To appear in IEEE International Conference on Advanced Video and Signal-Based Surveillance (AVSS'2005).

A Modular Multi-Camera Framework for Team Sports Tracking

J.B. Hayet * T. Mathes * J. Czyz † J. Piater * J. Verly * B. Macq †

* Montefiore Institute
Université de Liège
B-4000 Liège 1, BELGIUM

† UCL-TELE
Bâtiment Stévin, Place du Levant 2
B-1348 Louvain-la-Neuve, BELGIUM

Abstract

This article presents a modular architecture for multi-camera tracking in the context of sports broadcasting. For each video stream, a geometrical module continuously performs the image-to-model homography estimation. A local-feature based tracking module tracks the players in each view. A supervisor module collects, associates and fuses the data provided by the tracking modules. The originality of the proposed system is three-fold. First, it allows to localize the targets on the ground with rotating and zooming cameras; second, it does not use background modeling techniques; and third, the local tracking can cope with severe occlusions. We present experimental results on raw TV-camera footage of a soccer game.

1 Introduction

Multi-view tracking systems have become major actors in applications such as video surveillance or broadcasting, mainly because they allow to overcome the occlusion problem which is one of the most important challenges faced by tracking systems. In the case of sports broadcasting, multi-view vision systems enable video annotation with data on player motion (positions, velocities...) without using dedicated active sensors (such as RFIDs). This article describes a framework for developing such an application.

Even if our final aim is the implementation of a multi-camera tracking system for general situations (i.e. any place, any kind of target), we focus here on the case where the scene is partially known, e.g. some metric data. In this family of systems, most of the approaches developed in the past address three different problems: (1) how to establish (once or regularly) the geometry of the multi-camera system, (2) how to track targets in 2D video streams and (3) how to fuse data coming from the different streams.

As for the geometrical problem, an initial calibration phase is used in many systems under the assumptions that cameras remain static and that the targets are moving on a plane. In the *closed world* context [6], this calibration estimates image-to-ground homographies [6, 13] from known ground features. Some systems deal with rotating cameras by using rotary encoders together with calibrated internal camera parameters [9]. In the video-surveillance context, some systems estimate for each view a homography to a reference view [1, 14], whereas others use stereo vision [10].

The tracking methods are very diverse in the literature, and most of them are based on background models and on a *global* characterization of targets, e.g. in terms of color distributions and/or shape models [10, 11]. The central problem lies in handling occlusions between overlapping targets, so that using several views may reduce ambiguities. To this end, in the absence of stereo measurements [10], one has to rely on the assumption that the motion space is planar. Then, the occlusion problem is either handled explicitly in the image space with homographies [14] or in the ground plane by recursive filtering [13, 9].

Our work is inspired by the latter paradigm and contributes several original ideas. First, we do not limit ourselves to fixed cameras: Our system architecture is designed for rotating and zooming cameras. To do that, we maintain for each camera an estimate of the homography, and update it as frequently as possible.

This leads us to our second advantage: In contrast to many systems that assume static cameras [13], our system does not use background models but instead takes advantage of a tracking method based on local features only. We claim that such a choice of local appearance allows to cope better with occlusions in the images than many classical approaches.

Furthermore, geometrical and image tracking modules interact so that image-processing parameters can be adapted according to geometry. An example is the automatic adjustment of the detection scale for local features, which is inferred from homographies.

Finally, we heavily rely on a distributed, modular architecture that permits light-weight installations: All modules can run in separate processes on different PCs. The data to be transmitted over the network is restricted to target param-

This work was sponsored by the Région Wallonne under DGTRE/WIST contract 031/5439.

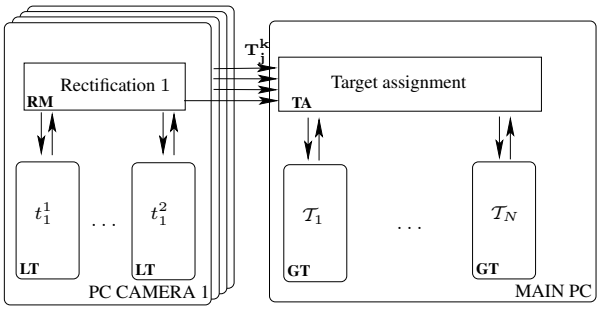


Figure 1: The distributed architecture for fusion of data from different cameras: images are processed by Local Trackers (LT); target parameters are rectified by Rectification Modules (RM), combined by the Target Assignment module (TA) and tracked by Ground Trackers (GT).

eters (positions and covariances) and status information.

Section 2 gives an overview of the architecture we propose. Then, Section 3 describes the algorithms running on each video stream, whereas Section 4 presents the fusion modules. Lastly, some results are presented in Section 5.

2 Overview

For a tracking system to work on several video streams, its architecture has to be distributed and modular. Our system is described in Fig.1. The components on the left address the different autonomous subsystems dealing with single video streams, and communicate with the supervisor component shown on the right. The different components are:

Local image trackers (LT) In each video stream k, n_k targets identified by the label t_j^k and with coordinates $\mathbf{t}_j^k = (u_j^k, v_j^k)^T,\ j = 1, \ldots, n_k$ are tracked, where (u, v) is the image coordinate system.

Rectification modules (RM) For each camera, one of these modules maintains the local homography to the ground plane. They transform image coordinates $\mathbf{t}_j^k$ into ground-plane coordinates $\mathbf{T}_j^k = (X_j^k, Y_j^k)^T,\ j = 1, \ldots, n_k$, where (X, Y) is the ground coordinate system.

A target assignment module (TA) It maintains a global view of the scene as it is in charge of handling the correspondences between locally tracked targets t_j^k and ground targets $\mathcal{T}$.

Ground trackers (GT) They track the N targets identified on the basis of correspondences among the local targets t_j^k. These global targets are identified by the label $\mathcal{T}_n$, for $1 \leq n \leq N$.

All these components are described in detail in the next sections.

3 Local modules

Each local video stream is processed by two separate modules, 2D (image-space) tracking and image rectification. The former tracks 2D targets in the video, whereas the latter transforms the targets image coordinates into model ground coordinates.

3.1 2D tracking

To allow the use of moving cameras, we avoid background models for motion detection. Furthermore, global characterizations of targets such as color distributions or contours are not recommended because of their limited robustness to partial occlusions. We prefer to employ local descriptors. On each target, we extract interest points that are described by their position and local appearance. By learning the spatial relationships between these points from a short training sequence, we are able to construct Point Distribution Models (PDMs) that permits soccer players to be tracked across mutual occlusions, even if they belong to the same team.

An object is tracked by matching the points of its PDM to currently extracted image points and by adapting the model parameters accordingly, including position, scale, orientation and shape. This kind of approach is able to handle camera translation, rotation and zoom. All model points are not required to match to current image points in order to constrain the model. This property makes the tracker robust to partial occlusions, which is particularly interesting for sports applications where there are many interacting targets. However, a limitation of our current algorithm is that we use only 2D models that do not handle major out-of-plane rotations of the players.

The tracking example from Fig. 2 illustrates how multiple targets are robustly tracked across severe mutual occlusions, even without filtering on the model parameters. Although the three players are from the same team, their identities are correctly preserved, because their poses are different at the moment of the crossing. A paper describing the modeling and tracking algorithms has been submitted elsewhere [8].

3.2 Image rectification

Image rectification is an essential task for the system to be able to merge data from different video streams, as, after rectification, all the tracking data are expressed in a common reference frame, i.e. the ground frame.

As we want to cope with typical TV broadcasting setups, the system cannot be calibrated in advance. operators move the camera, zoom onto interesting areas, etc. Moreover, we do not want to rely on motion sensors embedded in cameras [9]. Thus, the homography matrix that maps image points

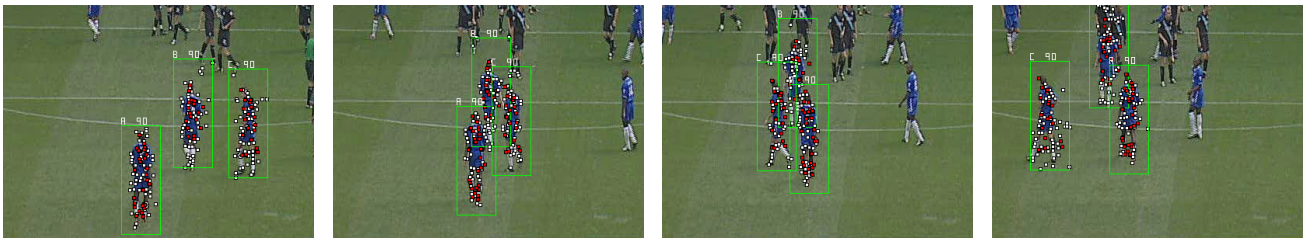
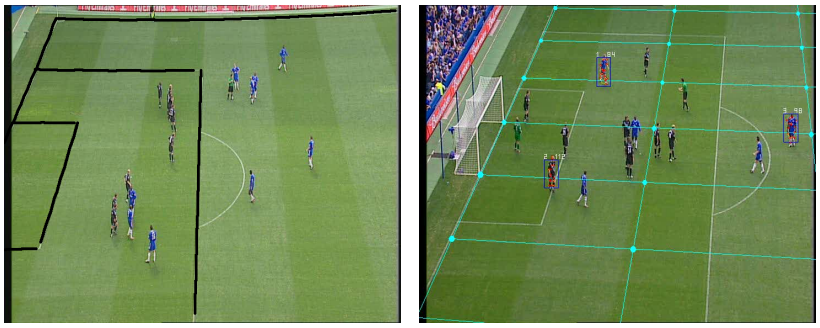


Figure 2: Tracking multiple players across mutual occlusions. White points indicate the currently extracted image points whereas red points indicate the model points. These four keyframes are 40 frames apart[1].



(a) (b)

Figure 3: We combine a feature-based approach and visual odometry to estimate image-to-ground homographies. The left panel (a) shows a reprojection of the model (note that radial distortion is taken into account). The estimated geometrical data are used to adapt the parameters of the tracking algorithm: In the right panel (b), the different scales used for Harris point detection inside each target ROI are illustrated by bright discs whose radius is proportional to σ_h.

to ground points has to be updated at each moment of the sequence based on image information only.

The approach we proposed recently [5] consists of two parts: (1) a line-based technique that assigns detected lines to a model and tracks these lines in order to update the homography and (2) a point-based technique that allows to estimate increments of homographies whenever no line features are visible. By definition, the latter accumulates inter-image homographies, so that a drift in the final estimate is unavoidable. However, occasional calls of the former suffice to eliminate this drift.

Figure 3(a) gives an example of line-based model reprojected onto the image under the estimated homography. The only assumptions we use in this step are that we know the model of the ground and that the cameras are only rotating and zooming (i.e., all points in successive images are transformed by the same incremental homography). Note that merely a partial knowledge the geometry of the line is required; the rest are estimated from distance cross ratios [5].

In this way, each camera k updates an estimate $\mathbf{H}^k$ of its image-to-ground homography. It transforms the 2D tracked targets positions $\mathbf{t}_j^k$ into 2D ground estimates $\mathbf{T}_j^k$,

$$\begin{pmatrix} \mathbf{T}_j^k \\ 1 \end{pmatrix} \sim \mathbf{H}^k \begin{pmatrix} \mathbf{t}_j^k \\ 1 \end{pmatrix}.$$

3.3 Adaptation of local detection scale

One of the main problems of the point-based tracking algorithm lies in the scale variation of interests points along a video sequence. Indeed, as the camera moves or zooms, the integration/derivation scale of Harris interest points evolves. The same problem exists for the line features that are tracked for maintaining the homography as explained in Section 3.2.

Let us assume that we know the initial scale $s(p_0)$ of a local feature p_0 detected at frame 0. We want to update this scale without having to use the costly methods commonly used to compute it [7]. To this end, we define a scale function

$$\sigma_h(\mathbf{H}, P) = \max_{Q \in \partial B(P)} \|\mathbf{H}^{-1}P - \mathbf{H}^{-1}Q\|,$$

[1] We used color figures to improve readability. For the final version, we will make sure they reproduce well in grayscale.

where $\mathbf{H}$ is an image-to-ground homography matrix (the camera superscript is abandoned here for clarity), P and Q are points on the reference plane and $\partial B(P)$ denotes the set of vertices of a reference square centered at P. This function σ_h is a rough approximation of s that does not take into account any 3D motions of the targets. Then the scale of the tracked point p_0 can be updated at time τ as

$$s(p_\tau) = s(p_0)\frac{\sigma_h(\mathbf{H}_\tau, \mathbf{H}_\tau p_\tau)}{\sigma_h(\mathbf{H}_0, \mathbf{H}_0 p_0)}.$$

This scale is then used to detect Harris points in the next frame. In our current implementation, the points associated with a given target are all detected at the same scale, so that the scale update is computed once per target. Figure 3(b) gives an illustration of σ_h values.

3.4 Propagation of uncertainty

Data fusion requires well-defined uncertainty measures $\mathbf{R}_j^k$ on $\mathbf{T}_j^k$. These uncertainties stem from two principal independent sources that include the uncertainty in the locations of the features used to compute/update the scene geometry (a 9×9 uncertainty matrix $\mathbf{V}^k$ on $\mathbf{H}^k$) and the uncertainty in the locations of the tracked points (a 2×2 uncertainty matrix $\mathbf{W}_j^k$ on $\mathbf{t}_j^k$).

The matrix $\mathbf{W}_j^k$ is assumed to be constant. As for $\mathbf{V}^k$, we use the framework proposed elsewhere for the estimation of the uncertainty in vectors computed by SVD based on perturbation theory [3]. As a result, we obtain the following uncertainties on $\mathbf{T}_j^k$:

$$\mathbf{R}_j^k = \mathbf{J}_V \mathbf{V}^k \mathbf{J}_V^T + \mathbf{J}_W \mathbf{W}_j^k \mathbf{J}_W^T$$

where $\mathbf{J}_V$ (resp. $\mathbf{J}_W$) is the Jacobian of the ground-to-image mapping with respect to the homography (resp. image points).

4 Target assignment and ground trackers

The supervisor module has to (1) gather and synchronize the target coordinates provided by the local trackers, (2) associate the coordinates (3) and fuse the coordinates in order to determine a global estimate of the target positions in world coordinates.

4.1 Data synchronization

For such a framework to handle information coming from physically distinct sources, data synchronization is a fundamental problem. The scheme we employ is depicted in Fig. 4. All target data are time-stamped by the local trackers. Temporally corresponding data from different sources may arrive at the supervisor at different times due to differences in processing time, network delays, etc. The supervisor program receives the data coming from the various sources in a fixed-size buffer. Thus, all data time-stamped τ_0 are awaited during a fixed interval $\delta\tau$ after which the available data are processed by the assignment and fusion modules. Any data arriving late are discarded.

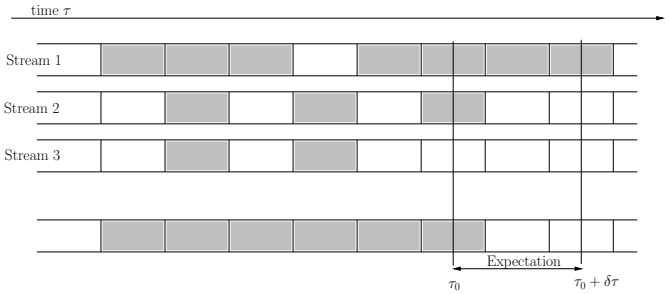


Figure 4: Data from different streams are stored in a temporal buffer where a delay $\delta\tau$ is applied before considering all the data corresponding to an instant τ_0.

4.2 Target assignment (Data association)

The purpose of the assignment module is to establish the correspondence between the local targets t_j^k in each video stream k and the target $\mathcal{T}_n$ in the physical world. As did Snidaro et al. [12], we adopt the Kalman filter framework to solve the problem. Measurements coming from the local trackers are used to update the state vector $\mathbf{x}_n = (X, Y, \dot{X}, \dot{Y})^T$ of target $\mathcal{T}_n$, comprising its position and velocity.

The module takes as input a series of lists of ground coordinates, each list coming from a local tracker analyzing one stream, and composed of ground coordinates of the targets detected in this stream. Thus for stream k we have the list of coordinates $\mathbf{T}_1^k, \ldots, \mathbf{T}_{n_k}^k$ with $\mathbf{T}_j^k = (X_j^k, Y_j^k)^T$ and the associated covariances $\mathbf{R}_1^k, \ldots, \mathbf{R}_{n_k}^k$, $k = 1, \ldots, K$ where K is the number of video streams (the time subscript is omitted for clarity). The computation of the $\mathbf{R}_j^k$ has been described in Section 3.4.

The problem of target assignment can be stated as follows. Given a target $\mathcal{T}_n$ ($n = 1, \ldots, N$), find in each stream k the indices λ_n^k such that $\mathbf{T}_{\lambda_n^k}^k$ corresponds to $\mathcal{T}_n$. The assignment is solved by constructing a cost matrix $\mathbf{A} = (a_{jn})$ in which each entry a_{jn} is the Mahalanobis distance between $\mathbf{T}_j^k$ and the predicted position of target $\mathcal{T}_n$. The cost matrix is expanded to take into account invisible objects and spurious detections. A legal assignment specifies only a single correspondence in any row or column of the matrix, and

the assignment cost is simply the sum of the costs of these elements. Finding the best assignment is then a matter of finding the assignment that minimizes this sum, which can be solved efficiently by combinatorial optimization[2].

4.3 Data fusion

Once the correspondence is found, the coordinates $\mathbf{T}^k_{\lambda^k_n}$ measured by different cameras can be fused. For each target $\mathcal{T}_n$, fusion is done using

$$\mathbf{z}_n = \left(\sum_k (\mathbf{R}^k_{\lambda^k_n})^{-1} \right)^{-1} \sum_k (\mathbf{R}^k_{\lambda^k_n})^{-1} \mathbf{T}^k_{\lambda^k_n}$$

in order to obtain a fused measurement $\mathbf{z}_n$ [4], which is then used in a standard Kalman filter to update the state $\mathbf{x}_n$ of target $\mathcal{T}_n$. To derive this equation, it is assumed that the noise on the ground coordinates coming from different video streams is independent.

5 Results

For our experiments, we have used real soccer TV footage consisting of a full game seen by four cameras. In practice, in most of game situations, no more than two or three views overlap. Figure 5 illustrates a case of overlapping views with three cameras, namely A, B and M, one located at the opposite goal and the two others on the side of the field.

The three pictures of Fig. 5 show the tracking performance at frame 128 of a 400-frame corner sequence. The scene is difficult to track since many targets of similar appearance overlap in the images. In Fig. 5, the yellow dots show the tracks over the previous frames as detected by the local trackers. Moreover, the cameras move, so that homography updates are critical. In these images, these updates are illustrated by the red lines, which are the back-projection of the line features of the ground model.

Data association for the same frame (128) is shown in Fig. 6. Note that this particular frame presents multiple sources of ambiguity: players from the same team are close to each other, some measurements have large uncertainties. Nevertheless, target assignment is correct in all cases.

Figure 7 represents the trajectories of the tracked players. Each color represents a different target, identified by a number. These trajectories are also back-projected into the original images, indicated by the white curves in Fig. 5. The global behavior is satisfactory even under severe occlusions in camera A, but all trajectories are as expected, including those of the partially occluded players 2, 5, 6 and 1.

Presently, one of the main limitations is that the projection to the ground plane of the image-space target coordinates is taken to be the bottom of the region of interest surrounding the tracked model. This results in high uncertainties in the ground-plane position, and can lead to bad associations in some cases, where the system creates new targets where it should not. Two solutions are currently being investigated: the use of contours to delimit the targets, and an exploitation of the PDM.

One point not addressed here is the initialization of the targets. In the example we give, this is done by hand. However, automatic initialization of PDMs is currently investigated through clustering of points with similar motion.

Videos illustrating the individual performances of both modules as well as overall system behavior can be found at https://biprossix.montefiore.ulg.ac.be/trictrac/videos.php.

6 Concluding remarks

In this paper, we have presented an original and efficient framework for multi-camera tracking. It can cope with rotating and zooming cameras by continuously maintaining an image-to-model rectification for each video stream. It uses a novel approach for tracking in image space based on local features grouped as Point Distribution Models. Our system takes advantage of geometrical data as, for instance, local scales can be computed and updated from homographies. We limited our first experiments to soccer TV footage with occlusion situations, for which very promising results have been presented.

A principal goal for future work is that the trackers maintain the identity of the individual targets for as long as possible. One immediate improvement is to introduce *trajectory-based* reasoning to the data-association step in addition of the current point-based stochastic reasoning. Moreover, rectification can be improved to work on longer runs by adding accidental point features to the ground model. We will apply this framework in various types of applications, in particular in video-surveillance scenarios.

References

[1] J. Black, T.J. Ellis, and P. Rosin. Multi view image surveillance and tracking. In *Proc. of IEEE Workshop on Motion and Video Computing*, pages 169–174, 2002.

[2] I. Cox and S. Hingorani. An efficient implentation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(2):138–150, 1996.

[3] A. Criminisi, I. Reid, and A. Zisserman. A plane measuring device. In *Proc. of the British Machine Vision Conference*, 1997.

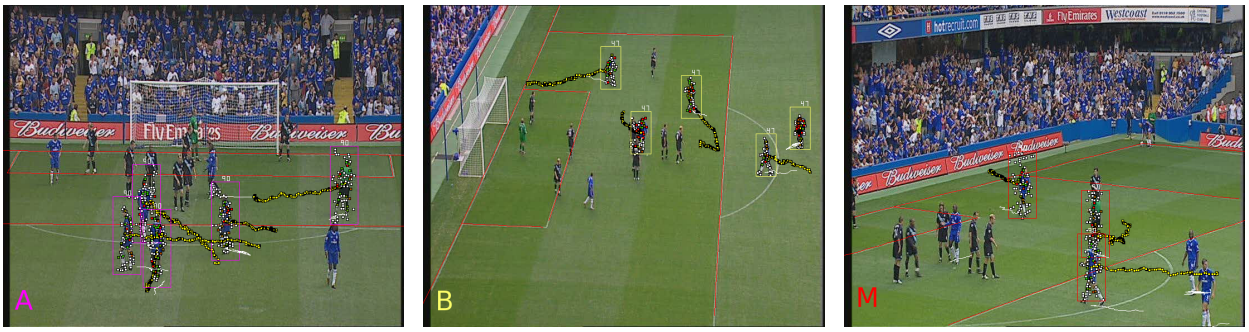


Figure 5: Tracking multiple players in multiple views. The scene is seen at frame 128 from cameras A, B and M. Yellow dots represent the evolution of the targets' image coordinates, and ROIs have colors specific to each camera. Red lines give the back-projection of the ground model by $\mathbf{H}_k$, and white curves are the back-projection of the fused ground trajectories.

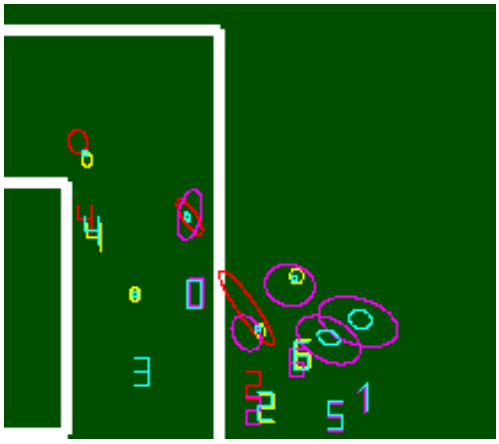


Figure 6: Measurements and their uncertainties from each camera (same colors as in Fig. 5). A-posteriori state estimates appear in blue. Numbers identify individual targets.

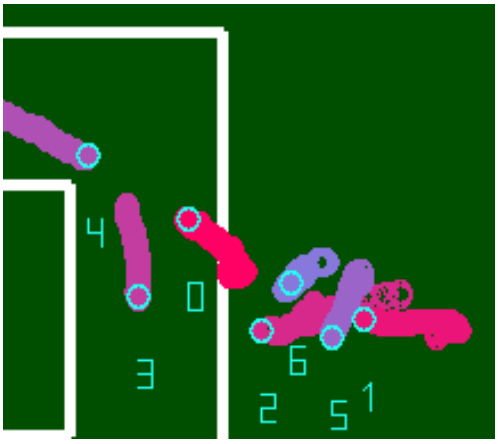


Figure 7: Evolution of the fused positions in the ground plane at frame 128. One color represents one ground target, and blue circles represent the current position.

[4] J. Gao and C.J. Harris. Some remarks on Kalman filters for the multisensor fusion. *Information Fusion Journal*, 3:191–201, 2002.

[5] J.B. Hayet, J. Piater, and J. Verly. Robust incremental rectification of sports video sequences. In *Proc. of the British Machine Vision Conference (BMVC'04)*, 2004.

[6] S. Intille and A. Bobick. Closed-world tracking. In *Proc. of the Int. Conf. on Computer Vision (ICCV'95)*, pages 672–678, 1995.

[7] T. Lindeberg. Feature detection with automatic scale selection. *Int. Journal on Computer Vision*, 30(2):79–116, November 1998.

[8] T. Mathes and J. Piater. Robust non-rigid object tracking using point distribution models. To appear in British Machine Vision Conf. (BMVC'05), 2005.

[9] T. Misu, S. Gohshi, Y. Izumi, Y. Fujita, and M. Naemura. Robust tracking of athletes using multiple features of multiple views. *Journal of WSCG*, 12(1–3):285–292, 2004.

[10] A. Mittal and L. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene. *Int. Journal of Computer Vision*, 51(3):189–203, 2003.

[11] C.J. Needham and R.D. Boyle. Tracking multiple sports players through occlusion, congestion and scale. In *Proc. of the British Machine Vision Conference (BMVC'01)*, pages 93–102, 2001.

[12] L. Snidaro, G. Foresti, R. Niu, and P. Varshney. Sensor fusion for video-surveillance. In *Proc. of the 7th Int. Conf. on Information Fusion*, pages 739–746, 2004.

[13] M. Xu, L. Lowey, and J. Orwell. Architecture and algorithms for tracking football players with multiple cameras. In *IEE Intelligent Distributed Surveilance Systems*, 2004.

[14] Z. Yue, S. Zhou, and R. Chellappa. Robust two-camera visual tracking with homography. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP'04)*, 2004.