Tracking Multiple Sports Players for Mobile Display

A. Aristidou
Department of Engineering
Kings College London
University of London
London, UK

P. Pangalos
Department of Engineering
Kings College London
University of London
London, UK

H. Aghvami
Department of Engineering
Kings College London
University of London
London, UK

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 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, an efficient real-time background modeling 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 color. 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. The results of the proposed methods, the players' location and trajectories are presented and discussed.*

Keywords: Kalman filter, condensation, multiple tracking, human detection, background subtraction.

1 Introduction

The increased public interest on sports led to the investment of huge amounts for their transmission via television. Many organizations, such as the FIFA World Cup, the Champions League are in the centre of interest for many sports funs and businessmen. All these, together with the oncoming Olympic Games, motivated us to begin searching efficient ways to present sports events as play animations for web display. The aim of this study is focused on the production of novel object detection and tracking method, using a single fixed camera, which automatically will tracks sports players, and identifies their real positions. 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 wireless services.

2 Literature Review

Many research projects focused on sports player tracking. Some used information such as texture, colour and shape [1]. Colour-based template matching is used in several approaches since it deals with occlusion problems [2], [3]. In contrast, some other studies estimate motion and player position by Kalman [5] or condensation filters [4]. Kalman filter has been used in many tracking applications due to its computational efficiency and its ability to predict future states.

There are several popular methods for tracking moving targets. 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. The methods that used a single camera fail on tracking objects which become occluded. 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 [5], [6], [7]. 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 [5], [8]. 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. The players' trajectory in panoramic view is also the output of a TV sequence in [7]. In [9], a 3D virtual animated view has been generated using two synchronised video sequences. 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.

3 Backround Estimation

Background subtraction is the most popular method for motion segmentation, especially with a relatively stable background [10], [11], [12], [13]. 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.

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. Some added problems concerning background estimation focus on the abrupt global illumination changes or on backgrounds that change their appearance (waves, clouds, etc). Therefore, the background initialisation should involve extracting the background image with many arbitrary moving objects and when the background is visible for short periods of time.

In this project, a robust real-time background modelling and maintenance has been applied, based on the segmentation of input images into stationary and non-stationary blocks, as reported in [14]. The background image is synthesised using the median value of the stationary regions. Thus, 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 concurrently visible and the background can be seen only for a short time.

4 Foreground Region Detection

Based on adaptive background subtraction, a running statistical background value of the intensity of each pixel is maintained for the foreground region detection. To track moving silhouettes from the image in each frame, the background subtraction method and a tracking algorithm 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.

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 [4], 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. These methods also 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.

An additional problem is the shade of the moving object. The shade increases the size of the object and changes its centroids. This can be solved using chromaticity. Chromaticity exploits the fact that an area cast into shadow that often results in a significant change in intensity without much change in chromaticity [1], [15]. Therefore, it's assumed that any significant intensity change without significant chromaticity change could have been caused by shadow. Chromaticity is computed as:

$$r_c = \frac{R}{R+G+B} \tag{1}$$

$$g_c = \frac{G}{R+G+B} \tag{2}$$

$$b_c = \frac{B}{R+G+B} \tag{3}$$

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. The criterion used to detect the moving regions in an image, by differencing current image with a reference background image is:

- **IF** $|cr_c - br_c| > Th$ **AND** $|cg_c - bg_c| > Th$ **AND** $|cb_c - bb_c| > Th$

  THEN → *the pixel is foreground,*
- **OTHERWISE** → *the 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.

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 [16]. 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.

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 is implemented in order to eliminate the noise. This process removes any small holes in the silhouette and smoothes out any interlacing anomalies. Further details are reported in [17] and [18].

Further information needs to be extracted for tracking. Firstly, the centroids of the target image boundary (x_c, y_c) can be determined as:

$$x_c = \frac{1}{N_b}\sum_{i-1}^{N_b} x_i \tag{4}$$

$$y_c = \frac{1}{N_b}\sum_{i-1}^{N_b} y_i \tag{5}$$

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. Some other important features used in this procedure are:

- (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. Hence, each object is presented as:

$$\mathrm{X} = \left(id, x_c, y_c, x_b, y_b, BW_x, BW_y, Area\right) \tag{6}$$

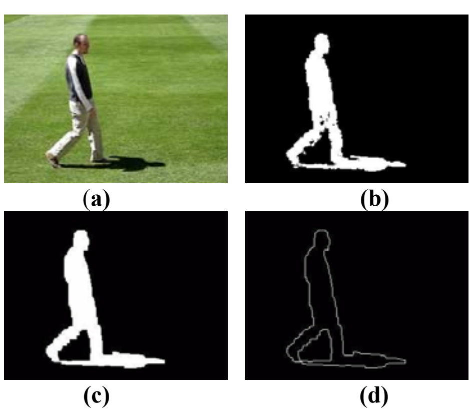

Figure 1: Foreground region pre-processing. **(a)** Current frame, **(b)** foreground detected region in binary pattern, **(c)** the moving foreground region after morphological filter, **(d)** foreground region's border extraction.

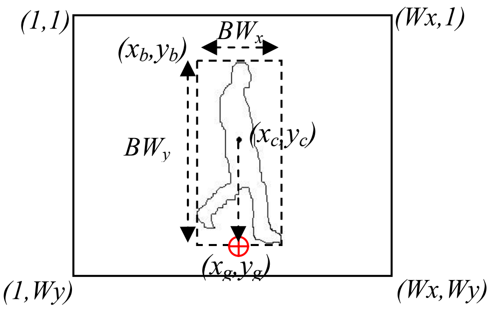

Figure 2: Sample representation of all the needed information.

5 Object Tracking

The aim of object tracking is the establishment of correspondence between blobs across 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 combination of a method that uses blob's information as well as the prediction of the next possible position of a player could be an efficient 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 [19]. However, tracking multiple targets is an initial drawback of this scheme. In [20] 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 [4]. Thus, this is an excellent approach and using this method, the use of multiple static cameras could be avoided, without the problem of occlusion.

5.1 Tracking using cost function

Extra information is needed in order to create the cost function. The cost function can calculate the correlation between blobs across frames. 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, $D = (R + G + B)$, when 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:

$$C_{L_i} = \left| \frac{P_L^{t-1}}{P_i^t} + \frac{R_L^{t-1}}{R_i^t} + \left| \frac{Area_L^{t-1} - Area_i^t}{100} + \frac{V_L^{t-1}}{10} \right| + \frac{D_L^{t-1}}{D_i^t} \right| \tag{9}$$

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. All the parameters of each region are updated using linear low pass filter prediction models as illustrated in the following equations:

Predicted Position:

$$PP_n = P_{n-1} + (t_n - t_{n-1}) \cdot V_{n-1} \tag{10}$$

Predicted Velocity:

$$PV_n = a \cdot V_{n-1} + (1-a) \cdot \frac{(P_n - P_{n-1})}{t_n - t_{n-1}} \tag{11}$$

Predicted Ratio:

$$PR_n = a \cdot R_{n-1} + (1-a) \cdot R_n \tag{12}$$

Predicted Colour:

$$PD_n = a \cdot D_{n-1} + (1-a) \cdot D_n \tag{13}$$

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.

The centroids of the detected foreground regions in the sequences are stored in a trajectory map. 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 kinds of situations, a high-level implementation procedure is triggered to predict the next possible position of that object. 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.

5.2 Tracking using prediction algorithm

A high-level implementation approach based on condensation algorithm is used, in order to cope with complex cases where moving objects are occluded by other objects. Given that players have sudden changes in velocity and in their motion direction, it's essentially to use an effective method based on the condensation algorithm.

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. Therefore, it is essential to establish a simple model that predicts each blob's information of a current frame t from the previous frame $t-1$ [4]. Hence:

$$x_{c,t}^i = x_{c,t-1}^i + \varepsilon_x \tag{14),}$$

$$y_{c,t}^i = y_{c,t-1}^i + \varepsilon_y \tag{15),}$$

$$x_{b,t}^i = x_{b,t-1}^i + \varepsilon_{bx} \tag{16),}$$

$$y_{b,t}^i = y_{b,t-1}^i + \varepsilon_{by} \tag{17),}$$

$$BW_{x,t}^i = BW_{x,t-1}^i + \varepsilon_{BWx} \tag{18),}$$

$$BW_{y,t}^i = BW_{y,t-1}^i + \varepsilon_{BWy} \tag{19),}$$

$$Area_t^i = Area_{t-1}^i + \varepsilon_{Area} \tag{20),}$$

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, 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.

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 $X_t = (x, y) \in \Re^2$ of the player at the next discrete time step. A sample linear stochastic difference equation governs this process:

$$\mathrm{X}_t = \mathrm{X}_{t-1} + w_{t-1} \tag{21}$$

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

$$Z_t = \mathrm{X}_k + v_{t-1} \tag{22}$$

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

$$p(w) \sim N(0, Q) \tag{23}$$

$$p(v) \sim N(0, R) \tag{24}$$

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 info from frame *t-1*. Hence:

$$x^i_{c,t} = (\hat{x}_{c,t} + x^i_{c,t-1})/2 + \varepsilon_x \qquad (25),$$

$$y^i_{c,t} = (\hat{y}_{c,t} + y^i_{c,t-1})/2 + \varepsilon_y \qquad (26),$$

$$x^i_{bx,t} = (\hat{x}_{bx,t} + x^i_{bx,t-1})/2 + \varepsilon_{bx} \qquad (27),$$

$$y^i_{by,t} = (\hat{y}_{by,t} + x^i_{by,t-1})/2 + \varepsilon_{by} \qquad (28),$$

$$BW^i_{x,t} = BW^i_{x,t-1} + \varepsilon_{BWx} \qquad (29),$$

$$BW^i_{y,t} = BW^i_{y,t-1} + \varepsilon_{BWy} \qquad (30),$$

$$Area^i_t = Area^i_{t-1} + \varepsilon_{Area} \qquad (31).$$

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.

Figure 3 presents the multi-object tracking implementation in frame sequences of a football match. Frames 102 and 110 show two players becoming merged and then split (frame 126). The segregation between blobs in frame 102 is achieved because an occluded region is determined, so the high-level implementation procedure is activated.

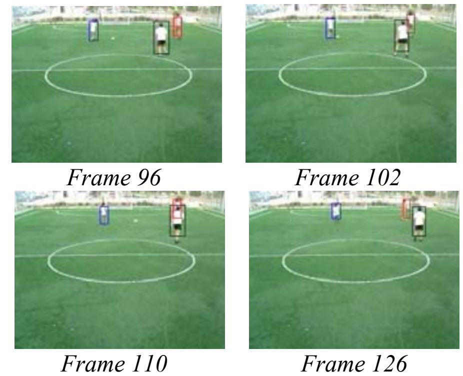

Figure 3: Tracking results across frames.

6 Field Region Extraction

The extract of the field region can be achieved having in mind that the regions that are not included in the terrain and the super-imposed captions should be excluded from the procedure.

Looking at various football fields, distinct characteristics of the terrain can be observed. 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. In sports having a grass arena, the field region is roughly coloured green. Thus, only the green regions of the image should be 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: *Red → 10* (± 10), *Green → 160* (± 50), *Blue → 10* (± 10).

A problem occurs 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. The noise problem can be circumvented using a morphological filter.

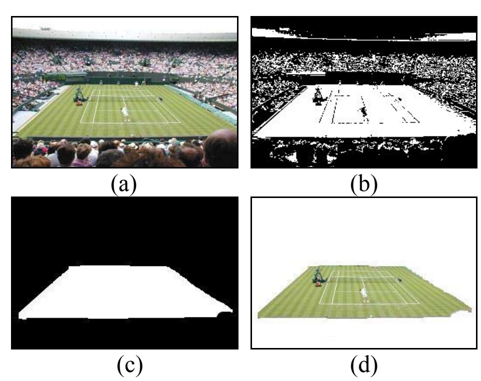

Figure 4: Field Extraction: **(a)** input image, **(b)** image before enhancement by morphological filter, **(c)** image after enhancement by morphological filter, **(d)** Field region extraction.

7 Projective Transformation

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. 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 presents this idea as well as the relationship between depth and size both on the X and Y-axes. The algorithm used is based on the bisection method.

It is clear that the transformation is very sensitive, especially in 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. 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*).

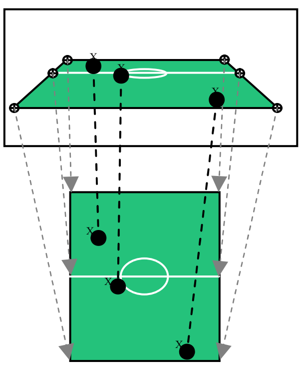

Figure 5: The projective transformation.

8 Results Discussion

Extensive experiments of the proposed system, evaluated in an outdoor 5 a side football match, verified the effectiveness of this model. The location of the *x-y* coordinates of the player's feet, found inside the playfield, and have been converted in a new set of coordinates of a ground platform. The tracker can successfully correlate the players' positions across frames.

In any case, the *id* of each blob is known. Therefore, the trajectory of each player that is stored in a trajectory map can also be presented and can be used for motions analysis, fitness and tactics of teams and players.

The method outlined in this project can be successfully demonstrated in several matches, such as football, tennis and rugby.

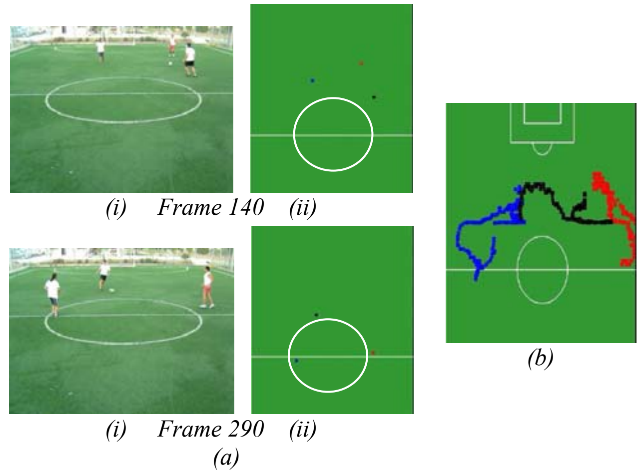

Figure 6: (a) Example of implementation: (i) the real image, (ii) The projected platform. (b) Trajectory of the moving objects

9 Conclusions and Future Research

The main contributions of this study are focused on the extension of the tracking algorithm that uses a cost function of blob information in combination with a condensation filter. In that way, it can cope with complex cases such as when regions enter/exit the scene, change velocity or directions and when they are occluded by other regions. These results are considered more accurate than in any other proposed method, because the player's locations are actually calculated using the foreground detection method and are predicted only in cases of occlusion. The suggested method provides excellent results, and requires low computational power.

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.

10 References

[1]. C. R. Wren, A. Azarbayejani, T. Darrell, A. Pentland. "Pfinder: Real-time tracking of a human body", *IEEE Transactions on PAMI*, 1997, Vol. 19, pages 780-785.

[2]. 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.

[3]. 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.

[4]. C.J.Needham and R.D. Boyle, "Tracking Multiple Sports Players through Occlusion, Congestion and Scale", *In Proc. BMVC*, UK, 2001, pages 93-102.

[5]. M. Xu, J. Orwell, L. Lowey, D.J. Thirde, “Architecture and Algorithms for Tracking Football Players with Multiple Cameras”, *IEE Proc. CISP,* 2005, Vol. 152, p. 232-241.

[6]. Y. Ohno, J. Miura, Y. Shirai, “Tracking Players and Estimation of the 3D Position of the ball in Soccer games”, *In Proc. 15th ICPR*, 2002, Vol I, pages 303-306.

[7]. 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.

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

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

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

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

[12]. W. Grimson, C. Stauffer, R. Romano, L. Lee, “Using Adaptive Tracking to Classify and Monitor Activities in a Site”, *In Proc. of IEEE Conference on CVPR*, 1998, p. 22-29.

[13]. Toyama, K., Krumm, J., Brumitt, B., Meyers, B. “Wallflower: Principles and practice of background maintenance”. *In Proc. of IEEE ICCV*, 1999, pages 255-261.

[14]. Dirk Farin, Peter H. N. de With, Wolfgang Effelsberg, “Robust Background Estimation for Complex Video Sequences”. *Proc. IEEE ICIP,* 2003, pages 145-148.

[15]. Ahmed Elgammal, David Harwood, and Larry S. Davis. “Non-parametric model for background subtraction”, *In Proc. of 6th ECCV*, Greece, 1999.

[16]. E. L. Hall, “Computer Image Processing and Recognition”. *Academic Press*, 1979.

[17]. H. Fujiyoshi, A. Lipton and T. Kanade. “Real-time human motion analysis by image skeletonization”. *In Proc. of the WACV,* 2004, vol.E87-D, no.1, pages 113-120.

[18]. L.Wang, T.Tan, H.Ning, W. Hu, “Silhouette analysis-based gait recognition for human identification”. *PAMI,* 2003, Vol. 25, pages 1505–1518.

[19]. M. Isard and A. Blake, “Contour tracking by Stochastic Propagation of Conditional Density”, *In Proc. ECCV*, 1996, pages 343-356.

[20]. J. MacCormick and A. Blake, “A probabilistic exclusion principle for tracking multiple objects”, In Proc. ICCV, 1999, pages 518-525.