiple cameras is an already implemented application [7], [10], [31], [34], [35], [36], [37]. Indeed, using multiple cameras in combination with the proposed techniques and architecture might provide better and more accurate results. We should also focus on a most efficient method of estimating the foreground, a process that constitutes one of the more difficult implementations of this study. The development of a reliable tracker is another difficult area, since many players become occluded and merged by other players. The changing velocity and dimension of the players comprised another drawback of the tracker. All these can be confronted using multiple cameras so that multiple-view trackers can solve many existing problems.

This study can be also used for surveillance in Close Circuit Television (CCTV). For this reason, future research should study human motion by introducing a classifier, such as SVM and Hidden Markov, which can recognise the type of the movement at the same time with the detection of the silhouette position.

Overall, an architecture has been presented to facilitate the positioning of sport players using object detection and tracking with single camera. Future work will see the introduction of the tracking analysis using multiple-view cameras, a more complex model for the shape, and the positional behaviour analysis of sports players.

References

[1] Dirk Farin, Peter H. N. de With, Wolfgang Effelsberg, "Robust Background Estimation for Complex Video Sequences". *In Proc. IEEE International Conference on Image Processing (ICIP),* September 2003, pages. 145-148.

[2] Y.L.Tian and A.Hampapur, "Robust Salient Motion Detection with Complex Background for Real-time Video Surveillance", *in Proc. Of IEEE Computer Society Workshop on Motion and Video Computing,* January 2005, p. 30-35.

[3] Kyungnam Kim, Thanarat H. Chalidabhongse, David Harwood, Larry S. Davis, "Background modelling and subtraction by codebook construction". *In Proceedings of the 2004 International Conference on Image Processing*, ICIP 2004, October 24-27, IEEE Volume 5, pages 3061-3064.

[4] Wang D, Feng T, Shum H, Ma S., "A Novel Probability Model for Background Maintenance and Subtraction", *The 15th International Conference on Vision Interface,* 2002.

[5] H. Fujiyoshi, A. Lipton and T. Kanade. "Real-time human motion analysis by image skeletonization". *In Proc. Of the Workshop on Application of Computer Vision,* January 2004, vol.E87-D, no.1, pages 113 – 120

[6] Leonid Taycher, John W. Fisher III, Trevor Darrell. "Incorporating Object Tracking Feedback into Background Maintenance Framework", *7th IEEE Workshop on Applications of Computer Vision*, 5-7 Jan 2005, pages 120-125.

[7] Sachiko Iwase and Hideo Saito, "Tracking Soccer Player Using Multiple Views", *In IARP Workshop on Machine Vision Applications (MVA2002),* 2002, pages 102-105.

[8] J. MacCormick and A. Blake, "*A probabilistic exclusion principle for tracking multiple objects*", In Proc. Int. Conf. On Computer Vision, 1999, pages 518-525.

[9] L.Wang, T.Tan, H.Ning, W. Hu, "Silhouette analysis-based gait recognition for human identification". *Pattern Analysis and Machine Intelligence,* 2003, Vol. 25, pages 1505–1518.

[10] M. Xu, J. Orwell, L. Lowey, D.J. Thirde, "Architecture and Algorithms for Tracking Football Players with Multiple Cameras", *IEE Proceedings – Vision, Image, and Signal Processing,* April 2005, Volume 152, Issue 2, p. 232-241.

[11] C.K. Chui and G. Chen, "Kalman Filter with Real-Time Applications", Springer, 1999.

[12] M. Isard and A. Blake, “Contour tracking by Stochastic Propagation of Conditional Density”, *In Proc. European Conf. Computer Vision*, 1996, pages 343-356.

[13] G.S. Pingali, Y. Jean, and I. Carlbom. “Real time tracking for enhanced tennis broadcasts”, *In IEEE CVPR*, 1998, pages 260–265.

[14] C.J.Needham and R.D. Boyle, “Tracking Multiple Sports Players through Occlusion, Congestion and Scale”, *In Proc. British Machine Vision Conference BMVA01*, Manchester, UK, September 2001. pages 93-102.

[15] Toyama, K., Krumm, J., Brumitt, B., Meyers, B. “Wallflower: Principles and practice of background maintenance”. *In Proceedings of IEEE International Conference on Computer Vision*, 1999, pages 255-261.

[16] D. Gutchess, M. Trajkovic, E. Cohen-Solal, D. Lyons, and A. K. Jain. “A background model initialization algorithm for video surveillance”. *In Proc. Of IEEE Conf. on Computer Vision,* 2001, pages 733-740

[17] Ahmed Elgammal, David Harwood, and Larry S. Davis. “Non-parametric model for background subtraction”, *In Proc. Of 6th European Conference of Computer Vision*, Corfu, Greece, September 1999.

[18] Y. H. Yang and M. D. Levine, “The background primal sketch: An approach for tracking moving objects”, *Machine Vision and Applications*, 1992, vol. 5, pages 17–34.

[19] P.L. Rosin, T. Ellis, “Image Difference Threshold Strategies and Shadow Detection”, *in Proceeding British Machine Vision Conference*, 1995, BMVA Press, pages 347–356.

[20] J. Vass, K. Palaniappan, X. Ahuang, “Automatic Spatio-Temporal Video Sequence Segmentation”, *in Proceeding IEEE International Conference on Image Processing,* October 1998, vol1, pages 958-962.

[21] R.T. Collins, A. J. Lipton, H. Fujiyoshi, T. Kanade, “Algorithms for Cooperative Multi sensor Surveillance”, *Proceeding of IEEE*, 2001, Vol. 89. No.10, pages 1456-1477.

[22] I. Haritaoglu, D. Harwood, L.S. Davis, “W4: Real-Time Surveillance of People and their Activities”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, August 2000, Vol. 22, No.8, pages 809-830.

[23] W. Grimson, C. Stauffer, R. Romano, L. Lee, “Using Adaptive Tracking to Classify and Monitor Activities in a Site”, *In Proceeding of IEEE Conference on Computer Vision and Recognition*, 1998, pages 22-29.

[24] Christof Ridder, Olaf Munkelt and Harald Kirchner, “Adaptive background Estimation and foreground detection using Kalman-filtering” *In Proc.of ICRAM*, 1995,p. 193-199.

[25] Andrea Cavallaro and Touradj Ebrahimi, “Video object extraction based on adaptive backgroumd and statistical change detection”. *In Proc. Of SPIE VCIP*, January 2000, pages 465-475.

[26] Patrick Piscanglia, Andreas Cavallaro, Michel Bonnet and Damien Douzchamps, "High level description of video surveillance sequences", *In Proc. Of ECMAST*, 1999, pages 316-331.

[27] M. Massey and W. Bender, "Salient Stills: Process and Practice", *IBM Systems Journal*, 1996, vol. 35, no 3&4, pages 557-573.

[28] C.R. Wren, A. Azarbayejani, T. Darrell and A. Pentland. "Pfinder: Real-time tracking of a human body", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997, vol 19, pages 780-785.

[29] E. L. Hall, "Computer Image Processing and Recognition". *Academic Press*, 1979.

[30] R. Hertley and A. Zisserman, "Multiple View Geometry in Computer Vision", *Cambridge University Press*, 2000

[31] R. Y. Tsai, "An efficient and accurate camera calibration technique for 3D machine vision", *In Proc. Of IEEE Conf. on Computer Vision and Pattern Recognition*, Miami Beach, FL, 1986, pages 364-374.

[32] T. Bebie and H. Bieri, "SoccerMan: Reconstructing soccer games from video sequences", *In Proc. ICIP,* 1998, pages 898-902.

[33] S.S. Intille and A. F. Bobick, "Closed-world tracking", *In Proc. ICCV*, 1995, pages 672-678.

[34] Y. Seo, S. Choi, H Kim and K. S. Hong, "Where are the ball and players?: Soccer game analysis with colour-based tracking and image mosaic", *In Proc. ICIAP*, 1997, pages 196-203.

[35] S. Iwase and H. Saito, "Parallel tracking of all soccer players by integrating detected positions in multiple view images", *In Pattern Recognition, ICPR*, 2004, Vol 4, pages 751- 754.

[36] Y. Ohno, J. Miura, Y. Shirai, "Tracking Players and Estimation of the 3D Position of the ball in Soccer games", *In Proc. 15th International Conference on Pattern Recognition*, 2002, Vol I, pages 303-306.

[37] O. Utsumi, K. Miura, I. Ide, S. Sakai, and H. Tanaka, "An object detection method for describing soccer games from video," *In Proc. IEEE ICME*, Lausanne, Switzerland 2002.

[38] P. Figueroa, N. Leite, R. Barros, I. Cohen, and G. Medioni, "Tracking soccer players using the graph representation", *In Proc. ICPR*, 2004, pages 787–790.