Detection and Tracking of Multiple Agents in Unconstrained Environments

D. Rowe*, I. Huerta*, J. Gonzàlez+, Juan J. Villanueva*

* *Computer Vision Center & Dept. d'Informàtica, Edifici O, Campus UAB, 08193 Bellaterra, Spain*
E-mail:drowe@cvc.uab.es

+ *Institut de Robòtica i Informàtica Industrial (UPC – CSIC), Llorens i Artigas 4-6, 08028, 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 modular and hierarchical architecture, and tracking enhancements by on-line building and updating multiple appearance models. Successful experimental results are accomplished on complex real sequences.

Keywords: Motion Analysis; Multiple-target tracking; Feature evaluation and selection.

1 Introduction

Multiple human-beings tracking has become an active research field. This interest is motivated by an increasing number of potential applications. However, 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. Finally, agents interact among themselves, grouping or 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. 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 ones; finally, section 6 shows some experimental results.

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 target association, while the latter is based on complex shape and motion modelling. 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. 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).

High-level approaches rely on accurate target modelling. 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, although this may be inappropriate in crowded scenarios with multiple target-occlusions. BraMBLe [3] 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. [4] use a PF based on colour-histogram cues. However, no multiple-target tracking is considered, and it lacks from an indepen-

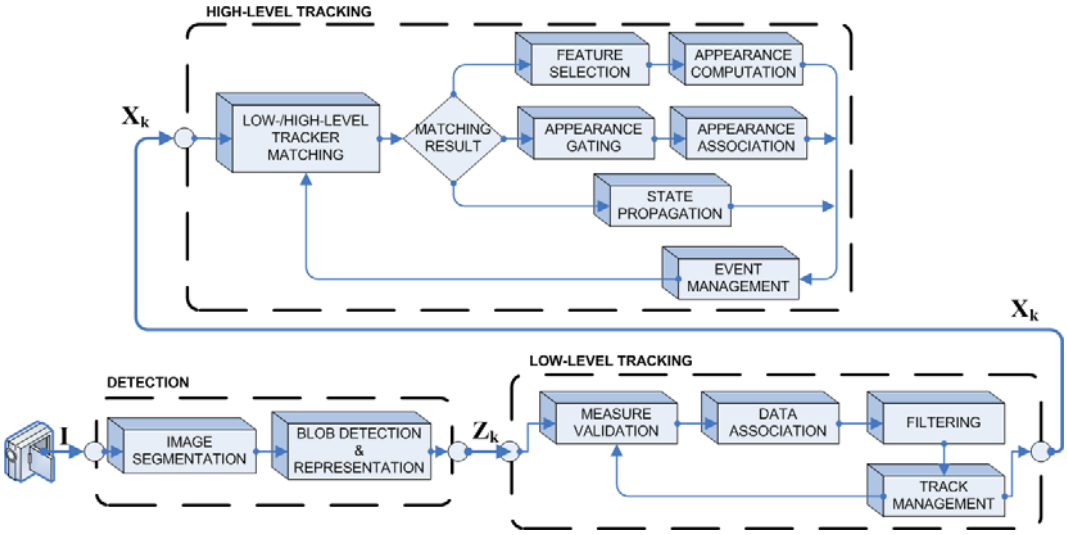


Figure 1: System architecture

dent observation process, since samples are evaluated according to the predicted image region histograms.

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. However, multiple-people tracking in complex environments require high-level reasoning. The lower level performs target detection, that is, pixel segmentation task, and object representations. Low-level tracking sets correspondences between observations and trackers, and perform state filtering. Tracks are finally managed. 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. 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. Image segmentation is performed following the method proposed by Horprasert et al. [2] which is based on a colour background-subtraction approach. Two distortion measures are established on brightness and chromacity. Pixels are classified into five categories: foreground, dark foreground (where no chromacity cues can be used), shadows, highlights, and background. Foreground blobs are subsequently detected, and an ellipse representation is computed.

4.1 Background model

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 pixel RGB-colour channel.

Two distortion measures are established: α, the brightness distortion, and CD, the chromacity distortion. Once each colour-channel value is normalised

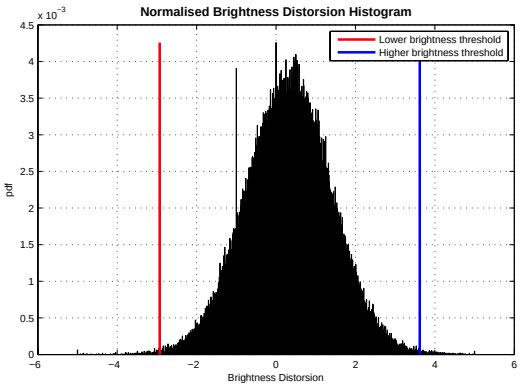


Figure 2: Threshold computation. Thresholds are automatically computed by cumulating histogram values and applying a detection rate.

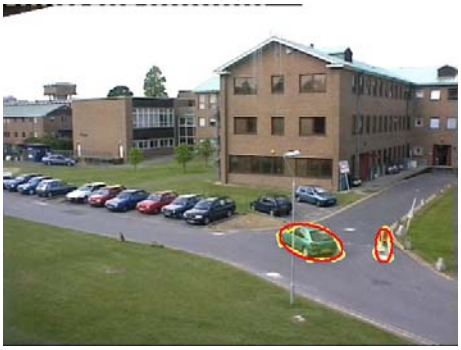
by their respective standard variation, the brightness distortion is computed by minimising the distance between the current pixel value and the chromacity line. The variation over time of both distortions for each pixel is subsequently computed by means of the Root Mean Square. These values are used as normalising factors so that a single threshold can be set for the whole image, see [2] for details.

Fig 2 shows the normalised brightness distortion histogram for a given frame, as well as the corresponding thresholds.

4.2 Image segmentation

Pixels are classified into five categories, depending on their chromacity and brightness distortion. For each frame, both normalised pixel distortions are computed. Those pixels whose chromacity distortion is higher than expected (that is, over the chromacity threshold) are marked as foreground. Those which are not, if the brightness distortion is more negative than the dark threshold, are marked as dark foreground. The rest are classified as highlight, if the brightness distortion is higher that the upper distortion threshold; or shadows, if the brightness distortion is lower than the lower distortion threshold. If none of these conditions hold, the pixel is classified as normal background. An example of foreground segmentation is show in Fig 3.(a).

(a)



(b)

Figure 3: Segmentation and detection examples. **(a)** The segmented foreground pixels are painted on white, while those ones classified as dark foreground are painted on yellow. Shadows are painted on green and highlights on red. **(b)** Detection example: red ellipses represent each target, and yellow lines denote their contour.

4.3 Blob detection

Once the current image has been segmented into the aforementioned five categories, blobs that may correspond to agents are detected. First, both foreground and dark-foreground maps are fused. Then, majority, opening and closing morphological operations are applied. Finally, a minimum-area filter is used. The surviving pixels are grouped into blobs. Each blob is labelled, their contours are extracted and an ellipse representation —which keeps the blob first and second moments— is computed. Thus, the *j*-observed blob at time *t* is given by the vector $\mathbf{z}_j^t = \left(x_j^t, y_j^t, h_j^t, w_j^t, \theta_j^t\right)$, where x_j^t, y_j^t represent the ellipse centroid, h_j^t, w_j^t are the major and minor axes, respectively, and the θ_j^t gives the angle between the abscissa axis and the ellipse major one. Fig 3.(b) shows an example of target detection.

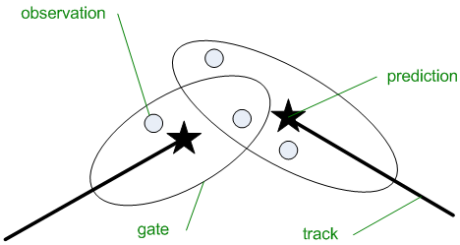


Figure 4: Observation association.

4.4 Low-level blob tracker

The target state is then estimated by filtering the sequence of noisy measures. Since their long-run dynamics are hardly predictable, a first-order dynamic model, where the acceleration is modelled as WAGN, is adopted. This assumption holds in most HSE applications. In a multiple-target tracking scenario, numerous observations may be obtained at every sampling period. 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. Measures are associated to the nearest tracker in whose gate they lie. 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 $\mathbf{S}_k$ and the observation MSD.

4.5 Data Association and Filtering

Measures are associated to the nearest neighbour tracker in whose gate they lie, see Fig 4. A more complicated data association method, such as PDAF or JPDAF, is not considered to be necessary since observations are usually just within one target gate. This is intrinsic to the segmentation method: if two targets are so close in the observation space as to introduce ambiguity in the data association process, the segmentation module is likely to segment just one blob corresponding to the group formed by both targets. This issue is addressed at the event-management section.

A bank of Kalman filters is implemented to estimate the state of all targets detected within the scene. As a special case, if no observation is associated to a particular target, its state is estimated using a Kalman Gain equal to zero, i.e. it is just propagated according to the dynamic model.

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. 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. In this work, the appearance-modelling approach presented by Collins et al. [1] 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 caused the target loss is over.

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 segmentation failures. Then, the system tries to associate it to newborn ones, presumably created once the event is over. If there are no tracker candidates, or they are not similar enough, their state is propagated.

5.2 Feature Selection

The target appearance is represented using colour histograms. Features are selected from a set of independent linear combinations of RGB channels. The $i-$feature target histogram is given by $\mathbf{p}^i = \{p_k^i; k = 1 : K\}$, where K is the number of bins.

Then, log-likelihood ratios of each feature are computed as:

$$L^i(k) = \log \frac{\max(p_k^i, \epsilon)}{\max(q_k^i, \epsilon)}. \quad (1)$$

Features are evaluated according to the variance-ratio of the log-likelihood which maximises the inter-class variance, while minimising the intra-class variance. Thus, features can now be ranked.

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 appearance changes. A pool of $M + N$ features is kept. These are the best M features at time t, and the best N long-run features. Mean appearance histograms are recursively computed:

$$\mathbf{m}_t^i = \mathbf{m}_{t-1}^i + \frac{1}{n_i}\left(\mathbf{p}_t^i - \mathbf{m}_t^i\right). \quad (2)$$

Similarity between two histograms is computed using a metric d_B based on *Bhattacharyya coefficient* $\rho = \sum_{k=1}^{K}\sqrt{p_k q_k}$. Target similarity is decided according to this metric and its statistics:

$$\mu_t^i = \mu_{t-1}^i + \frac{1}{n-1}\left(d_{B,t}^i - \mu_t^i\right), \quad (3)$$

$$\sigma_t^2 = \frac{n-3}{n-2}\sigma_{t-1}^2 + (n-1)\left(\mu_t^i - \mu_{t-1}^i\right). \quad (4)$$

5.4 Appearance Association

Low-level trackers lost their track during long-duration segmentation failures. Once the target is re-detected, a new tracker is instantiated. When this track become stable, it is confirmed and a high-level tracker is created. The former tracker were propagated. A tracker association process is performed, and the system concludes that both trackers are in fact representing the same target.

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.

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 occlusions. Detection results are shown in Fig. 5, and tracking ones in Fig. 6.

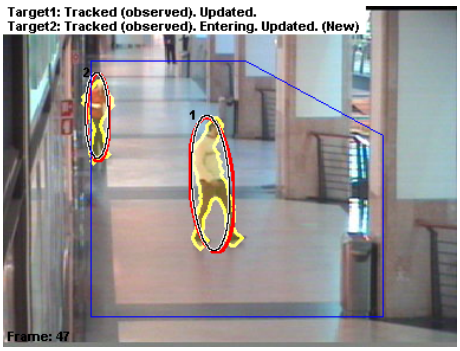
7 Conclusions

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.

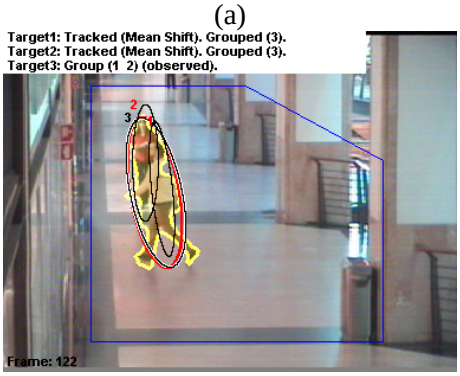
(a)

(b)

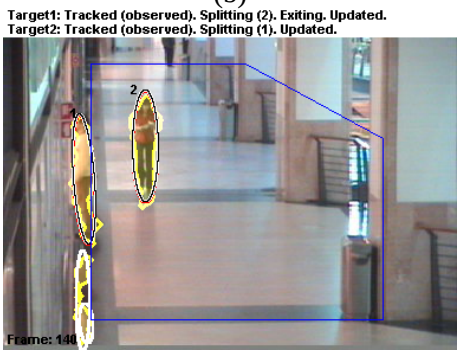
Figure 5: (a) Segmented frame. (b) Detected objects.



(a)



(b)



(c)

Figure 6: Tracking results: red ellipses denote detections, whereas white and black ones are low- and high-level tracker estimates, respectively.

Acknowledgements

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

References

[1] R. Collins, Y. Liu, and M. Leordeanu. Online Selection of Discriminative Tracking Features. *PAMI*, 27(10):1631–1643, 2005.

[2] 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.

[3] M. Isard and J. MacCormick. BraMBLe: A Bayesian Multiple-Blob Tracker. In *8th ICCV, Vancouver, Canada*, volume 2, pages 34–41. IEEE, 2001.

[4] K. Nummiaro, E. Koller-Meier, and L. Van Gool. An Adaptive Color-Based Particle Filter. *IVC*, 21(1):99–110, 2003.