Unconstrained Multiple-People Tracking

Daniel Rowe[1], Ian Reid[2], Jordi Gonzàlez[3], and Juan Jose Villanueva[1]

[1]Computer Vision Centre, Universitat Autònoma de Barcelona, Spain
[2]Active Vision Lab, Oxford University, United Kingdom
[3]Institut de Robòtica i Informàtica Industrial, UPC, Barcelona, Spain

Abstract. This work presents two main contributions to achieve robust multiple-target tracking in uncontrolled scenarios. A novel system which consists on a hierarchical architecture is proposed. Each level is devoted to one of the main tracking functionalities: target detection, low-level tracking, and high-level tasks such as target-appearance representation, or event management. Secondly, tracking performances are enhanced by on-line building and updating multiple appearance models. Successful experimental results are accomplished on sequences with significant illumination changes, grouping, splitting and occlusion events.

1 Introduction

Multiple human-beings tracking has become an active research field among the computer-vision community. This interest is motivated by an increasing number of applications related to Human Sequence Evaluation (HSE) [6]. Despite this interest, this still constitutes an open problem far from been solved. People tracking involves dealing with non-rigid targets whose dynamics are subject to sudden changes. In open-world applications, the number of agents within the scene may vary over time, and neither their appearance, nor their shape can be specified in advance. In unconstrained environments, the illumination and background-clutter distracters are uncontrolled, affecting the perceived appearance, which depends on issues such as the agents' position or orientation. Finally, agents interact among themselves, grouping and splitting, and causing occlusions.

Our goal is to implement and experimentally verify a novel approach which deals with the aforementioned difficulties. As a result, agents' trajectories will be obtained, as well as quantitative and qualitative information about their state at any time —such as their speed or whether they are being occluded. This paper is organized as follows: section 2 covers the most common current approaches; section 3 outlines the proposal; section 4 describes the low-level modules, whereas section 5 details the high-level appearance tracker; finally, section 6 shows some experimental results, and section 7 concludes this paper.

2 Related Work

Tracking can be carried out relying either on a bottom-up or a top-down approach. The former consists on foreground segmentation, and a subsequent target association, while the latter is based on complex shape and motion modelling.

K. Franke et al. (Eds.): DAGM 2006, LNCS 4174, pp. 505–514, 2006.

Motion Segmentation can be performed by means of optical flow, background subtraction, or frame differencing. Correspondences can be accomplished using nearest neighbour techniques, or by means of Data Association filters [2]. A prediction stage is usually incorporated, thereby providing better chances of tracking success. Filters such as the Kalman filter, or extensions such as the EKF or UKF are commonly used. More general dynamics and measurement functions can be dealt with by means of Particle Filters (PF) [1].

On the other hand, high-level approaches rely on accurate target modelling [5]. Thus, complex templates and high-level motion patterns are a-priori learned, and used to reduce the state-space search region. Contour tracking have been widely explored [9], although this may be inappropriate in crowded scenarios with multiple target-occlusions. BraMBLe [8] is an interesting approach to multiple-blob tracking which models both background and foreground using MoG. However, no model update is performed, there is a common foreground model for all targets, and it may require an extremely large number of samples, since one sample contains information about the state of all targets. Nummiaro et al. [10] use a PF based on colour-histogram cues. However, no multiple-target tracking is considered, and it lacks from an independent observation process, since samples are evaluated according to the predicted image region histograms.

Comaniciu et al. [4] introduce an attractive technique —called mean shift— which tackles target localisation by performing a gradient-descent search on a image region of interest. However, their method tracks just one target, initialised by hand, and the appearance model is never updated. Collins et al. [3] present an effective enhancement with on-line selection of discriminative features. It aims to maximise the distinction between the target appearance and its surroundings. Still, it tracks just one target, initialised by hand and which may suffer from model drift. In both cases, just rigid target regions are tracked, and since multiple-target tracking is not considered, interaction events are not studied.

3 Approach Outline

Non-supervised multiple-human tracking is a complex task which demands a structured framework. This work presents a hierarchical system whose levels are devoted to the different functionalities to be performed, see Fig. 1.

Reliable target segmentation is critical in order to achieve an accurate feature extraction without considering prior knowledge about the potential targets, specially in dynamic scenes. However, complex agents who move through cluttered environments require high-level reasoning. Thus, this proposal consists on a bottom-up approach, whose results are eventually refined by a top-down process.

The lower level performs target detection. The first module accomplishes the segmentation task, while the second one filters the obtained image masks, extracts object blobs, and obtains object representations which can be handled by low-level trackers. The latter establish coherent target relations between frames. Firstly, *gates* —regions where the observations are expected to appear— are computed. Subsequently, *data association* is performed by setting correspondences

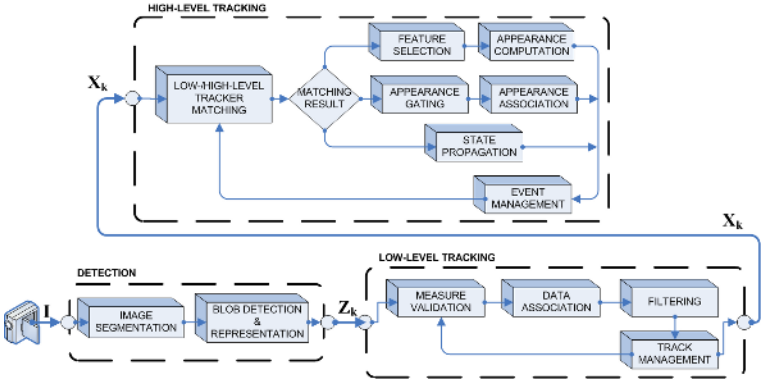


Fig. 1. System architecture

between observations and trackers. Afterwards, *filtering* is carried out by estimating new target states according to the associated observations using a bank of Kalman filters. Finally, the *track-management* module (i) initiates tentative tracks for those observations which are not associated; (ii) confirms tracks with enough supporting observations; and (iii) removes low-quality ones. Results are forwarded to high-level trackers, and fed back to the measure-validation module.

Confirmed low-level tracks are associated to high-level trackers. Hence, tracking events can be managed, and target tracking can be achieved even when image segmentation is not feasible, and low-level trackers are removed (during long-duration occlusions or grouping). Therefore, whenever the track is stable, the target appearance is computed and updated; those high-level trackers which remain orphans are processed to obtain an appearance-based data association, thereby establishing correspondences between lost high-level trackers and new ones; finally, those targets which have no correspondence are propagated according to the learned motion model. The *event* module determines what is happening within the scene, such as target grouping or entering the scene. These results are fed back allowing low-/high-level tracker matching.

4 Blob Detection and Low-Level Tracking

The first level aims to detect targets within the scene. Two modules are implemented to segment the image, and obtain a suitable object representation. Image segmentation is performed following the method proposed by Horprasert et al. [7] which is based on a colour background-subtraction approach. The background is statistically modelled on a pixel-wise basis, using a window of N frames. During this training period, the mean $\mathbf{E}_i$ and standard deviation σ_i of each ith-pixel RGB-colour channel is computed. Two distortion measures are established: α, the brightness distortion, and CD, the chromacity distortion. The variation over time of both distortions for each pixel is subsequently computed, and used as normalising factors for α and CD, so that a single threshold —automatically

computed according to the learned pixels distribution— can be set for the whole image. This 4-tuple constitutes the pixel background model.

Pixels are classified into five categories, depending on their chromacity and brightness distortion: foreground, dark foreground (where no chromacity cues can be used), shadows, highlights, and normal background. Foreground blobs are subsequently detected: both foreground maps are fused; morphological operations are applied and a minimum-area filter is used; and remaining pixels are grouped into labelled blobs, their contours are extracted, and an ellipse representation is computed. Thus, the j-observed blob at time t is given by $\mathbf{z}_j^t = \left(x_j^t, y_j^t, h_j^t, w_j^t, \theta_j^t\right)$, where x_j^t, y_j^t represent the ellipse centroid, h_j^t, w_j^t are the major and minor axes, and θ_j^t the ellipse orientation.

The target state is then estimated by filtering the sequence of noisy measures. In this work, it is assumed that human beings move slowly enough compared to the frame rate. Since their long-run dynamics are hardly predictable, a first-order dynamic model is adopted. This assumption holds in most HSE applications. The observation vector at time t is given by the blob detection module. The target state is then defined by $\mathbf{x}_j^t = \left(x_j^t, \dot{x}_j^t, y_j^t, \dot{y}_j^t, h_j^t, \dot{h}_j^t, w_j^t, \dot{w}_j^t, \theta_j^t\right)$. Thus, a constant-speed approach is used, where the acceleration is modelled as WAGN.

In a multiple-target tracking scenario, numerous observations may be obtained at every sampling period. Measure validation consists in establishing the regions where the target observations are expected. Thus, gates are set according to the innovation covariance matrix $\mathbf{S}_k$, and a specific Mahalanobis Square Distance (MSD), thereby defining an ellipsoid which encloses a probability mass given by the confidence interval associated with the MSD. This means that measures can be validated for a given confidence interval by calculating the MSD between the predicted observation and the actual one, and comparing this value with the Mahalanobis radius for this confidence interval.

Measures are associated to the nearest tracker in whose gate they lie, since observations are usually just within one target gate. This is intrinsic to motion segmentation: close targets are likely to be segmented just as one blob corresponding to the group. A bank of Kalman filters estimates the state of all targets detected within the scene. When no observation is associated to a particular target, its state is propagated according to the dynamic model.

Target tracks are instantiated, confirmed and removed according to the values of two indicators: the square root of $\mathbf{S}_k$ determinant, and the observation MSD. The former is related to the track uncertainty given by the variance of the eigenvector dimensions. While an observation is associated, $|\mathbf{S}_k|^{\frac{1}{2}}$ will decrease to its asymptotic value, and the time taken depends only on the system model. It is a reliable indicator of how many observations have been consecutively associated, without setting thresholds or specifying cases. The quality of the observation is taken into account by evaluating the MSD of each target associated observation. Therefore, a track is instantiated every time an observation remains orphan. When $|\mathbf{S}_k|^{\frac{1}{2}}$ and the MSD value indicate that the track is stable, the tracker is confirmed. If $|\mathbf{S}_k|^{\frac{1}{2}}$ grows far beyond reasonable values, the tracker is removed.

5 High-Level Appearance Tracker

The aforementioned bank of Kalman filters estimates the state of multiple targets. However, it cannot cope with those situations where segmentation fails, such as grouping events, or non-smooth changes in position or shape. These issues are addressed by implementing high-level trackers which include information relative to the target appearance and tracking events. Unfortunately, the target appearance cannot be specified in advance, and it should be continuously updated, since it strongly depends on the target position and orientation, and on the light sources. Further, in order to be able to track them when target segmentation is not feasible, it is modelled taking into account the local clutter.

In this work, the appearance-modelling approach presented by Collins et al. [3] is followed. This uses multiple colour features, which are evaluated and ranked. However, contrary to their method, a pool of features is now maintained, and smoothed characteristics are computed. Thus, the initialisation is solved, and tracker association is feasible once the event that cause the target loss is over. Possibilities of inconsistent localisations due to feature switch are minimised by introducing the distinction between long-run features and the current best ones.

5.1 Tracker Matching

This module performs the matching between low- and high-level trackers. Whenever a low-level tracker is confirmed, a high-level tracker is instantiated and associated. In case that the new-born tracker does not collide with two or more existing trackers, the target appearance will be computed (see Fig 1). In other case, it is marked as a group tracker. In subsequent tracker matchings, high-level tracker parameters relative to the target position and shape are updated. Further, while the track is still confirmed, appearances will also be updated.

Low-level trackers are removed during long-duration occlusions or groupings, since no observation is received, and the track loses confidence. In this case, the high-level tracker is not matched to any low-level tracker. Then, the system tries to associate it to new-born ones, presumably created once the event is over. If there are no tracker candidates, or they are not similar enough in the appearance sense, their state is propagated according to the learned motion model.

5.2 Feature Selection

The target appearance is represented using colour histograms, since they less sensitive to rotations in depth or target deformations. Features are selected from a set of independent linear combinations of RGB channels, including raw R, G, and B, intensity, or chrominance approximations. Features are then normalised to the range $[0 - 255]$, and subsequently discretised into 64 bins. This is a sensitive decision: a low number of bins prevents from target-clutter disambiguation; on the other hand, a high value favours erroneous representations that appear when distributions are estimated from an insufficient number of samples. The

$i-$feature target histogram is given by $\mathbf{p}^i = \left\{p_k^i; k = 1:K\right\}$, where K is the number of bins. The probability of each feature is calculated as:

$$p_k^i = C^i \sum_{a=1}^{M} \delta\left(b\left(x_a\right) - k\right), \tag{1}$$

where C^i is a normalisation constant which ensures $\sum_{k=1}^{K} p_k^i = 1$, δ the Kronecker delta, $\{x_a; a = 1:M\}$ the pixel locations, M the number of target pixels, and $b\left(x_a\right)$ a function that associates pixels to corresponding bins. In a similar way, $\mathbf{q}^i$ represents the $i-$feature background histogram, computed from the background model. Then, log-likelihood ratios of each feature are computed as:

$$L^i\left(k\right) = \log \frac{\max\left(p_k^i, \epsilon\right)}{\max\left(q_k^i, \epsilon\right)}, \tag{2}$$

where ϵ is set to the minimum histogram value to prevent dividing by zero or taking the logarithm of zero, but avoiding also magnifying the corresponding log-likelihood value. Thus, shared colour bins have a log-likelihood close to zero, whereas foreground bins have a positive one, and background bins a negative one. Features are then evaluated according to the variance-ratio of the log-likelihood:

$$VR^i\left(L; p, q\right) = \frac{\mathrm{var}\left(L^i; \left(p^i + q^i\right)/2\right)}{\mathrm{var}\left(L^i; p^i\right) + \mathrm{var}\left(L^i; q^i\right)}, \tag{3}$$

which maximises the inter-class variance —background and target bin clusters—, while minimises the intra-class variance. Thus, features can be ranked according to their variance ratio: the higher, the better.

5.3 Appearance Computation

Contrary to the work of Collins, long-run features are kept and smoothed. These will be crucial for target loss recovery. Further, by smoothing the histograms the representation is less sensitive to possible localisation errors, and sudden and temporal appearance changes due to illumination fluctuations. A pool of $M + N$ features is kept. These are the best M features at time t, and the best N long-run features: those which have been at top of the feature rank more times. These features are only dropped when new features enter the pool, and overcome them. For each M feature, the mean appearance histogram is recursively computed:

$$\mathbf{m}_t^i = \mathbf{m}_{t-1}^i + \frac{1}{n_i}\left(\mathbf{p}_t^i - \mathbf{m}_t^i\right), \tag{4}$$

where n_i is the number of times that the histogram has been updated. Similarity between two histograms is computed using the *Bhattacharyya distance*

$d_B = \sqrt{1 - \sum_{k=1}^{K} \sqrt{p_k q_k}}$. A similarity criterion must establish when two histogram are close enough. Thus, the mean and variance of d_B between the smoothed histogram and the new one are also computed and updated:

$$\mu_t^i = \mu_{t-1}^i + \frac{1}{n-1}\left(d_{B,t}^i - \mu_t^i\right), \tag{5}$$

$$\sigma_t^2 = \frac{n-3}{n-2}\sigma_{t-1}^2 + (n-1)\left(\mu_t^i - \mu_{t-1}^i\right). \tag{6}$$

In this way, the Bhattacharyya distance distribution can be parameterised.

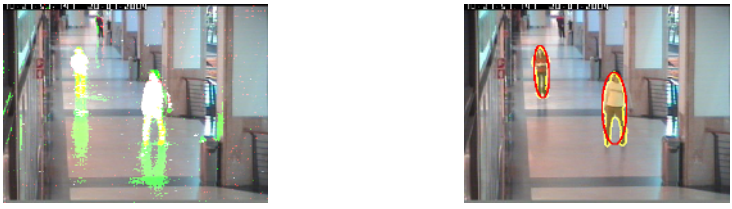
5.4 Appearance Association

Low-level trackers lost their track during long-duration segmentation failures, such an occlusion event. Once the event is over, the target is again detected and a new tracker is instantiated. When this track become stable, it is confirmed and a high-level tracker is created. The former high-level tracker —and the target appearance models— were propagated. A tracker association process is performed, and the system concludes that both trackers are in fact representing the same target. This is done as follows: new-born trackers are handled as observations, and they are gated according to the lost trackers in the feature space. Thus, coincident features between both trackers are selected. Since feature selection depends on the local environment, and the targets move while they are grouped, the feature pool is subject to changes. However, the assumption that some long-run features are still good enough to be selected holds in most scenarios.

The Bhattacharyya distance between the histograms of each coincident feature is evaluated. Those which correspond to the the lost tracker are in fact smoothed models computed while the segmentation was reliable. Features are gated using the previously calculated mean and variance of the Bhattacharyya distance. Finally, the tracker is associated to the nearest one, according to the Bhattacharyya distance, within the gate. If none of the features is within the gate of the lost tracker, a new association process is tried at the next time step.

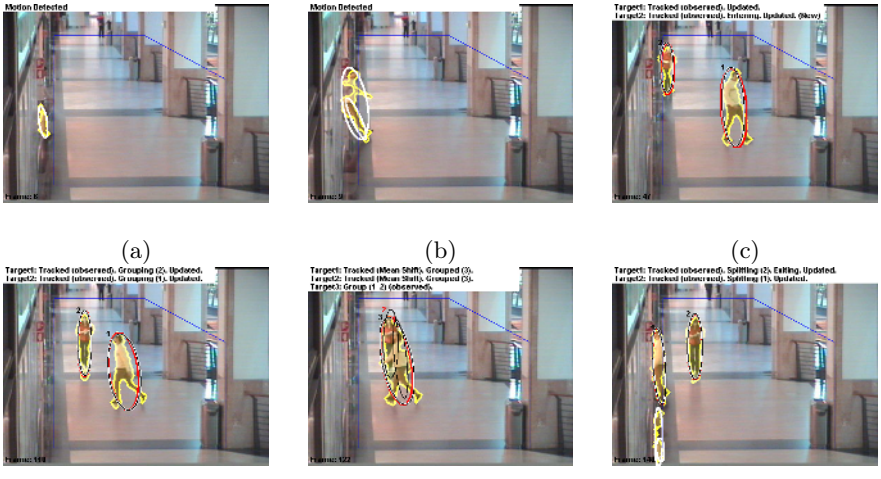
5.5 Event Management

Six significant states are defined: single target, target grouping, grouped and splitting, and target entering and exiting the scene. Once the target position and size is estimated, a collision map is computed. Thus, when two single targets are colliding, their state change into *grouping*. If they also collide with a confirmed group tracker, their state is set to *grouped*. Once they no longer collide with a confirmed group tracker, their state change to *splitting*. If they stop colliding at any state, they become *single* again. The collision map is used also to determine whether a new-born tracker represents a group.



(a) (b)

Fig. 2. (a) Segmentation: foreground pixels are painted on white, while those ones classified as dark foreground are on yellow, shadows on green, and highlights on red. (b) Detection: red ellipses represent each target, and yellow lines denote their contour.



(a) (b) (c)

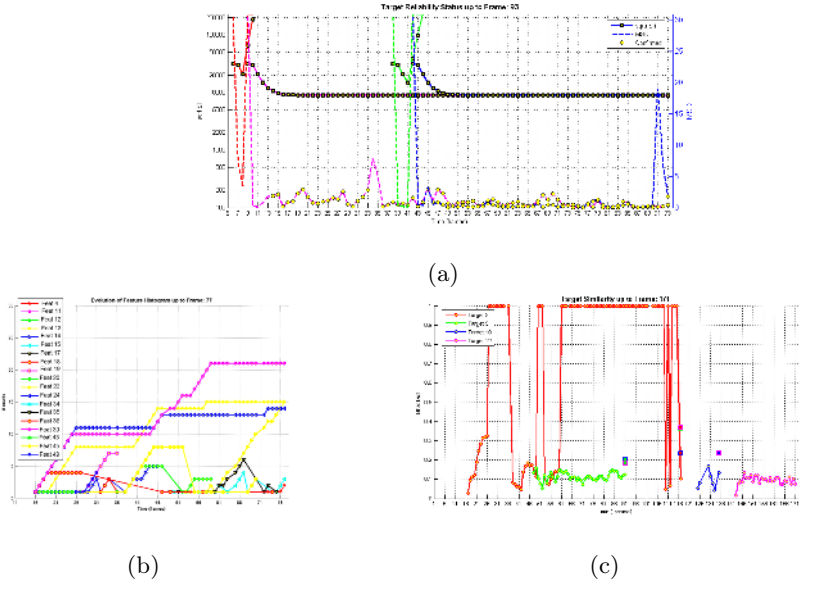
(d) (e) (f)

Fig. 3. Tracking results: red ellipses denote detections, whereas white and black ones are low- and high-level tracker estimates, respectively. The blue box denotes the ROI.

6 Experimental Results

The approach performance has been tested using the CAVIAR database. Two targets are tracked simultaneously, despite their being articulated and deformable objects whose dynamics are highly non-linear. One of them performs a rotation in depth and heads towards the second one, eventually occluding it. The background colour constitutes a strong source of clutter. Furthermore, the illuminant depends on both position and orientation. Significant speed, size, shape and appearance changes can be observed, jointly with events such as grouping or splitting, and occlusions.

Detection results are shown in Fig. 2, tracking ones in Fig. 3, and the low-tracking evolution in Fig 4.(a). At frame 6, an agent enters the scene, motion is



(a)

(b) (c)

Fig. 4. (a) Tracks confirmation and removing. (b) Feature pool: the best $M = 3$ features at time t, and the best $N = 3$ long-run features are kept for appearance modelling. (c) Colour lines represent the intra-target similarity. Squares give the inter-tracker d_B.

detected, and a Kalman filter is created. A major size change occurs at frame 9: the target is completely inside the scene. Thus, a new Kalman filter is implemented, but both trackers are kept while their tracks have enough confidence, see Fig. 3.(b), and MSD value in Fig 4.(a). At frame 11, the first low-level tracker is removed. At frame 14, the track become stable, and a high-level tracker is instantiated. It is marked as a new born, entering the scene, and its appearance models as being updated. At frame 108, the segmentation of target 2 partially fails due to local illuminant changes, which leads to stop model updating. Grouping is detected at frame 110. At frame 122, a high-level tracker following the group is created. When the group splits, trackers are correctly re-associated.

The evolution of the feature pool for target 1 is shown in Fig 4.(b). Several facts can be noticed: some features are periodically among the best ones (features 13, 24 and 39); this repetitive behaviour is presumably due to the agent orientation and gait. Some features join the pool and quickly become one of the best ones as the agent moves and the background changes. Finally, other features are dropped and re-selected several times. These behaviours suggest that keeping a stable set of features may be useful for tracker association after tracking failure.

The Bhattacharyya distance between each new target detection and the smoothed model of feature 20 is represented in Fig 4.(c). When this feature is not selected or target cannot be detected, the distance is set to one. The inter-target d_B is also represented by two-colour squares, denoting both targets involved. At frame 91, the distance between the model of tracker 5 the one of

tracker 10 (the group) and 12 (the same target after the grouping) is computed. At frame 115 the same is done for tracker 3. The distance between tracker 1 and trackers 5 and 12 is almost double than the distance between the tracker 5 and 12, and in the same range of the intra-target distance computed during the successive detections. Thus, the association can be successfully carried out.

7 Conclusions

In this work a principle and structured system is presented in an attempt to take a step towards solving the numerous difficulties which appear in unconstrained tracking applications. It take advantages of both bottom-up and top-down approaches. A robust and accurate tracking is achieved in a non-friendly environment with several non-white light sources, high appearance and shape target variability, and grouping, occlusion and splitting. Both targets are successfully tracked despite no a-priori knowledge is used. The system adapts itself depending on the number of targets, the best local features, or which events are taking place. Future research will be focused on developing a method to perform target localisation within a group region, once the best features for disambiguating targets from background are already computed and smoothed.

Acknowledgements. This work has been supported by the Research Department of the Catalan Government, the EC grant IST-027110 for the HERMES project, and by the Spanish MEC under projects TIC2003-08865 and DPI-2004-5414. J. Gonzàlez also acknowledges the support of a Juan de la Cierva Postdoctoral fellowship from the Spanish MEC.

References

1. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A Tutorial on PFs for On-line Non-linear/Non-Gaussian Bayesian Tracking. *SP*, 50(2):174-188, 2002.
2. Y. Bar-Shalom and T. Fortran. *Tracking and Data Association.* A. Press, 1988.
3. R. Collins, Y. Liu, and M. Leordeanu. Online Selection of Discriminative Tracking Features. *PAMI*, 27(10):1631-1643, 2005.
4. D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based Object Tracking. *PAMI*, 25(5):564-577, 2003.
5. J. Deutscher and I. Reid. Articulated Body Motion Capture by Stochastic Search. *IJCV*, 61(2):185-205, 2005.
6. J. Gonzàlez. *Human Sequence Evaluation: The Key-frame Approach.* PhD thesis, UAB, Spain, 2004.
7. T. Horprasert, D. Harwood, and L. Davis. A Robust Background Subtraction and Shadow Detection. In *4th ACCV, Taipei, Taiwan,* volume 1, pages 983-988, 2000.
8. M. Isard and J. MacCormick. BraMBLe: A Bayesian Multiple-Blob Tracker. In *8th ICCV, Vancouver, Canada,* volume 2, pages 34-41. IEEE, 2001.
9. J. MacCormick and A. Blake. A Probabilistic Exclusion Principle for Tracking Multiple Objects. *IJCV,* 39(1):57-71, 2000.
10. K. Nummiaro, E. Koller-Meier, and L. Van Gool. An Adaptive Color-Based Particle Filter. *IVC,* 21(1):99-110, 2003.