**Machine Vision and Applications manuscript No.**
(will be inserted by the editor)

# On Tracking Inside Groups

**D. Rowe[1], J. Gonzàlez, M. Pedersoli, J.J. Villanueva**

Computer Vision Centre

Received: date / Revised version: date

**Abstract** This work develops a new architecture for multiple-target tracking in unconstrained dynamic scenes. It is conformed by a detection level which feeds a two-stage tracking system. A remarkable characteristic of the system is its ability to track several targets while they group and split, without using 3D information. Thus, a special attention is paid to the feature-selection and appearance-computation modules, and to those involved on tracking through groups. The system aims to work as a stand-alone application in complex and dynamic scenarios. No a-priori knowledge about either the scene or the targets, based on a previous training period, is used. Hence, the scenario is completely unknown beforehand. Successful tracking has been demonstrated in well-known databases of both indoor and outdoor scenarios. Accurate and robust localisations have been yielded during long-term target merging and occlusions.

## 1 Introduction

Multiple-target tracking is one of the most complex and active research fields in computer vision, specially when it concerns the analysis of unconstrained human-populated environments. People tracking is therefore highly appealing. This interest is also prompted by an increasing number of potential applications in which people tracking is a key task. Among these, we may find smart video safety and video surveillance [2, 7], intelligent gestural user-computer interfaces [10], or any other applications in orthopedics, sports, natural-language scene description, or computer-animation fields, related to Human-Sequence Evaluation (HSE) [5]. Further, trying to emulate the performances of natural-vision systems represents a real challenge.

In every situation where no a-priori knowledge is available about either the scene or the targets, motion-based tracking outperforms appearance-based tracking. This is specially the case when target appearances considerably evolve over time. An accurate initialisation is often not feasible, and the need of adaptation usually leads to the phenomenon known as model drift. However, there are many situations where segmentation-from-motion, and the subsequent observation-tracker correspondence, is not possible. Among those, one may include target grouping, or even worse, target occlusion, see Fig. 1.

In order to cope with these events, several approaches have been proposed in the related literature. The target state could be propagated according to the learned dynamic model, but this usually does not suffice,

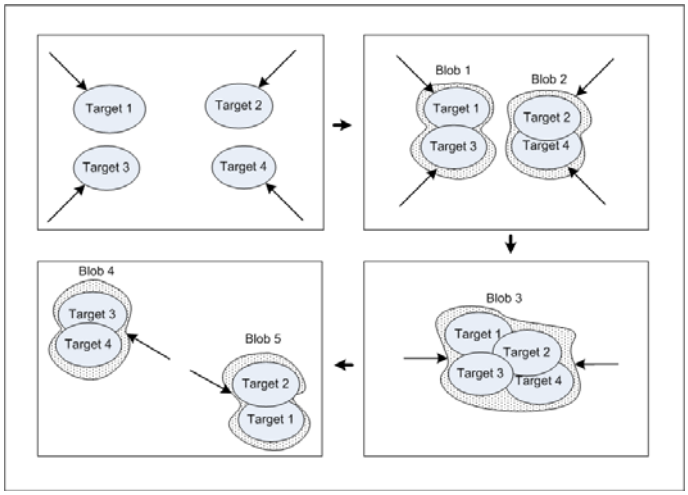



**Fig. 1** Target interaction. To keep the identity of multiple targets which cannot be independently segmented is a challenging task. Notice the different group membership of targets in blob 1 and 4.

since its motion is generally subject to sudden changes, and the possibility of losing the target increases with the time the target is non-detected. A coarse localisation can be obtained by considering that they are inside the group region [18]. However, grouped targets are not accurately tracked, and no complex situation can satisfactorily be tackled —for instance, those in which a group of more-than-two members split. Wu et al. [17] address occlusions events within a Particle Filter (PF) framework by implementing a Dynamic Bayesian Network (DBN) with an extra hidden process for occlusion handling. Alternatively, several approaches take advantage of 3D information by making use of a known camera model and assuming that the targets move on a known ground plane [19].

Therefore, targets within a group cannot be accurately tracked in 2D domains without using appearance information. However, appearance-based tracking may be extremely sensitive to illumination changes; and further, the background and nearby targets can act as appearance distracters, thereby causing the tracker to erroneously lock on them. Moreover, in open-world scenarios the target appearance cannot be specified in advance. It should also be continuously updated, since it strongly depends on the target position, its orientation to the camera and possibly several light sources, or the body posture in case of people tracking.

Several approaches have been taken in the literature to perform an appearance-based tracking. Particle Filters (PF), have been widely used to perform stochastic estimation [8,11,16]. Specifically, Nummiaro et al. [13] use a PF based on colour-histogram cues. However, no multiple-target tracking is considered, which implies that no scene event such as target grouping or occlusion can be analysed, and it lacks from an independent observation process, since samples are evaluated according to the histograms of the predicted image region. Perez et al. [14] propose also a PF based on a colour-histogram likelihood. They introduce interesting extensions in multiple-part modelling, incorporation of background information, and multiple-target tracking. Nevertheless, it may require an extremely large number of samples, since one sample contains information about the state of all targets, dramatically increasing the state dimensionality. In addition, no appearance

model updating is performed, what usually leads to target loss in dynamic scenes.

BraMBLe [9] is an appealing approach to multiple-blob tracking which models both background and foreground using Mixtures of Gaussians (MoG). Nevertheless, no model update is performed, there is a common foreground model for all targets, and suffers for the curse of dimensionality, as all PF-based methods which tackle multiple-target tracking combining information about all targets in every sample.

Target localisation following a gradient-descent search on a weighted image region has also been widely used in the visual tracking field [1,4,3]. In this case, the search is deterministic. This is usually done according to an histogram similarity measure related to the Bhattacharyya coefficient. However, these methods do not work in unconstrained situations. Thus, the approach presented by Comaniciu et al. [4] tracks just one target, initialised by hand, and the appearance model is never updated.

Recently, Collins et al. [3] presented an interesting tracker, based also on the mean-shift algorithm, 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, and it may suffer from model drift, although models are anchored to the first frame, which is manually segmented. Nevertheless, in both cases, just rigid targets are tracked (or rigid regions of them), appearance changes are limited, and since multiple-

target tracking is not considered, interaction events are not studied. These facts cannot be seen as minor issues in real applications.

McKenna et al. [12] combined colour and gradient information in their adaptive background subtraction approach. Tracking is done by means of data association. Three levels of representation are used, namely, regions, people and groups. People appearance is modelled using colour histograms. Visibility indexes, obtained from the probabilities that the pixels correspond to unoccluded people, are used to disambiguate occlusions. However, problems arise when several people and the background have a similar appearance. It is also assumed that the target appearance do not significantly change while the targets are grouped.

Consequently, tracking multiple targets simultaneously as they group and split remains a challenging task. The tracking success will be determined by the ability of distinguishing the target from potential distracters. In this proposal, motion segmentation is used on isolated targets, and the system takes advantage of these situations to build accurate target models. Then, a distracter-robust mean-shift method is proposed to track them inside groups. It copes with clutter distracters by selecting the most convenient colour-related features.

A set of appearance models is continuously conformed, smoothed and updated. Thus, multiple target representations are built using several models for each of them, while they are simultaneously being tracked. Further, colour information relative to the target surroundings, background and

other close targets is used to tune the appearance models. Potential model drift is precluded by carefully updating the appearance models, thereby ensuring proper tracking despite noisy measures, estimate errors, partial or complete occlusions, and changes in the illuminant and camera viewpoint. The proposed system aims to work as a stand-alone application in a non-friendly, complex and dynamic open scenario, which is completely unknown beforehand. Hence, no a-priori knowledge about either the scene or the targets, based on a previous training period, is used.

The remainder of this paper is organized as follows. Section 2 outlines the system architecture. A proposal which specifically addresses independent target tracking while they group and split is detailed in section 3. Section 4 shows some experimental results obtained from well-known databases, and finally, section 5 summarises the conclusions, and proposes some future-work lines.

## 2 Tracking Framework

Due to the inherent complexity involved in non-supervised multiple-human tracking, a structured framework is proposed to accomplish this task. This approach takes advantage of the modular and hierarchically-organised system that we have published in some preliminary works [6,15]. It is based on a set of co-operating modules distributed in three levels, which work following both bottom-up and top-down approaches. These levels are defined

according to the different functionalities to be performed, namely target detection, low-level tracking (LLT), and high-level tracking (HLT), see Fig. 2.

A remarkable characteristic of this architecture is that the tracking task is split into two levels: a lower level based on a short-term blob tracker, and a long-term high-level appearance tracker. The latter automatically builds and tunes multiple appearance models, manages the events in which the target is involved, and selects the most appropriate tracking approach according to these.

In any tracking system, reliable target segmentation is critical in order to achieve an accurate feature extraction without considering any prior knowledge about potential targets, specially in dynamic scenes. However, complex interacting agents who move through cluttered environments require high-level reasoning. Thus, our approach combines in a principled architecture both bottom-up and top-down approaches: the former provides the system with initialisation, error-recovering and simultaneous modelling and tracking capabilities, while the latter builds the models according to a high-level event interpretation, and allows the system to switch among different operation modes.

The first level performs target detection. The segmentation task is accomplished following a statistical colour background-subtraction approach. Next, the obtained image masks are filtered, and object blobs are extracted. Each blob is labelled, their contours are computed, and they are parametrically represented.

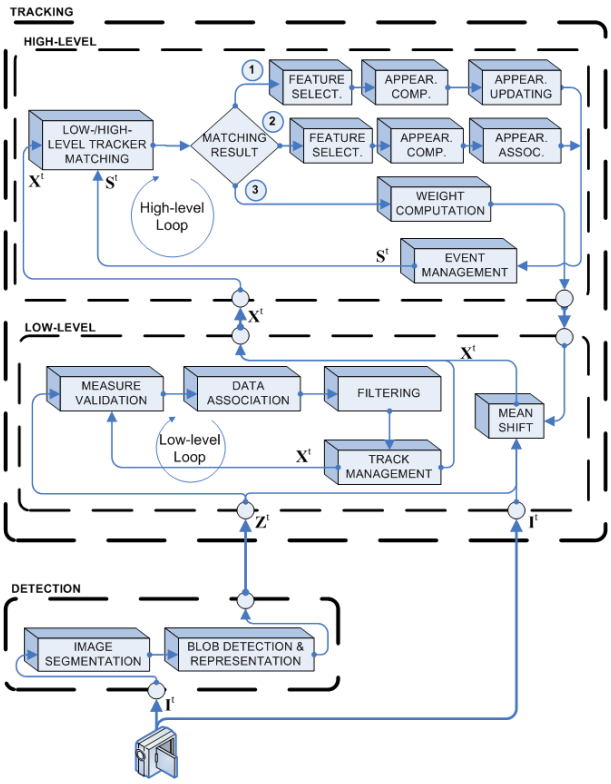



**Fig. 2** System architecture. $\mathbf{I}_t$ represents the current frame, $\mathbf{Z}_t$ represents the observation matrix, $\mathbf{X}_t$ the target low-level state matrix, and $\mathbf{S}_t$ the target's high level state matrix. Matching results are explained in the text. In this paper we mainly focus on the feature-selection and appearance-computation modules, and on those involved on tracking through groups, albeit the whole architecture is used in the presented experimental tests.

By using a parametric representation to model each blob, the spurious structural changes that they may undergo are constrained. These include target fragmentation due to camouflage, or the inclusion of shadows and reflections. Moreover, this representation can be handled by the LLT's,

thereby filtering the target state and reducing also these effects. Representations based on ellipses are commonly used [4,13]. Here an orientable ellipse is chosen —which keeps the blob first and second order moments. 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 $\theta_j^t$ gives the angle between the abscissa axis and the ellipse major one. Low-level motion trackers establish coherent target relations between frames by setting correspondences between observations and trackers, and 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 HLT's, and fed back to the measure-validation module. See [6] for details.

A HLT is associated to each confirmed LLT. Hence, tracking events are managed. This allows target tracking even when image segmentation is not feasible, and low-level motion trackers are removed, such as during long-duration occlusions or grouping events. As a result of the matching, three cases are considered: (i) if the track is stable, the target appearance is computed and updated, see matching result (1) in Fig. 2; (ii) those HLT's which remain orphans are processed to obtain an appearance-based data association, thereby establishing correspondences between lost HLT's and new ones, see matching result (2). The details of these procedures can be

found in [15]; and, (iii) those targets which have no correspondence are tracked in a top-down process using low-level appearance-based trackers, see matching result (3). An *event* module determines what is happening within the scene, such as whether target is grouping or a target is entering the scene. These results are fed back, thereby allowing low-level and HLT matching. In this paper we develop the feature-selection and appearance-computation modules, by paying special attention to grouping situations, and those modules involved on the third matching result, namely, weight computation and mean-shift tracking through groups.

## 3 Tracking Inside Groups

As it has been above stated, targets within a group cannot be accurately tracked without using appearance information, since motion cues are not sufficient. The task of the appearance-based trackers in the proposed system is to track those targets whose low-level motion tracker has been removed due to the lack of associated observations. Further, in order to increase the system performance, this is also done in every grouping or splitting situation, when the corresponding target has no observation associated — which means that a group is formed, but its tracker is not confirmed yet.

However, the target's appearance evolves in a unknown manner over time, and the local background and nearby targets may mimic its appearance. In order to achieve a successful tracking, this ambiguity must be minimised. Thus a feature space is selected by taking all distracters into

account, and model bins shared with distracters are made less significant. Further, potential drift of the appearance models is precluded by performing a careful updating according to the detected events and the evaluation of the tracking results.

### *3.1 Appearance representation*

The target appearance is represented by means of colour histograms. Histograms are broadly used to represented human appearance, since they are claimed to be less sensitive than colour templates to rotations in depth, the camera point of view, non-rigid targets, and partial occlusions [3,4,12,13]. They are also usually used to represent non-parametric distributions, provided that they allow one to achieve real-time performances given the low computational cost required. Thus, the histogram of a target is given by:

$$\mathbf{p} = \{p_k; k = 1 : K\}, \tag{1}$$

where $K$ is the number of histogram bins, and the discrete probabilities of each bin are calculated as:

$$p_k = C_N \sum_{a=1}^{M} g_E \left( \|\mathbf{x}_a\|^2 \right) \delta \left( b\left(\mathbf{x}_a\right) - k \right), \tag{2}$$

where $C_N$ is a normalisation constant required to ensure that $\sum_{k=1}^{K} p_k = 1$,

$\delta$ the Kronecker delta, $\{\mathbf{x}_a; a = 1 : M\}$ the pixel locations, $M$ the number of target pixels, and $b(\mathbf{x}_a)$ a function that associates the given pixel to its corresponding histogram bin. $g_E(x)$ is the convex and monotonic profile of an isotropic kernel which allows one to perform gradient-based searches, which need differentiable similarity functions. Further, by assigning lower weights to pixels farther from the centre, the influence of boundary clutter is diminished. Here, an elliptical Epanechnikov kernel has been used [4,13].

The above-defined appearance histograms are computed given a cropped image region. Two sources of information are available to decide which pixels should be considered as belonging to the target, namely the silhouette of the associated observation —given by the detection module— and the filtered ellipse —given by the LLT. In this work a conservative approach has been used in order to minimise the risk of failures caused by model drift. Thus, only those pixels which belong to both the detected silhouette and the filtered ellipse are considered. By doing so, background pixels which have been erroneously detected —e.g due to reflections— or those inside the tracked ellipse are removed. Also, non-reliable boundary foreground pixels, such as those of the end of the limbs, are usually not taken into account.

### *3.2 Feature selection*

Colour cues have been here selected to model the target appearance. Numerous colour spaces can be used, and each of them has tunable parameters, resulting in an enormous space of potential features. By selecting the most

appropriate one, a maximum discrimination between the target and local distracters is obtained. The following feature-selection technique has been used in [3] to maximise the target discrimination from its surroundings. There, features are selected from a set of linear combinations of the R, G and B channels:

$$\mathcal{F} = w_1 R + w_2 G + w_3 B \,|w_C \in \{-2, -1, 0, 1, 2\}\,, C = R, G, B \qquad (3)$$

which includes raw R, G, and B, intensity, and common chrominance approximations. The total number of candidates is $5^3$. Non-independent combinations are removed, leaving a set of 49 features. Computed values are then normalised to the range $[0 : 255]$, and subsequently discretised. In the present implementation the number of bins is set to $K = 64$. This is a sensitive decision since a low number of bins will prevent from target-clutter disambiguation, but, on the other hand, a high value favours erroneous representations that appear when distributions are estimated from an insufficient number of samples, and makes the representation too sensitive to minor illumination changes. The $i$-th target histogram is given by $\mathbf{p}^i$, while $\mathbf{q}^i$ represents the background histogram, computed from the background model. Features are then ranked in the following way. First, the log-likelihood ratio of each feature is computed:

$$L_k^i = \log \frac{\max\left(p_k^i, \epsilon\right)}{\max\left(q_k^i, \epsilon\right)}, \qquad (4)$$

where $\epsilon$ is set to prevent dividing by zero or taking the logarithm of zero. 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. The variance of the log-likelihood according to a general discrete distribution $\phi$ can be computed as:

$$\mathrm{var}\left(\mathbf{L};\phi\right) = E\left[\mathbf{L}^2\right] - E\left[\mathbf{L}\right]^2 = \sum_k \phi_k {L^2}_k - \left(\sum_k \phi_k L_k\right)^2. \tag{5}$$

The variance ratio of the log-likelihood for each feature $VR^i$ is defined as:

$$VR^i\left(\mathbf{L}^i;\mathbf{p}^i,\mathbf{q}^i\right) = \frac{\mathrm{var}\left(\mathbf{L}^i;\left(\mathbf{p}^i+\mathbf{q}^i\right)/2\right)}{\mathrm{var}\left(\mathbf{L}^i;\mathbf{p}^i\right)+\mathrm{var}\left(\mathbf{L}^i;\mathbf{q}^i\right)}, \tag{6}$$

and then, features are ranked according to these values: the higher, the better. Thus, the selection maximises the inter-class variance —that is, the distance between background and target clusters of bins— while minimising the intra-class variance —tightly clustering both background and target bins.

This feature-selection scheme is enhanced in this work by solving the initialisation, smoothing the representation, and completing it so that tracker association is feasible once the event that cause the target loss is over. The possibility of an inconsistent localisation due to feature switch is also min-

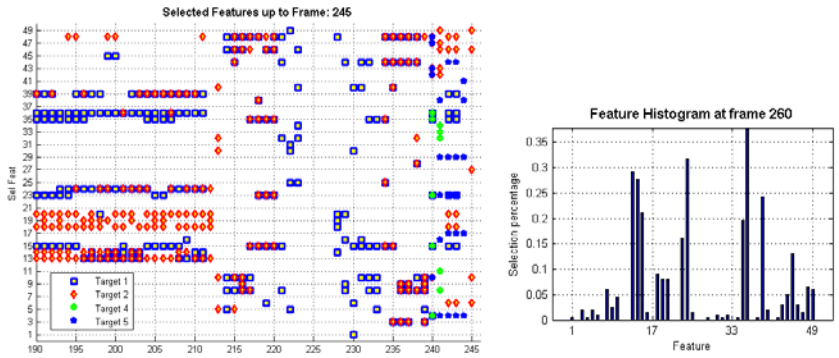
imised, by introducing the distinction between long-run features and the current best ones (as explained next). Further, this scheme is generalised in order to take into account distracters caused by nearby targets.

*3.2.1 Feature Selection in a n-Class Problem* Features are here selected considering not only the best distinction between the local background and the target, but also between the target and its group partners. This is done by generalising the variance-ratio expression given in Eq. (6) to a $n$-class problem:

$$VR^i\left(\mathbf{L};\mathbf{p}^{i,1},...,\mathbf{p}^{i,j},...,\mathbf{p}^{i,J},\mathbf{q}^i\right)=\frac{\text{var}\left(\mathbf{L}^i;\frac{1}{J+1}\left(\mathbf{q}^i+\sum_{j=1}^{J}\mathbf{p}^{i,j}\right)\right)}{\text{var}\left(\mathbf{L}^i;\mathbf{q}^i\right)+\sum_{j=1}^{J}\text{var}\left(\mathbf{L}^i;\mathbf{p}^{i,j}\right)}, \quad (7)$$

being $n=J+1$, where $J$ gives the number of partners including the target, whose histograms are given by $\mathbf{p}^{i,j}$, where $i$ denotes the feature index, and $j$ the target one; and $\mathbf{q}^i$ models the $ith-$background colour distribution. Thus, in order to allow the system to build reliable appearance models using the features which best distinguish a target from its potential distracters, once a grouping event is detected, the partners histograms are also used in the feature selection procedure.

*3.2.2 Model pool* Contrary to the paper of Collins [3], long-run features are here kept and smoothed. As it will be shown, these features are useful later on, after a target loss caused by an occlusion. Further, by smoothing




(a) (b)

**Fig. 3** Selected features. (a) Example of selected features on a given interval where targets are grouped. (b) Histogram of selected features on the whole sequence.

the histograms, the representation is less sensitive to potential initialisation and subsequent localisation errors, and to sudden and temporal appearance changes, for instance due to illumination fluctuations.

Hence, a pool of $M + N$ features is kept: the best $M$ features at time $t$, and the best $N$ long-run features, i.e. those which have been at the top of the feature ranking more times. These features are only dropped when new features enter the pool, and eventually overcome the formers.

By analysing the evolution of the model pool —see Fig. 3— several facts can be noticed: some features are periodically among the best ones; this repetitive behaviour is presumably due to similar agent orientations and gait during tracking. Some features join the pool and quickly become one of the best ones, as the agent moves and the local background changes. Finally, other features are dropped and re-selected several times: they are

periodically among the best ones, but they are not selected enough times, and due to the pool size are dropped when others join the set. These behaviours strongly suggest that keeping a stable set of features may be useful for tracker association after a tracking failure.

Whenever there is enough confidence on the tracker to update the appearance, all $M + N$ models are updated. This is done in a recursive way using an adaptive filter:

$$\overline{\mathbf{p}}_t^{i,j} = \overline{\mathbf{p}}_{t-1}^{i,j} + \alpha_{\mathbf{p}} \left( \mathbf{p}_t^{i,j} - \overline{\mathbf{p}}_{t-1}^{i,j} \right), \tag{8}$$

where $\alpha_{\mathbf{p}} \in [0, 1]$ denote the adaptation rate which weights the most recent values versus the historic one, and $\overline{\mathbf{p}}_t^{i,j}$ the smoothed histogram of target $j$ at time $t$ using the $i$-th feature. Old values are exponentially forgotten according to this rate: the bigger it be, the faster old data are forgotten.

However, contrary requirements must be fulfilled: (i) when a feature is recently added, the model should be fast adapted, in order to cope with potential detection errors during the initialisation; (ii) medium-term models should not be excessively adapted, to prevent model-drift phenomena; (iii) long-term models should be adaptive enough, so that the system can handle unexpected appearance changes. This suggest defining the adaptation rate in terms of time, and to employ a principled function $\alpha_{\mathbf{p}}(t)$. Thus, a recursive mean filter is first used, thereby fulfilling the two first requirements, but the adaptation rate is fixed to a high enough value after an initialisation pe-

riod, and thus model adaptation could be performed during arbitrary long time periods.

In this way, once a target is detected, and the corresponding low-level motion-based tracker is confirmed, the target is being tracked while it is simultaneously being modelled by the HLT. New features can be added, while stable ones build robust appearance models, even during hard situations, as it will demonstrated later on.

A similarity measure between two histograms can be computed using the following metric [4,13]:

$$d_B = \sqrt{1 - \rho\left(\mathbf{p}, \mathbf{q}\right)}, \tag{9}$$

where $\rho$ is the Bhattacharyya coefficient*:*

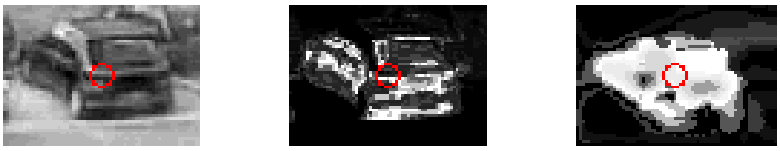
$$\rho\left(\mathbf{p}, \mathbf{q}\right) = \sum_{k=1}^{K} \sqrt{p_k q_k}. \tag{10}$$

A similarity criterion must be set in order to establish when two histogram are close enough. For this purpose, every time the smoothed histogram is updated, the mean and variance of Bhattacharyya metric $d_B$ between the former histogram and the new one, are recursively updated:

$$\mu_t^{i,j} = \mu_{t-1}^{i,j} + \frac{1}{n^{i,j} - 1}\left(d_{B,t}^{i,j} - \mu_{t-1}^{i,j}\right), \tag{11}$$

$$\left(\sigma_t^{i,j}\right)^2 = \frac{n^{i,j} - 3}{n^{i,j} - 2}\left(\sigma_{t-1}^{i,j}\right)^2 + \left(n^{i,j} - 1\right)\left(\mu_t^{i,j} - \mu_{t-1}^{i,j}\right)^2, \tag{12}$$

where $n^{i,j}$ is the number of times this particular feature histogram has been



(a) (b) (c)

**Fig. 4** Weighted image. (a) Image mapped according to the selected feature. (b) Corresponding weighted image. (c) Weighted image for a feature with higher Variance Ratio.

updated. In this way, the metric distribution can be parametrised and used to establish a confidence measure.

### *3.3 Mean-shift Tracking*

This technique achieves target localisation by performing a gradient-descent search on a image region of interest, which is previously weighted. The target model is given by the histogram $\overline{\mathbf{p}}$, while the target candidate distribution at the image location $\widehat{\mathbf{x}}_0$ is represented by $\hat{\mathbf{p}}(\widehat{\mathbf{x}}_0)$. The similarity between two histograms is computed using the metric defined in Eq. (9).

The mean-shift procedure recursively moves the candidate position to a new location, while searching the local maximum according to the aforementioned metric [4]. That is to say, a new location is searched in the neighbourhood of the former one by maximising the similarity between the target model and the candidate one, computed from the current image at this location, see Fig. 4. This is approximately equivalent to minimise the second term of the Taylor expansion of the Bhattacharyya coefficient which

represents a weighted-data density estimate computed with the kernel profile [4]. Thus, the new location is given by:

$$\hat{\mathbf{x}}_1 = \frac{\sum_{a=1}^{M} \mathbf{x}_a w_a \dot{g_E}\left(\left\|\frac{\hat{\mathbf{x}}_0 - \mathbf{x}_a}{}\right\|^2\right)}{\sum_{a=1}^{M} w_a \dot{g_E}\left(\left\|\frac{\hat{\mathbf{x}}_0 - \mathbf{x}_a}{}\right\|^2\right)}, \tag{13}$$

where the weights $w_a$ are given by:

$$w_a = \sum_{k=1}^{K} \sqrt{\frac{\overline{p}_k}{\hat{p}_k\left(\hat{\mathbf{x}}_0\right)}} \delta\left(b\left(\mathbf{x}_a\right) - k\right). \tag{14}$$

By choosing an Epanechnikov kernel, both kernel profile derivatives $\dot{g}_E$ in Eq. (13) can be removed by taking into account that the derivative of the profile of an Epanechnikov kernel is a constant. Thus, the algorithm works as follows:

1. the histogram of the target candidate $\hat{\mathbf{p}}$ is computed at location $\hat{\mathbf{x}}_0$,
2. weights are computed according to Eq. (14),
3. the next target location $\hat{\mathbf{x}}_1$ is derived following Eq. (13),
4. if $\|\hat{\mathbf{x}}_0 - \hat{\mathbf{x}}_1\| < \epsilon$, or the maximum number of iterations has been reached, stop. Otherwise set $\hat{\mathbf{x}}_0 \leftarrow \hat{\mathbf{x}}_1$ and go to step 1.

This procedure is here enhanced by making a principled use of all available sources of information in order to minimise the risk of target loss when no accurate motion-segmentation can be performed, see Fig. 5. These include an updated background model, the current frame segmentation, the

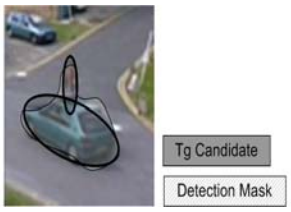



**Fig. 5** Sketch of merging targets. Foreground candidate regions include pixels of the background and of nearby targets. The detection mask include shadows and reflections, but do not enclose all target pixels due to camouflage problems.

estimate state of all targets within the scene, their multiple-appearance models, and the prediction of collisions according to the learned dynamic models.

*3.3.1 Introducing Motion Cues* The current segmentation can be used to weight the influence of each pixel on the candidate histogram, and on the weighted sub-image where the search is performed. By doing so, the system is making use of the results obtained by the detection level, but without neglecting the possibility of segmentation errors.

A mean-shift procedure assigns weights to each histogram bin according to a relation between the model and candidate histograms, and then back-projects these values into the image, before computing the new proposed localisation. Thus, each pixel is weighted according to its supposed membership to a determine target, given its appearance model.

By using motion information, those pixels within either the target candidate region or the search region which are not segmented are weighting according to an estimate error-segmentation rate, see Fig. 6. In addition,

(a) (b) (c)

**Fig. 6** Segmentation weighting. (a) Cropped candidate. (b) Search region mask. (c) Candidate region mask (previous to apply the Epanechnikov kernel).

candidate pixels contribute to the histogram with a value in the interval $[0:1]$, according to the applied kernel. In this proposal, the Epanechnikov kernel is used combined with the resulting detection mask, thereby minimising the risk of over-weighting significant distracters bins.

*3.3.2 Bin-weighting* So far, background and partners' information has been used to select the features that best discriminate the target from a local environment. However, even for the best features, histogram bins could be shared between the target and potential distracters. This fact leads to an erroneous localisation, which finally ends causing the drift of the appearance models. This can also be accelerated due to the fact that the foreground is hardly ever perfectly delineated. To minimise tracking failures due to this issue, the following approach is proposed.

The background-weighting approach proposed in [4] is here generalised by including three sources of information: the appearance models of the partners, the learned background model, and the current surroundings. Further, this is applied to each appearance model computed from a particular feature $f_i$. The learned background presents the advantage that it contains

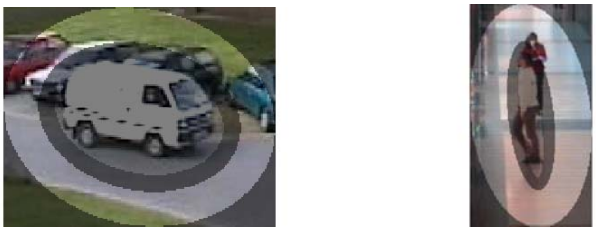


**Fig. 7** Examples of centre-surround model with safety margin. Regions from centre to border: target estimation, safety margin, surrounding background, and non-local background.

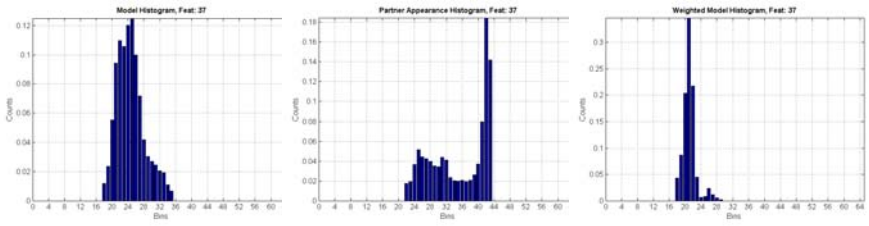
no foreground information, but it may differ from the current one due to the occlusion of some light source. A centre-surround approach which also includes a safety region between them, is used to cope with these effects, see Fig. 7. It also includes information about nearby targets, at the cost of potential incorporation of target pixels, specially if the target shape cannot be fairly represented by an ellipse. This is minimised by the inclusion of the safety margin. A conservative approach has again been chosen: all significant bins in any of the aforementioned sources of knowledge of potential distracters will have its importance diminished.

The surroundings histogram is computed using the outer region which encloses the target:

$$\mathbf{s}^i = \left\{ s_k^i ; k = 1 : K \right\}. \tag{15}$$

Hence, this histogram is used to compute a weight for each bin, depending on its significance:

$$\omega_k^{i,s} = \left\{ \min \left( \frac{s_k^{i*}}{s_k^i} \right) ; k = 1 : K \right\}, \tag{16}$$




(a) (b) (c)

**Fig. 8** Involved histograms. (a) Model histogram. (b) Partner histogram. (c) Weighted model histogram

where $s_k^{i*}$ is the minimum non-zero value. The same technique is used to compute weights according to the learned background model, $\omega_k^{i,q}$, and each partner, $\omega_k^{i,j}$. Thus, for the $l$-target among the $J$ targets of the group, the total weight of each model bin, given the $i$-th model feature, is obtained combining these weights:

$$\omega_k^{i,l} = \omega_k^{i,s}\omega_k^{i,q} \prod_{j=1, j\neq l}^{J} \omega_k^{i,j}. \tag{17}$$

These weights can then be applied to the target model to diminish the importance of those bins which are shared with potential distracters, see Fig. 8. Bin weighting according to the appearance of local distracters can be seen like a probabilistic exclusion principle. Such a technique has also been used in [11] in order to avoid that an edge feature can correspond to several targets. In other words, one particular evidence must not contribute to mutually exclusive hypotheses. In our case, shared model bins

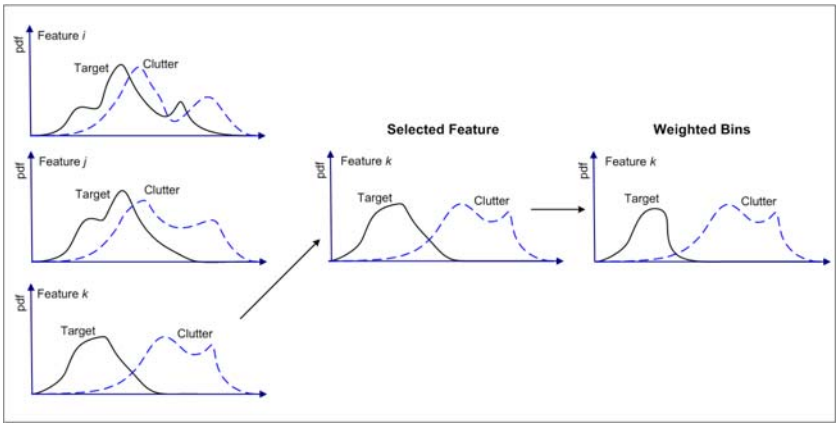



**Fig. 9** Disambiguating target and clutter appearances.

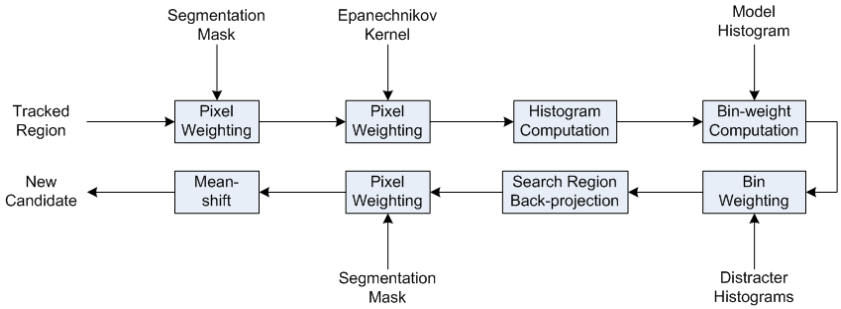



**Fig. 10** Combined motion and appearance weighing approaches.

must not reinforce the same local maximum in the weighted image where the mean-shift is computed, see Fig. 9. The complete approach which combines both motion and appearance cues to perform target localisation is shown in Fig. 10.

### *3.4 Criteria to Perform an Appearance Updating*

A bank of $M + N$ mean-shift procedures is run, and each of them uses an appearance model tuned at one of the selected features. These models need

to be updated even when the targets are grouped, since their appearances are always subject to undergo significant changes, specially when the targets are in motion. However, the updating of the appearance models must be carefully done in order to avoid model drift. Therefore, the results of the different mean-shift runs are only fused once they have been properly filtered according to appearance and localisation criteria.

First, those mean-shift procedures which have not converged after a number of iterations are not considered reliable enough to take them into account to perform the updating. Next, an appearance gate is computed to filter those features whose histograms significantly differ from the models according to the learned feature statistics based on the Bhattacharyya coefficient:

$$d_{B,t}^{i,j} < \mu_t^{i,j} + c\sigma_t^{i,j}, \tag{18}$$

where $c$ is the factor which set the confidence region. Finally, a robust target localisation is obtained by filtering potential position outliers among the remaining features. This avoids that a feature model locked on a distracter similar in appearance corrupts the localisation computation. This is done by computing the position mean and variance, and removing the outliers. The procedure is iterated until convergence.

When at least one model survives the appearance filtering, the localisation is considered reliable enough to perform a model updating and a new

feature selection. Given that a candidate localisation is always necessary, in the event of non having any reliable result according to the appearance criterion, the above robust-target localisation is performed, and the position is updated, but the appearance models are kept unmodified.

## 4 Experimental Results

The performance of the system has first been tested using sequences taken from two well-known databases —the CAVIAR database[1] and PETS 2001 Test Case Scenario[2]— and own sequences from the projects in which the authors are involved. CAVIAR sequence corresponds to indoor sequences which have been recorded in a mall centre; PETS one contains outdoor sequences taken in a scene which includes roads, parking places, and green areas surrounding several buildings; a crosswalk scene is analysed in *CVC_Zebra1*; finally, *Hermes_Outdoor_Cam1* sequence presents a great diversity of situations, where three people and three cars act on a robbery sequence, while suitcases and bags are carried, left and picked from the floor.

In the sequence *OneLeaveShopReenter1cor* (CAVIAR database, 389 frames at 25 fps, 384 x 288 pixels), two targets are tracked simultaneously, despite their being articulated and deformable objects whose dynamics are highly non-linear, and that move through an environment which locally mimics the target colour appearance. The first target performs a rotation and heads towards the second one, eventually occluding it. The background

[1] http://homepages.inf.ed.ac.uk/rbf/CAVIAR

[2] http://peipa.essex.ac.uk/ipa/pix/pets

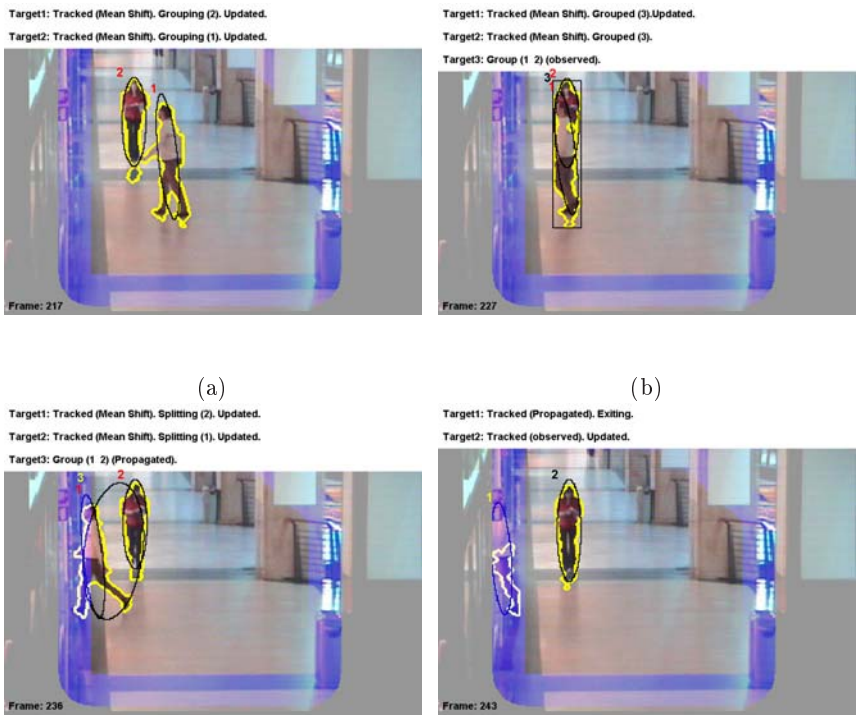



**Fig. 11** Sample tracking results on a indoor sequence. Targets are accurately tracked despite strong occlusion of target 2, the fact that the floor and the torso of target 1 have the same colour distribution, and that this distribution evolves from reddish to bluish hues as the motion is performed.

colour distribution is so similar to target one that it constitutes a strong source of clutter. Furthermore, several oriented lighting sources are present, dramatically affecting the target appearance depending on its position and orientation (notice the bluish effect on the floor on the right of the corridor, and the reddish one on the floor of the left of the corridor). Thus, significant

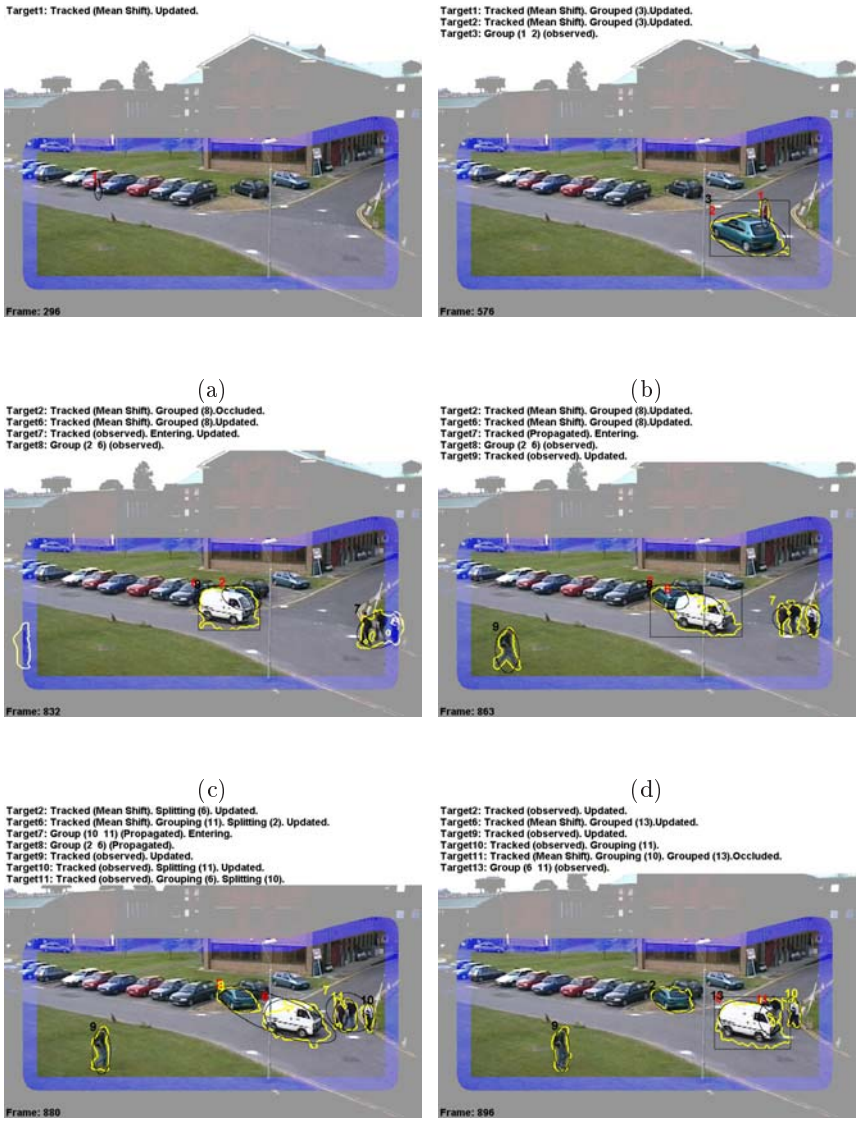
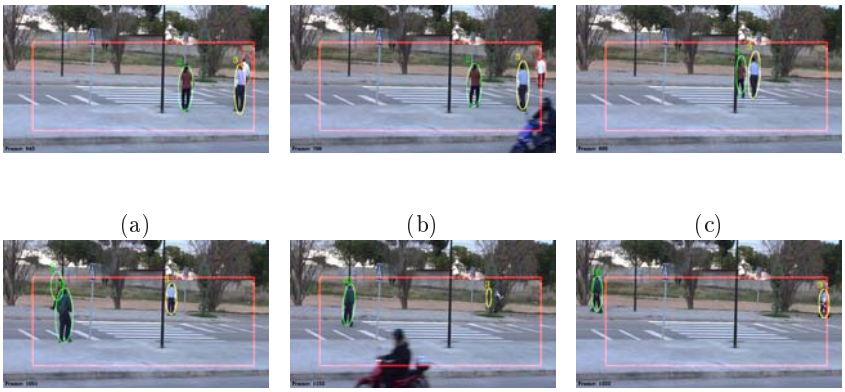



**Fig. 12** Sample tracking results on a PETS sequence. An accurate localisation of tg 1 is obtained despite no segmentation is available (a); robust performance under a heavy occlusion are shown in (c),(d); tg 6 is accurately tracked despite it is grouped, and it cannot be segmented during most of the sequence.



(a) (b) (c)

(d) (e) (f)

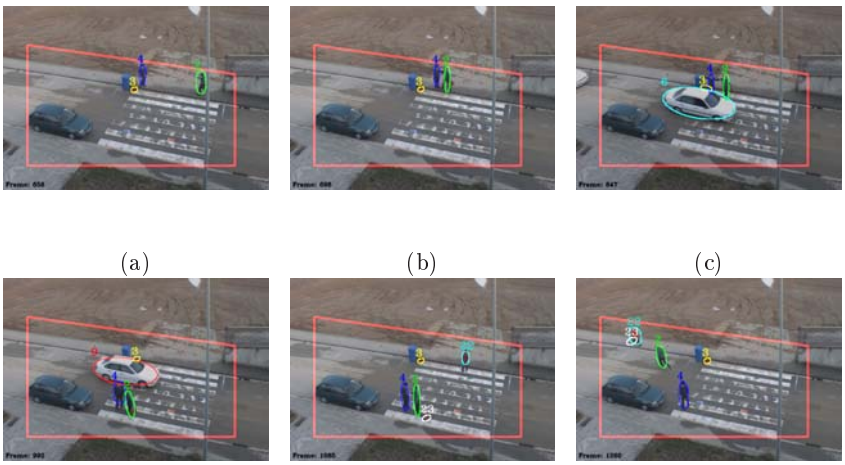
**Fig. 13** Sample tracking results on *CVC_ Zebra1* sequence. Targets are successfully tracked despite mutual occlusions in (a) and (d), or occlusions with the background in (c) and (e); interaction and scene events are correctly inferred.

speed, size, shape and appearance changes can be observed, jointly with events such as people grouping, partial occlusions and group splitting.

The sequence *DATASET1_ TESTING_ CAMERA1* (PETS database, 2688 frames at 29.97 fps, 768 x 576 pixels) presents a high variety of targets entering into the scene: three isolated people, two groups of people, three cars, and a person who exits from a parked car. These cause multiple tracking events in which several targets are involved in different grouping, grouped, and splitting situations simultaneously.

Successful results are obtained in the analysed sequences[3]. This field still being a *novel* open research line, there is unfortunately a lack of widely

[3] The reader is encouraged to see the whole processed sequences at http://iselab.cvc.uab.es/?q=agent_motion



(a) (b) (c)

(d) (e) (f)

**Fig. 14** Sample tracking results on *HERMES_Outdoor_Cam1* sequence. The dissolution of a non-detected group is correctly detected in (a), (e); targets are successfully tracked through groups in (b), partial occlusions in (c), and complete occlusions in (d); left objects are detected in (e), an correctly tracked after being picked up in (f).

accepted test data-bases, ground-truth data, and evaluation criteria. Performance evaluations are often based on quantitative metrics which depends on qualitative events, and results are usually evaluated by means of visual inspection. In order to allow algorithm comparisons, a standard methodology must be developed and assumed, test sequences should be synchronised and calibrated, and ground truth data must be available.

A ground-truth annotation tool has been developed, and the interaction between human and computer is aided by using a pen tablet. Thus, foreground regions can be annotated, visualised and edited. Targets are labelled, and visible and occluded regions are pointed out, as well as significant parts

as head or feet. As a result, a XML file is generated with the annotation data, and a set of target image masks are stored.

This data is used to measure the system performance. Events are correctly detected, albeit cannot ever be placed at a exact time instant, due to its inherent subjective component, see Table 1 relative to the sequence shown in Fig. 14. However, target 1 does not keep its ID after leaving the bag, due to major shape and appearance changes, and two new trackers are instantiated —target 3 and target 4. Hence, target 1 is referred as target 4 after bag —target 3— is left. Consequently, subsequent tracker instantiations have the labels shifted. For instance, group is referred as target 4 in the annotated events and target 5 in the computed ones. Nevertheless, this ID change is desirable in other cases, such as a man leaving a child.

Further, several trajectory indicators over the tracked targets are computed and presented in Table 2. Thus, every time a new blob is detected, a LLT is instantiated. This usually happens when targets merge into groups, they dissolve themselves, or targets undergo significant changes due to camouflage, occlusions, etc. Thus, the number of LLT's is much higher than the number of targets in every analysed sequence. When a LLT become stable, a HLT is created and associated with it. These are hopefully subsequently associated with the HLT that is already tracking the target. In this case, the target identity is not broken. When this process last more than one frame, the identity is temporarily broken. Since a HLT is created after the event is over, together with the fact that HLT are also instantiated to track groups,

**Table 1** Annotated and computed events on Hermes sequence. The attribute denote the targets involved.

| Annotation (t) | ID | Attrib. | Result (t) | ID | Attrib. |
|---|---|---|---|---|---|
| observed (550) | 1 | – | observed (550) | 1 | – |
| entering (629) | 2 | – | entering (629) | 2 | – |
| — | | | dissolving (655) | 1 | – |
| splitting (662) | 1 | 3 | splitting (655) | 4 | 3 |
| splitting (662) | 3 | 1 | splitting (655) | 3 | 4 |
| grouping (681) | 1 | 2 | grouping (682) | 4 | 2 |
| grouping (681) | 2 | 1 | grouping (682) | 2 | 4 |
| grouped (689) | 1 | 4 | grouped (697) | 4 | 5 |
| grouped (689) | 2 | 4 | grouped (697) | 2 | 5 |
| group (689) | 4 | 1&2 | group (697) | 5 | 2&4 |

the number of HLT's is higher than the actual number of targets, even if the identities are correctly kept.

Temporarily broken ID in CVC_Zebra sequence is due to an important partial occlusion of target 3 with a tree, see Fig. 13.(e). In the Hermes sequence this fact happens when the suitcase is picked up, due to significant segmentation errors. The permanent broken ID, and the false positives are due to *ghosts* yielded by a non-detected motionless car which starts motion. No target is ever undetected: the rate of false negative is zero in all the sequences analysed.

In order to to experimentally explore the effect of the different modules, several tests on the Hermes sequence have been carried out using the previ-

**Table 2** Trajectory measures.

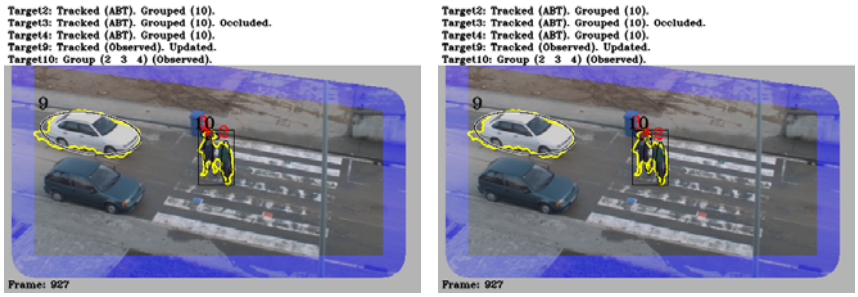
| Measure\Sequence | CAVIAR | PETS | CVC | HERMES |
|---|---|---|---|---|
| Targets | 2 | 8 | 4 | 8 |
| LLT | 8 | 78 | 138 | 86 |
| HLT (targets) | 4 | 28 | 11 | 36 |
| HLT (groups) | 1 | 13 | 3 | 11 |
| Temp. Broken ID | 0 | 0 | 1 | 2 |
| Perm. Broken ID | 0 | 0 | 0 | 2 |
| False Positives | 0 | 0 | 0 | 2 |
| False Negatives | 0 | 0 | 0 | 0 |

**Table 3** Effect of the different modules on tracking performance.

| Module\Measure | Temp. Broken ID | Perm. Broken ID | FP | FN |
|---|---|---|---|---|
| Normal operation | 2 | 2 | 2 | 0 |
| No use of bin weighting* | 2 | 2 | 2 | 0 |
| No ABT updating | 4 | 4 | 2 | 0 |
| No motion cues in ABT | 3 | 6 | 2 | 0 |
| Combined removal | 3 | 9 | 2 | 0 |

*The indicators do not show a worse performance since the redundancies provided by the different modules make the errors no catastrophic enough to cause a target loss. However, a poor target localisation is obtained, as shown in Fig. 15.

ous indicators. Thus, as shown in Table 3, the removal of any of this modules cause make the performance worse. Nevertheless, it should be remarked that these modules work in cooperation to maximise the target disambiguation




(a) (b)

**Fig. 15** Target localisation. Example of a poor target 4 localisation due to the fact of no using the bin-weighting module in (b) in comparison with (a).

from potential distracters. Therefore, since they provide some redundancy for the sake of robustness, the effect of removing only some of them may be not significantly noticeable.

Finally, it worth to say some remarks on real-time performances. Multiple-people tracking in unconstrained and dynamic scenarios are one of the most computationally-speaking demanding task in Computer Vision. Image-sequence capture and transfer is already feasible in real-time. Nevertheless, real-time performances may be achieved without excessively compromising accuracy and robustness by placing special care in three main tasks, namely, hardware implementation, code optimisation, and algorithm designing. The computational load at any time $t$ depends on particular issues which cannot be controlled, such as the number of targets within the scene, and the size of these targets or the scene itself in number of pixels. It also depends on design decisions which may be critical to achieve robust and accurate

performances —such histogram sizes, or the number of features selected in this the proposal.

Thus, the computational complexity will be given by the complexity of each of the algorithms run at each module. For instance, the cost of the mean-shift algorithm is given by [4]:

$$C_O \approx N_i \left( c_h + M c_s \right), \tag{19}$$

where $N_i$ is the mean number of iterations per frame an target, $c_h$ the cost of computing the candidate histogram, $M$ the number of target pixels, and $c_s$ the cost of an addition, a squared root, and a division.

Significant speed improvements can be achieved by processing pixel-wise operations in parallel. In addition, many systems can benefit from specific hardware implementations like FPGA, DSP, GPU, etc. Low-level languages which give more direct access to the underlying machine allow faster computation as the expense of less readability and maintainability. On the contrary, interpreted languages are executed from source form, and are consequently slower. However, the code is often more flexible, allowing a faster prototyping.

The current system is implemented as a *Matlab* prototype. The focus has been placed on achieving robust and accurate performance, instead of on a careful code optimisation. Subsequent implementations of bottleneck modules in C++ have yielded speed improvements which reduce 25 times

the computation time of these particular functions. This would allow the system to process the above sequences at an average rate around 10 fps in a *Pentium V* @ 3200Mhz.

## 5 Concluding Remarks

A structured framework is here used to combine in a principled way both bottom-up and top-down tracking approaches. It consists in modular and hierarchically-organised architecture able to track multiple targets simultaneously. In this paper we focus on tracking several targets independently while they are grouped, thereby yielding an accurate and robust target localisation. Thus, we develop the feature-selection and appearance-computation modules, by paying special attention to the special characteristics of grouping situations. Features are selected considering not only the best distinction between the local background and the target, but also between the target and its group partners.

A model pool is built, and long-run features are kept and smoothed. These features are useful after a target loss caused by an occlusion to recover the target. Further, by smoothing the histograms the representation is less sensitive to potential initialisation and subsequent localisation errors. Then, a second operation mode, consisting in an appearance-based tracking, is added to tackle grouping and occlusion events. Motion and appearance cues, relative to potential distracters, are taken into account when performing the

gradient search. A principled model updating scheme is followed to avoid model drift.

Experiments on complex indoor and outdoor scenarios have been successfully carried out, thereby demonstrating the system ability to deal with difficult situations in unconstrained and dynamic scenes. No a-priori knowledge about either the scene or the targets, based on a previous training period, is required.

Future work will focus on distinguishing among people, vehicles and other objects in motion. Next, body parts could be extracted and tracked. This can enhance agent tracking during long-term occlusions. Finally, initialisation should include a group segmentation method so that agents who enter the scene together could be segmented and independently tracked.

## References


1. R. Collins. Mean-shift Blob Tracking through Scale Space. In *CVPR, Madison, WI, USA*, volume 2, pages 234–240. IEEE, 2003. (Cited on page 5)
2. R. Collins, A. Lipton, and T. Kanade. A System for Video Surveillance and Monitoring. In *8th ITMRRS, Pittsburgh, USA*, pages 1–15. ANS, 1999. (Cited on page 2)
3. R. Collins, Y. Liu, and M. Leordeanu. Online Selection of Discriminative Tracking Features. *PAMI*, 27(10):1631–1643, 2005. (Cited on pages 5, 12, 14, and 16)

4. D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based Object Tracking. *PAMI*, 25(5):564–577, 2003. (Cited on pages 5, 10, 12, 13, 19, 20, 21, 23, and 37)
5. J. Gonzàlez. *Human Sequence Evaluation: The Key-frame Approach*. PhD thesis, UAB, Spain, 2004. (Cited on page 2)
6. J. Gonzàlez, D. Rowe, J. Andrade, and J.J. Villanueva. Efficient Management of Multiple Agent Tracking through Observation Handling. In *6th VIIP, Mallorca, Spain*, pages 585–590. IASTED/ACTA PRESS, 2006. (Cited on pages 7 and 10)
7. I. Haritaoglu, D. Harwood, and L. Davis. W4: real-time surveillance of people and their activities. *PAMI*, 22(8):809–830, 2000. (Cited on page 2)
8. M. Isard and A. Blake. Icondensation Unifying Low-level and High-level Tracking in a Stochastic Framework. In *5th ECCV, Freiburg, Germany*, volume 1, pages 893–908, 1998. (Cited on page 4)
9. M. Isard and J. MacCormick. BraMBLe: A Bayesian Multiple-Blob Tracker. In *8th ICCV, Vancouver, Canada*, volume 2, pages 34–41. IEEE, 2001. (Cited on page 5)
10. R. Kahn, M. Swain, P. Prokopowicz, and R. Firby. Gesture Recognition Using the Perseus Architecture. In *CVPR, San Francisco, USA*, pages 734–741. IEEE, 1996. (Cited on page 2)
11. J. MacCormick and A. Blake. A Probabilistic Exclusion Principle for Tracking Multiple Objects. *IJCV*, 39(1):57–71, 2000. (Cited on pages 4 and 25)
12. S. J. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. *CVIU*, 80(1):42–56, 2000. (Cited on pages 6 and 12)
13. K. Nummiaro, E. Koller-Meier, and L. Van Gool. An Adaptive Color-Based Particle Filter. *IVC*, 21(1):99–110, 2003. (Cited on pages 4, 10, 12, 13, and 19)

14. P. Pérez, C. Hue, J. Vermaak, and M. Gangnet. Color-based Probabilistic Tracking. In *7th ECCV, Copenhaguen, Denmark*, pages 661–675. Springer LNCS, 2002. (Cited on page 4)

15. D. Rowe, I. Reid, J. Gonzàlez, and J. Villanueva. Unconstrained Multiple-people Tracking. In *28th DAGM, Berlin, Germany*, pages 505–514. Springer, 2006. (Cited on pages 7 and 11)

16. E. Wan and R. van der Merwe. The Unscented Kalman Filter for Nonlinear Estimation. In *AS-SPCC, Lake Louise, Canada*, pages 153–158. IEEE, 2000. (Cited on page 4)

17. Y. Wu, T. Yu, and G. Hua. Tracking Appearances with Occlusions. In *CVPR, Wisconsin, USA*, volume 1, pages 789–795. IEEE, 2003. (Cited on page 3)

18. T. Yang, S. Li, Q. Pan, and J. Li. Real-time Multiple Object Tracking with Occlusion Handling in Dynamic Scenes. In *CVPR, San Diego, USA*, volume 1, pages 970–975. IEEE, 2005. (Cited on page 3)

19. T. Zhao and R. Nevatia. Tracking Multiple Humans in Complex Situations. *PAMI*, 26(9):1208–1221, 2004. (Cited on page 3)