Pattern Recognition Letters 32 (2011) 867–877

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Probabilistic people tracking with appearance models and occlusion classification: The AD-HOC system ☆

Roberto Vezzani *, Costantino Grana, Rita Cucchiara

Dipartimento di Ingegneria dell'Informazione - University of Modena and Reggio Emilia, Via Vignolese 905/B, 41125 Modena, Italy

ARTICLE INFO

Article history:
Received 6 October 2009
Available online 10 November 2010
Communicated by R. Davies

Keywords:
Object tracking
Appearance-based tracking
Occlusion
Probabilistic tracking

ABSTRACT

AD-HOC (*Appearance Driven Human tracking with Occlusion Classification*) is a complete framework for multiple people tracking in video surveillance applications in presence of large occlusions. The appearance-based approach allows the estimation of the pixel-wise shape of each tracked person even during the occlusion. This peculiarity can be very useful for higher level processes, such as action recognition or event detection. A first step predicts the position of all the objects in the new frame while a MAP framework provides a solution for best placement. A second step associates each candidate foreground pixel to an object according to mutual object position and color similarity. A novel definition of *non-visible regions* accounts for the parts of the objects that are not detected in the current frame, classifying them as *dynamic*, *scene* or *apparent* occlusions. Results on surveillance videos are reported, using in-house produced videos and the PETS2006 test set.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

Appearance-based tracking is a well established paradigm for predicting, matching and ensuring temporal consistency of detected deformable objects in video streams. Appearance driven techniques are very often adopted as a valid alternative to approaches based on 3D reconstruction and model matching, by computing the visual appearance of the objects in the image plane only, without the need of defining camera, world and object models. Especially in human motion analysis applications, the exploitation of *appearance models* or *templates* is straightforward, since they enable the knowledge not only of the location and speed of visible people but also of their visual aspect at each frame.

In particular, deterministic approaches require that previously selected regions of the image plane (object candidates) are matched against the estimated appearance templates. By means of a suitable similarity function, the best match between observations and stored objects is selected as tracking result.

Tracking systems which use a deterministic and appearance-based approach offer several advantages in real-time applications: generality and flexibility, since no object model is required; speed, since only one solution is maintained at each frame step; and the large amount of information available, since appearance is normally kept at pixel level. Besides, appearance-based tracking of data extracted by a previous segmentation inherits the typical errors caused by segmentation: partial segmentation, splitting and merging problems must be solved at tracking level. In addition, the major challenge of appearance-based tracking is the frequent presence of visual occlusions. Occlusions make the current observation totally or partially unavailable for some time intervals.

For these reasons, many works address occlusion problems: most of them deal only with *dynamic occlusions* (sometime called *inter-object occlusions*), i.e. occlusions due to other moving objects.

Few works only, instead, deal explicitly with the problem of *scene occlusions*, i.e. occlusions due to still objects in the scene which are closer to the camera. This work focuses on the problem of dealing with different classes of occlusions in the appearance-based deterministic tracking framework to follow deformable shapes and in particular human shapes.

The primary goal is to have, at each frame, a pixel-level appearance model as accurate as possible, with a very reactive update process coping with frequent shape variations. At the same time, we want to address the issue of both dynamic and scene occlusions during their occurrence, selectively updating the appearance model.

The paper provides a formal definition of our approach, called *Appearance Driven Human tracking with Occlusion Classification* (*AD-HOC*). A preliminary description of this system was proposed in (Vezzani, 2008).

As one of the main novelties of the work we propose a formal model of *non-visible regions*, i.e. non negligible parts of the appearance model not observable in the current frame. Non visible regions are classified in three classes, depending on the possible

☆ Selected as best paper at Themis2008.

* Corresponding author. Fax: +39 059 2056129.
E-mail addresses: roberto.vezzani@unimore.it (R. Vezzani), costantino.grana@unimore.it (C. Grana), rita.cucchiara@unimore.it (R. Cucchiara).

0167-8655/$ - see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2010.11.003

cause: *dynamic occlusions*, *scene occlusions* and *apparent occlusions* (which are simply shape variations). The object model is differently updated on each case; the appearance is modified with the current color value only in the visible pixel; the probability value associated with each pixel is reinforced in visible pixels, smoothed in non-visible points due to apparent occlusion and "frozen" in points hidden by a dynamic or static occlusion.

AD-HOC tracking can be used for any object shape whatsoever. Nevertheless, in this paper we refer to human tracking since the algorithm is particularly suitable to deal with the dynamic shape changes and non-rigid motion of humans. This is due to the fact that the appearance models are described only using RGB distributions of pixels. It is important to point out that this description requires the assumption that different objects may be distinguished by their color. In fact occlusion between objects which are not statistically separable in the color space may be wrongly managed by the system, impeding the correct object segmentation.

The paper is structured as follows. In the next section some related works of appearance-based tracking are discussed. Section 3 contains an introductory formulation of the probabilistic equations the tracking is based on and explains in detail the tracking algorithm. Finally, in Section 4, qualitative and quantitative results over several test sequences are reported.

2. Related works

The research activity in people tracking has a long history, as highlighted in the famous survey of Yilmaz et al. (2004).

The use of appearance-based tracking with templates for deformable objects, and in particular for humans, is now widespread, since the pioneer works of Haritaoglu et al. (2000) and Bobick and Davis (2001).

In these papers, the adaptive templates were initially modeled more for human activity analysis than for tracking. The former, W4 (Haritaoglu et al., 2000), introduced appearance models and probability maps called "gray-scale textural appearance" and "shape component", respectively. The textural templates kept the adaptive information of the shape's visual appearance at gray-level and are included with the probability maps in the "weighed similarity" function. Templates were not used to cope with people occlusions. The presence of people occluded by other ones (forming a "group of people") was instead addressed with a specific algorithm counting the heads with projective histograms. In (Bobick and Davis, 2001) the authors proposed a two-component temporal template with a *motion energy image* and a *motion history image* which was exploited to recognize different human actions.

In subsequent works, appearance models have been inserted directly in the tracking loop. These models are put in correspondence with the current observation composed by the moving blobs obtained with a segmentation process. Typically with fixed camera videos, the observed data consists of foreground blobs after background suppression. Methods based on mixtures of Gaussians (Stauffer et al., 2000), adaptive median (Cucchiara et al., 2003), a combination of both (Greenhill et al., 2004), or other methods can be exploited to extract foreground points normally grouped into blobs before the tracking algorithm. Senior et al. (2006) defined appearance models and probabilistic maps to track people and vehicles also partially overlapped. In this work, short term occlusions are implicitly taken into account in the adaptive model. Both short term occlusions and small shape deformation contribute to smooth the appearance model according to an adaptive coefficient. Tracking fails if the occlusion's duration is too long. In case of groups of people, projective histograms of the probability map are evaluated: blobs are separated vertically in the parts where the probability map is low. This approach only works if the persons are sufficiently separated. Similarly, in (Cucchiara et al., 2004), in order to cope with dynamic occlusions of other people, the segmented blobs have been grouped into macro-objects with potential occlusions and then appearance-based tracking is solved with models against the points of the macro objects. Similar solutions are currently adopted in many video surveillance systems and prototypes. The work in (Senior et al., 2006) has been implemented in IBM S3 (Brown et al., 2005). The solution of W4 has been improved in (Thome and Miguet, 2005), using texture and shape templates at level of body limbs. The definition of body limbs is often adopted if tracking is followed by a posture classification step. In a very recent work (Ramanan et al., 2007), an initial limb body model initializes the search of people with HMM, and then appearance models of limbs are learned during the tracking to have accurate data about people's postures and actions.

One of the most cited solution is the more recent work of Zhao and Nevatia (2004). They state that blobs cannot be used efficiently for people tracking, since they do not incorporate object constraints; blobs are subject to structural changes (merges and splits) that cause the combinatorial search to be expensive. Therefore, they propose the use of a simple human model (a normalized ellipsis after homography) instead of a blob. In this region, they compute a *textural template T* (appearance model in the RGB space) and a *foreground probability* (*Fp*). Tracking is solved in a discriminative way by evaluating the best position with a similarity function comprising *T*, *Fp*, and the current image. The expected motion is obtained via a Kalman Filter. Similarly, *AD-HOC* is driven by an appearance and probability model. Unlike in (Senior et al., 2006, 2004), we do not associate the model to a blob, nor to a blob in conjunction with a human model like an ellipsis, but we work at pixel level trying to associate each foreground pixel to each object in the tracked object set. This allows a more precise appearance update for each object and a straightforward detection of dynamic occlusions.

Our proposal does not exploit generative methods such as particle filtering (Isard and MacCormick, 2001) Monte Carlo-based methods (Zhao and Nevatia, 2004) or Bayesian networks with HMM (Wu et al., 2003) that keep the status of each object represented by a pdf, possibly discretized by a set of candidate solutions. In many applications where real-time computation is mandatory, a trade-off between number of evaluated solutions and the granularity of each solution must be defined. Generative methods can take thousands of solutions (for instance 1000 or 2000 particles in (Wu et al., 2003)). In order to compensate for this computational charge they are often used in conjunction with compact appearance models, such as color histograms or color distribution (Isard and MacCormick, 2001; Lanz, 2006). In our case, since the appearance model is defined and updated at pixel level and many comparisons are necessary, especially in footage with frequent and large occlusions, discriminative methods that consider tracking as an optimization problem is preferred.

Most of the aforementioned approaches address the problem of occlusion. Generative models do so by definition since they maintain a large set of hypotheses, and some of them survive and are recovered after the occlusion. On the other hand, discriminative appearance-based approaches generally deal only with the problem of occlusion detection. The ratio between the number of observable points and the points of the appearance models provides a measure of occlusion likelihood: in (Zhao and Nevatia, 2004) and in (McKenna et al., 2000), this ratio determines whether the object is partially or totally occluded, and in that case the model is not updated. In (Thome and Miguet, 2005), an appearance model for tracking segmented objects is proposed: in case of occlusions, authors state they adopt particle filtering with a correlation measurement, but there are no information about how they detect

occlusions and how they switch between deterministic and particle filtering tracking.

Also in (Jepson et al., 2003) an occlusion is detected when pixel appearance deviates too much from the object model; Jepson et al. proposed in that paper an appearance model which is a mixture of three components: a stable part, learned over long periods of time, a "fluctuating" part which accounts for rapid temporal variations, and an outlier process. Short-term occlusions are solved by not assigning the occluded pixels to the stable part. In (Liu et al., 2005), the robust Kalman Filter applied to each point makes the appearance model resistant to short-time partial occlusions, but the authors admit its failure to handle long-time occlusions.

We also perform occlusion detection as in (Zhao and Nevatia, 2004; McKenna et al., 2000), with a confidence measure weighted by probability, but we add a step for occluded regions classification to update the model by selectively copying with both static and dynamic occlusions. In (Javed and Shah, 2002) occlusions are classified as "inter-object occlusions", i.e. dynamic occlusions, or occlusions due to "thin scene structures" or "large structures". The first generates group of people, the second a temporary split and the third makes the object disappear. Some heuristic rules are proposed to cope with grouping, splitting and temporary disappearance. The recent work of Wu and Nevatia (2006) is also aimed at coping with both inter-object and scene occlusion. In this case there is no occlusion classification but human tracking is done by parts. A greedy correspondence algorithm is used whenever possible, i.e. when the body parts are visible; otherwise a mean-shift tracker (Comaniciu et al., 2003) is adopted which allows a robust tracking even when parts of the body are not visible.

3. The two-step object tracking

AD-HOC (Appearance Driven Human tracking with Occlusion Classification) tracking considers that the core element of the system is the object O_m, which is described by its state vector $O_m = \{\{o_i\}, \mathbf{x}_m, \vec{v}_m, \Pi_m\}$, where $\{o_i, i = 1,\ldots,N_m\}$ is the set of the N_m points constituting the object, $\mathbf{x}_m$ and $\vec{v}_m$ are respectively the position with respect to the image coordinate system and the velocity of the centroid. Π_m is the probability of being the foremost object, also called *probability of non-occlusion*. Since objects can be seen as overlapped, this probability can be computed as the probability of not being occluded by any other visible object. By using this probability it is also possible to recover a depth order of the detected objects: completely visible objects have a probability of non-occlusion close to 1, and this value decreases for objects in the distance. Thus, the depth can be recovered ordering the objects by descending values of Π_m.

Each point o_i of the object is characterized by its position $\mathbf{x}_i$, i.e. the x and y coordinates computed with respect to the object centroid, by its color components $\mathbf{c}_i = (R,G,B)$ and by its likelihood α_i to belong to the object. By means of the color components and the α values for each pixel of the object, it is possible to generate two different images which represent the object model: the appearance image (Fig. 1(b)) which contains the color of each point and a probability mask (Fig. 1(c)) which reports their reliability. Static or unchanged pixels will have high reliability values, while moving and changing parts (e.g., the arms and the legs) will usually have a low reliability.

At each frame t, the scene contains a set of objects $\mathcal{O}^t = \{O_1,\ldots,O_M\}$ which we suppose are generating the foreground image F^t. As the object models, also the foreground image can be represented by the set of its points $\{f_l, l = 1,\ldots,L\}$, i.e. the points extracted by a foreground–background segmentation technique. If foreground and background are statistically separable in the color space a background subtraction technique may be sufficient, otherwise more complex approaches may be required, since our approach assumes that a reliable foreground mask is provided. Similarly to an object point, each foreground point f_l is characterized by its position $\mathbf{x}_l$ with respect to the image coordinate system and by its color $\mathbf{c}_l = (R,G,B)$.

The tracking system aims at evaluating the set of objects in the scene at time/frame $t+1$ based on the foreground images observed up to now. In a probabilistic framework, this is obtained by maximizing the following probability density function (pdf):

$$p(\mathcal{O}^{t+1}|F^0,\ldots,F^{t+1}) \tag{1}$$

by exploring all possible sets of objects $\mathcal{O}^{t+1}$ and the corresponding state vector values. In order to perform this MAP (maximum a posteriori) evaluation, we make the assumption of having a first order Markovian model, meaning that

$$p(\mathcal{O}^{t+1}|F^0,\ldots,F^{t+1}) = p(\mathcal{O}^{t+1}|F^{t+1},\mathcal{O}^t). \tag{2}$$

Moreover, by using the Bayes theorem, it is possible to write

$$p(\mathcal{O}^{t+1}|F^{t+1},\mathcal{O}^t) \propto p(F^{t+1}|\mathcal{O}^{t+1})p(\mathcal{O}^{t+1}|\mathcal{O}^t)p(\mathcal{O}^t). \tag{3}$$

It is not possible to optimize Eq. (3) analytically, therefore testing of all possible objects sets and variable status would be required, by changing their position, appearance, and probability of non-occlusion. This is definitely unfeasible, so we break the optimization process in two steps, first locally optimizing the position, then updating the appearance model.

In Fig. 2 the complete architecture of *AD-HOC* tracking is shown. The dotted box includes the background suppression module (SAKBOT (Cucchiara et al., 2003)) used in our experiments.

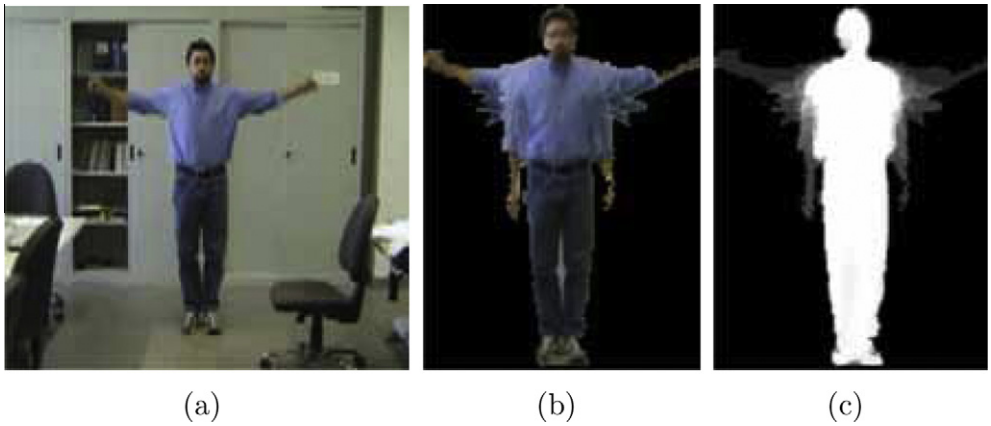


Fig. 1. (a) Input frame, (b) appearance image and (c) probability mask of an object model.

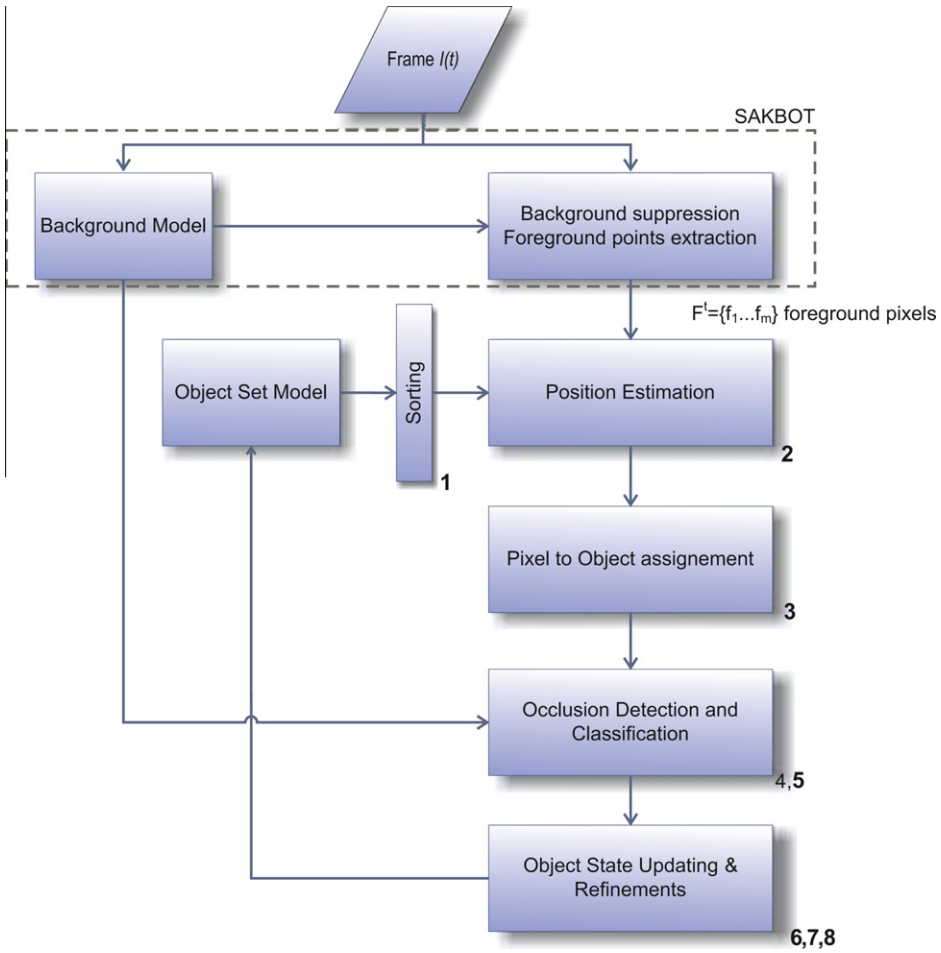


Fig. 2. *AD-HOC* tracking architecture.

3.1. Position optimization

The first task of the algorithm is the optimization of the centroid position $\mathbf{x}_m$ of each tracked object. In Eq. (3) the term $p(\mathscr{O}^t)$ can be removed, since we just keep the best solution from the previous frame. The term $P(\mathscr{O}^{t+1}|\mathscr{O}^t)$, that is the motion model, is obtained for each object via a circular search area of radius r around the predicted position $\hat{\mathbf{x}}_m$, obtained with a first order linear model using the previous position and velocity:

$$\hat{\mathbf{x}}_m^{t+1} = \mathbf{x}_m^t + \vec{v}_m^t, \tag{4}$$

$$p\left(O_m^{t+1}\middle|O_m^t\right) = \begin{cases} \frac{1}{\pi r^2} & \text{if } \left\|\mathbf{x}_m^{t+1} - \hat{\mathbf{x}}_m^{t+1}\right\| < r \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

In order to measure the likelihood of the foreground to be generated by an object, we need to define a relation among the points of F and O_m. This definition is required because the coordinate systems of each object are defined with respect to the object centroid.

To this aim we can define the operator $\delta_m(f_l)$, which returns the point $o_i \in O_m$ corresponding to the foreground point f_l, if it exists:

$$\delta_m(f_l) = o_i \in O_m | \mathbf{x}_l = \mathbf{x}_i + \mathbf{x}_m. \tag{6}$$

Similarly, we can define the inverse operator $\phi(o_i)$ which gives the foreground point f_l corresponding to the object point o_i. From the definitions of the δ and ϕ operators we can obtain the following sets:

- $\delta_m(F)$ is the set of the points belonging to the object O_m which match a foreground point;
- $\phi(O_m)$ is the set of foreground points matching points of the object O_m;
- $\phi(\mathscr{O})$ is the set of foreground points matching a point of at least one object.

Fig. 3 graphically shows the previously defined operators and sets.

The term $p(F|\mathscr{O})$ represents the likelihood of observing the foreground image given the objects positioning. We compute this term in a point-wise manner and three different cases can occur: foreground points matching object points, foreground points without any match (new points), and object points without any match (hidden points). New points may be generated by news objects entering the scene or shape changes in already visible objects; similarly, shape changes as well as occlusions can lead to hidden points. The term $p(F|\mathscr{O})$ can be thus written as:

$$p(F|\mathscr{O}) \propto \Gamma_{match} \cdot \Gamma_{new} \cdot \Gamma_{hide}. \tag{7}$$

Γ_{match} takes into account foreground points which match at least one object point and it is based on the similarity to their corresponding colors:

$$\Gamma_{match} = \prod_{f_l \in \phi(\mathscr{O})} \left[\sum_{O_m} p(f_l | o_i = \delta_m(f_l)) \cdot \Pi_m \right]. \tag{8}$$

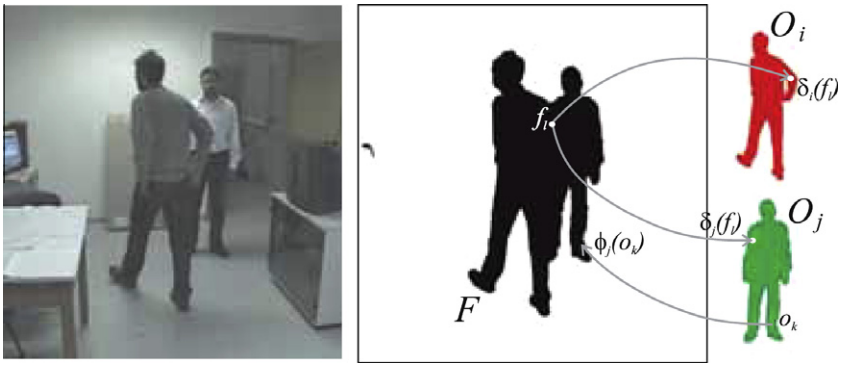


Fig. 3. Schema of the operators ϕ and δ which associate foreground points to object points and vice versa.

The inner sum is done over all the objects O_m overlapping in the considered foreground point f_l and the non-occlusion probability Π_m is used as weight. The conditional probability $p(f_l|o_i)$ of a foreground pixel given an object point is modeled by a Gaussian distribution, centered on the RGB appearance value of the object point and weighted with the α probability:

$$p(f_l|o_i) = \frac{1}{(2\pi)^{3/2}|\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{c}_l-\mathbf{c}_i)^{\mathrm{T}}\Sigma^{-1}(\mathbf{c}_l-\mathbf{c}_i)} \cdot \alpha_i, \tag{9}$$

where $\Sigma = \sigma_3^2 I$ is the covariance matrix in which the three color channels are assumed to be uncorrelated and with fixed variance σ^2. The choice of sigma is related to the amount of noise in the camera. For our experiments we choose values of σ around 20.

For simplicity, we consider the probability of a point to appear (new point) as constant ($\bar{\alpha}$) independently from the reason (shape change, occlusion, etc.). With some empirical estimation we have set $\bar{\alpha} = 0.4$. Instead, the probability of a point to disappear is strictly related to its *alpha* value. The Γ_{new} and Γ_{hide} are thus defined as:

$$\Gamma_{new} = \prod_{f_l \notin \phi(\mathcal{O})} \bar{\alpha}, \quad \Gamma_{hide} = \prod_{O_m} \prod_{o_i \notin \delta_m(F)} \alpha_i. \tag{10}$$

From a computational point of view, the evaluation of the best objects' alignment would require to check all possible combinations of the object positions. It is reasonable to assume that the contribution of the foremost objects in Eq. (7) would be predominant, so we locally optimize the function, by considering only the foremost object for every point. The algorithm proceeds as follows:

1. A list with the objects sorted by their probability of non-occlusion Π_m (assuming that it is inversely related to the depth ordering) is created;
2. The first object O_m is extracted from the list and its position $\mathbf{x}_m$ is evaluated by maximizing the probability of Eq. (7) removing the contributes of all other objects:
$$p(F|O_m) \propto \prod_{f_l \in \phi(O_m)} [p(f_l|\delta_m(f_l)) \cdot \Pi_m] \cdot \prod_{f_l \notin \phi(O_m)} \bar{\alpha} \cdot \prod_{o_i \notin \delta_m(F)} \alpha_i. \tag{11}$$
3. After finding the best $\mathbf{x}_m$, the subset of matched foreground points $\{f_l \in \phi(O_m)\}$ is removed from F, the object O_m is removed from the list as well, and the process continues from 2, until the object list is empty.

Eq. (11) is not statistically significant in case there are too few points, which happens whenever objects are nearly completely occluded; in such a situation few pixels could force a strong change in the object center positioning. For this reason we introduce a *Confidence* measure for the center evaluation, to account for these cases:

$$Confidence(O_m) = \frac{\sum_{o_i \in \delta_m(F)} \alpha_i}{\sum_{o_i \in O_m} \alpha_i}. \tag{12}$$

The confidence measure indicates how large is the currently visible part of the object with respect to the stored model. Each point is weighted with its alpha value. If during the tracking the *Confidence* drops below a given threshold (set to 0.5 in our experiments, i.e. less than half of the model pixels are visible in the current frame), the optimized position is not considered reliable, therefore only the initial prediction is used, setting $\mathbf{x}_m^{t+1} = \hat{\mathbf{x}}_m^{t+1}$.

The confidence measure decreases each time a part of the monitored track is no more visible. Unfortunately different reasons may lead to this situation in addition to occlusion, such as track losses or sudden shape changes. Thus, the confidence measure is not really an occlusion measure and the two terms are not related by a strict one-to-one relationship. However, despite the real reason, whenever the confidence indicator assumes low values, not enough information is available and it is better to trust on the prediction only, which is based on hopefully stronger previous information.

3.2. Point to object assignment

This is the second phase of the optimization of Eq. (3). Once all the objects have been aligned to observed data, we aim at evaluating the remaining parts of each object state. The first assumption we made is that each foreground pixel belongs to one object only. In other words, we need to label the foreground points $f_l \in \phi(\mathcal{O})$ setting their labels λ_l equal to the corresponding object index $m \in 1,\ldots,M$. To this aim we perform a pixel to object assignment as:

$$\lambda_l = \mathrm{argmax}_m p(f_l|\delta_m(f_l)), \tag{13}$$

where the second term is the same Gaussian model as in Eq. (9). Directly from the above assignment rule, we can divide the points belonging to each object model into visible O_m^V and non-visible O_m^{NV} points: the subset $O_m^V = \{\delta_m(f_l)|\lambda_l = m\}$ consists of all the points of O_m that correspond to foreground pixels and have won the pixel assignment of Eq. (13). Instead, non-visible points $O_m^{NV} = O_m - O_m^V$ do not have any match in the foreground mask or they turn out to be occluded.

After the assignment, the α_i and color $\mathbf{c}_i$ values of each object point o_i are then updated:

$$\alpha_i^{t+1} = \begin{cases} \lambda \cdot \alpha_i^t + (1-\lambda), & o_i \in O_m^V, \\ \lambda \cdot \alpha_i^t, & o_i \in O_m^{NV}, \end{cases} \tag{14}$$

$$\mathbf{c}_i^{t+1} = \begin{cases} \lambda \cdot \mathbf{c}_i^t + (1-\lambda) \cdot \mathbf{c}_l, & o \in O_m^V, \\ \mathbf{c}_i^t, & o_i \in O_m^{NV}, \end{cases} \tag{15}$$

where $\mathbf{c}_l$ is the color of the foreground point $f_l = \phi(o_i)$.

For computational reasons, if α_i goes below a predefined threshold (always set to $1-\lambda$), the point o_i is removed from the object. The last step of the object state updating concerns the *probability of non-occlusion* Π_m. To this aim, we first define the occlusion matrix Θ^{t+1}, whose elements θ_{ij}^{t+1} account for the probability that O_i occludes O_j. The element θ_{ij} is incremented each time object O_i occludes O_j. In order to define the occlusion matrix, we need to introduce some quantities. The first one is a_{ij}, which is the number of foreground points matching both O_i and O_j and assigned to O_i. If only two objects are involved in an occlusion, we could define $\theta_{ij} = e^{-\frac{a_{ji}}{a_{ij}}}$, meaning that θ_{ij} is equal to 0 if all the overlapping points are assigned to O_j. On the contrary, if all the points are assigned to O_i, θ_{ij} will be 1. If more than two objects are involved in the occlusion (and thus in the point to object assignment), while evaluating θ_{ij} it is necessary to take into account how many overlapping points are assigned to either of the objects O_i and O_j and how many are assigned to the others. β_{ij} is the percentage of the area shared between O_i and O_j assigned to O_i or O_j. When a third object occludes both O_i and O_j, we cannot infer the mutual depth order between O_i and O_j. In such a situation we assume that their previous order (i.e. their corresponding θ values) is not changing. β_{ij} acts as a weight between the previous ordering and the new estimation. Formally, the compact formulation for θ_{ij} is:

$$\theta_{ij}^{t+1} = \begin{cases} 0, & \beta_{ij} < \beta_{occl}, \\ (1-\beta_{ij})\theta_{ij}^{t}, & a_{ij} = 0, \\ (1-\beta_{ij})\theta_{ij}^{t} + \beta_{ij} e^{-\frac{a_{ji}}{a_{ij}}}, & a_{ij} \neq 0. \end{cases} \tag{16}$$

Fig. 4 describes an example of occlusion. In Fig. 4(e) a_{ij} and a_{ji} are shown, while Fig. 4(f) reports θ_{ij} and θ_{ji}. After an initial uncertainty, the pixel assignments ratio drives the occlusion matrix values to identify the correct depth ordering of the two objects. The probability of non-occlusion Π_m for each object is finally computed as:

$$\Pi_m^{t+1} = 1 - \max_i \theta_{im}^{t+1}. \tag{17}$$

3.3. Shape changes and new objects detection

Within the probabilistic framework previously described we can "assign and track" all the foreground pixels belonging to at least one object. In addition, the foreground image can contain points without any corresponding object, due to shape changes or the entrance into the scene of new objects as well. We suppose that a blob of unmatched foreground points is due to a shape change if it is connected to an object, and in such a situation its points are added to the model of the nearest object; otherwise a new object is created. In both cases the α value of each new point is initialized to the $\bar{\alpha}$ value of Eq. (10).

3.4. Occlusion detection and classification

In the described model, all the non-visible points of an object O_m^{NV} have the same model update: the appearance does not change, while the likelihood α decreases. After a "non-visibility" period depending on λ, they disappear. As already mentioned, points can be non-visible due to either occlusion or shape change. Unfortunately, these two reasons would require two different and conflicting solutions. To keep memory of the object shape even during an occlusion the object model needs to be updated slowly; at the same time a fast updating process better deals with shape changes.

It is thus necessary to introduce a higher level of reasoning in order to discriminate between occlusions and shape changes. The set of non-visible points O_m^{NV} provides the candidate points for occluded regions. After a connected components labeling step (Grana et al., 2010) over O_m^{NV}, a set of *non-visible regions* is created. Each region can be classified as:

1. *dynamic occlusions* R_{DO}: occlusions due to overlapping with another object, closer to the camera; therefore the pixels of this region were assigned to the other object;
2. *scene occlusions* R_{SO}: due to (still) objects, included in the scene and therefore into the background model and therefore not extracted by the foreground segmentation algorithm, but actually located closer to the camera;
3. *apparent occlusions* R_{AO}: regions which are not visible because of shape changes, silhouette motion, shadows, or self-occlusions.

The detection of the first type of occlusion is straightforward, since we always know the position of the objects, and we can easily detect when two or more of them overlap. Dynamic occlusion regions consist of the points shared between an object O_i and another object O_j but not assigned to O_i.

To distinguish between scene and apparent occlusions, positions and shapes of the objects in the background would be helpful, but they are usually not provided. However, the background image can provide information about them, and in particular we propose to directly use the background edges to this aim. In case of a scene occlusion we would expect to find edge points in correspondence of the boundary between the non-visible region and the visible part of the corresponding object. We thus consider the separation points between the visible and non-visible part of each object, i.e. points of the non-visible regions touching visible region points. When the percentage of separation points which is also in correspondence with an edge of the background image is high enough, meaning that the contour of the occluded region has a good match with the background edges, we can infer that a still object (included in the background) is hiding the non-visible part of O_m. The region O_m^{NV} is labeled as R_{SO}, otherwise as R_{AO}. In other words, if the visible and the non-visible part of an object are separated by an edge, then we are plausibly facing an occlusion between a still object in the scene and the observed moving object.

Fig. 5 shows an example: a person is occluded for a large part by a stack of boxes that are included in the background image. Two parts of its body are not segmented and two candidate occlusion regions are generated (Fig. 5(d)): one of them is a shadow included in the object model but which has now disappeared. In Fig. 5(e) the border of O_m^{NV} is shown, with pixels that match with the edges marked in black. In static occlusion (Region R_1) due to a background object the percentage of the border points matching background edges is high; therefore the region is classified as R_{SO}. On the contrary, for the apparent occlusion (the shadow R_2) we have no matching pixels, and consequently this region is classified as R_{AO}.

The updating Eq. (14) must be modified as follows:

$$\alpha_i^{t+1} = \begin{cases} \lambda \cdot \alpha_i^t + (1-\lambda), & o_i \in O_m^V, \\ \lambda \cdot \alpha_i^t, & o_i \in R_{AO}, \\ \alpha_i^t, & o_i \in R_{DO} \cup R_{SO}. \end{cases} \tag{18}$$

Unlike the previous formulation, when a pixel belong to a dynamic or static occlusion ($o_i \in R_{DO} \cup R_{SO}$), the α value is frozen to its previous value, i.e. the system keeps memory of the occluded part.

As a final example, in Fig. 6 a man (with a white t-shirt) is walking behind a table (included in the background model) and stopping there. Moreover, he also crosses another person walking in the opposite direction. Fig. 6(b) shows the foreground regions. In the cases where the model update is adaptive, but not selective with occlusion classification, after a given period of time, depending on the update coefficient, the memory of his occluded legs is

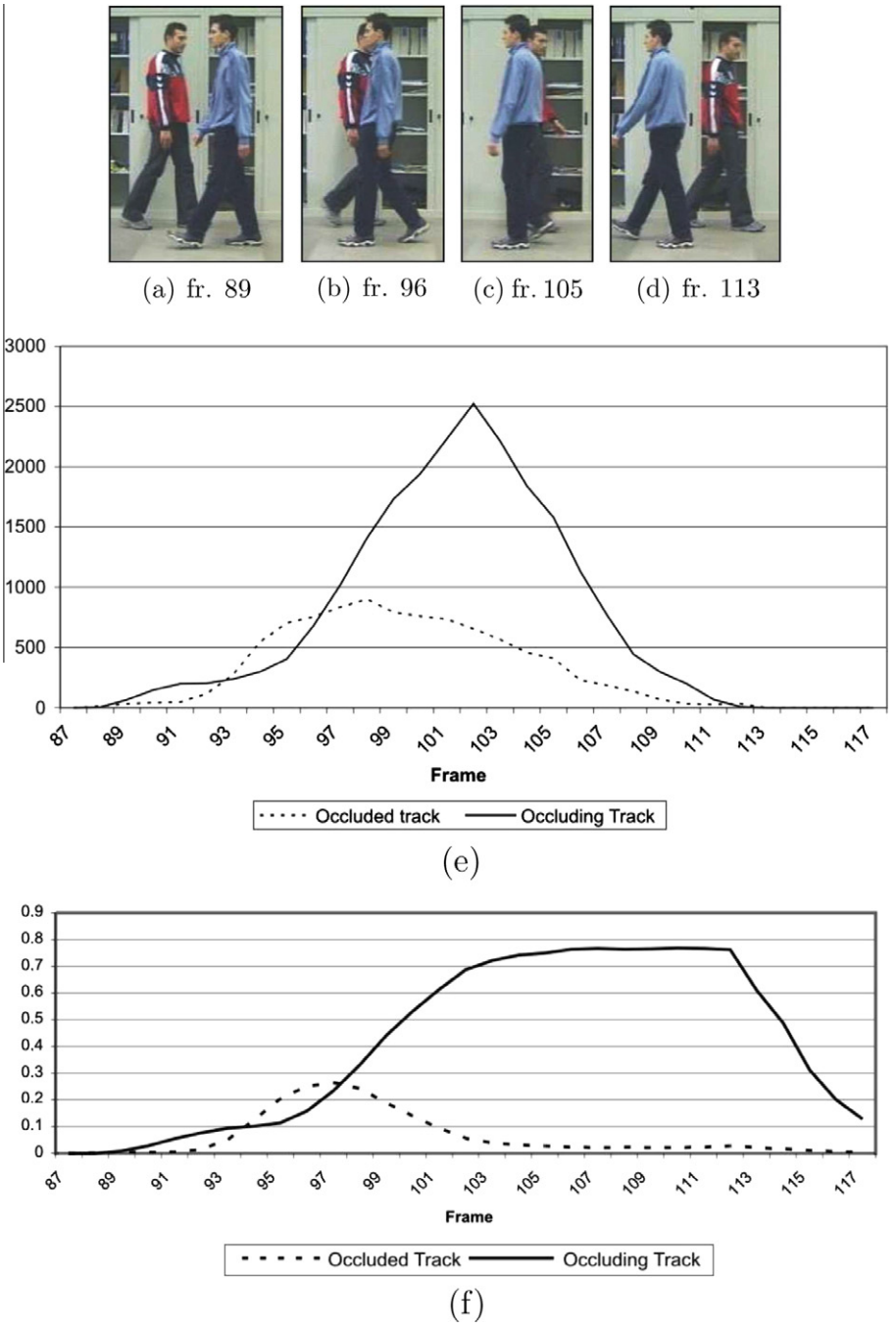


Fig. 4. (a–d) Four sample frames with two people: the person *i*, coming from the left, is occluded by the person *j* coming from the right; (e) number of assigned point a_{ij} and a_{ji} during the occlusion. (f) Probabilities of occlusion $Po(O_i,O_j)^t$ and $Po(O_j,O_i)^t$.

lost, and therefore the centroid and the bounding box are not correctly evaluated (Fig. 6(c)). This is a problem in applications such as posture detection, since the silhouettes of the segmented objects, and all the other features that can be computed, hardly match a standing-up posture model. The problem is solved with the detection of scene occlusions, where the occluded part of the object is frozen, leading to a better evaluation of object's position. The update of the frozen pixels will resume as soon as they become visible again.

3.5. Refinements: split and merge

AD-HOC tracking can handle all static and dynamic occlusions occurring while the object is in the scene, after an initial correct detection. It assumes that initial conditions are ideal, i.e. one person is entering in the scene at a time, not being occluded by other objects. However, in order to correctly manage also these limit conditions, the tracking system must resume well-known problems of objects merging and splitting. The same object may initially appear as two different objects because of occlusions: in the example of Fig. 7, two objects are associated to the only person entering the scene, due to the occlusion by the table. In our work, we exploit the motion vectors of the object to detect whether a merge is needed. In case the objects are near and have similar motion vectors they are merged together (see Fig. 7(c)).

The opposite problem arises when a group of people enters the scene together (see Fig. 8). Since they are represented by a single blob, only one object is created. In order to have a general purpose

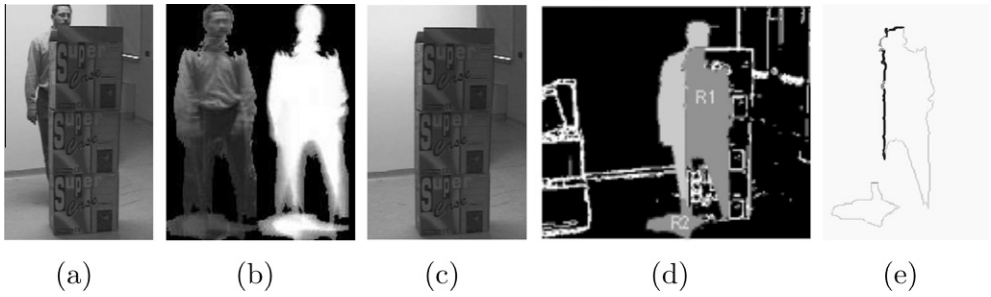


Fig. 5. Example of R_{SO} regions: (a) The input frame. (b) The color and the α representation of the object points. (c) The current background model (d) The visible part of the track (blue) and the candidate occlusion regions (red). (e) the borders of the non-visible regions. Points that have a good match with edge pixels of the background are marked in black. Therefore R_1 is classified as a R_{SO} region while R_2 as a R_{AO} region.

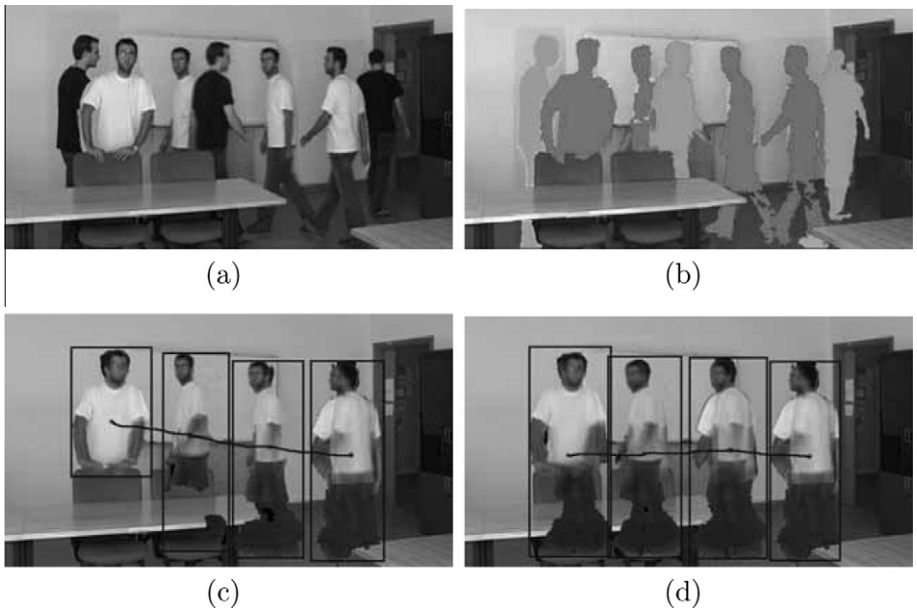


Fig. 6. Example of background object occlusion. (a) A composition of the input frames. (b) Pixel Assignment. (c) Evolution through time of the appearance model of one of the tracks in case the background object occlusions are not handled. (d) The same appearance model with the handling of R_{SO}s.

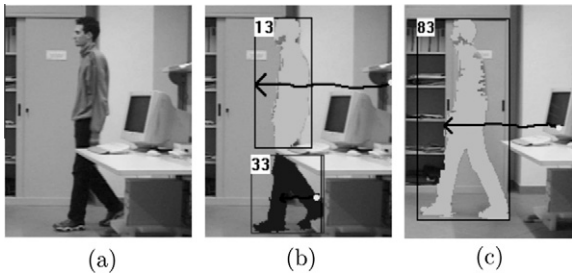


Fig. 7. Merge. (a) Input frame; (b) two objects created; and (c) the objects are merged.

approach independent from the object shape, we decided to split the objects only in case of group separation. To this aim, the probability mask is periodically analyzed to check the presence of two or more well-separated connected components; in such case, the object is split (Fig. 8(c)) and one or more new objects are created.

4. Experiments

The system has been devised for a project of Indoor Surveillance to control the behavior of people at home and detect dangerous situations, such as people falling and lying motionless on the floor for a long time. It has been used for video surveillance with a single camera and, in particular, as a basic step for posture detection (Cucchiara et al., 2005). In such a situation a frame by frame people behavior control requires a complete tracking module with occlusion handling capabilities. The experiments have been carried out on videos coming from ViSOR (Vezzani et al., 2008) and PETS (PETS, 2000). In our experiments, the surveillance system is able to process about 10 frames per second on a standard PC, including an initial visual object segmentation module with background suppression and the shadow removal module (Cucchiara et al., 2003).

Execution times for four benchmark videos (sample frames for each video are shown in Fig. 9) taken from the ViSOR repository are reported in Table 1; the corresponding numbers of each row are reported in the architecture chart of Fig. 2. These videos contain different numbers of people in the scene, as well as different average percentage of frames occupied by objects. Video 1 (Fig. 9(a))



Fig. 8. Split. (a) A single object for two people; (b) the probability mask where two different components are visible; and (c) the object is split in two objects.

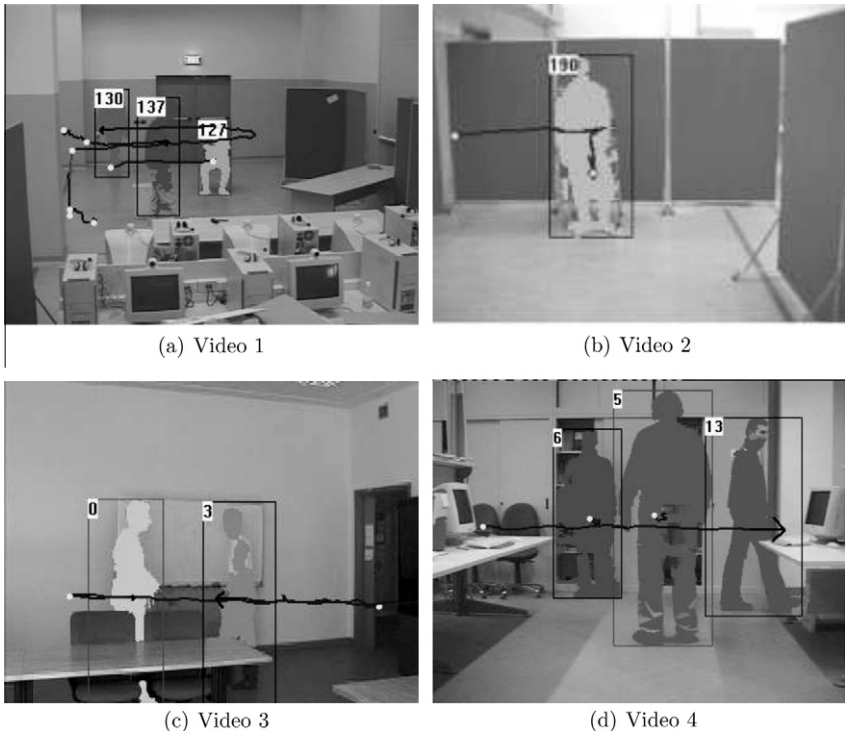


Fig. 9. Sample frames of the videos of Table 1.

Table 1
Average execution times per frame in ms for four sample videos of *AD-HOC*.

	Video 1	Video 2	Video 3	Video 4
% Average object size [pixels]	4%	7%	13%	25%
Number of people	3	1	2	3
1. Sorting by object depth	0.07	0.14	0.14	0.14
2. Position evaluation	13.25	23.25	21.10	57.17
3. Pixel to object assignment	0.44	0.27	0.45	1.11
4. Morphological operations	0.29	0.46	0.49	1.37
5. Occlusion handling	4.06	8.01	10.18	16.48
6. Object state updating	2.27	4.11	4.08	10.07
7. Refinements – merge	0.04	0.04	0.02	0.05
8. Refinements – split	0.33	0.55	0.75	1.41
Total	20.75	36.83	37.21	87.8

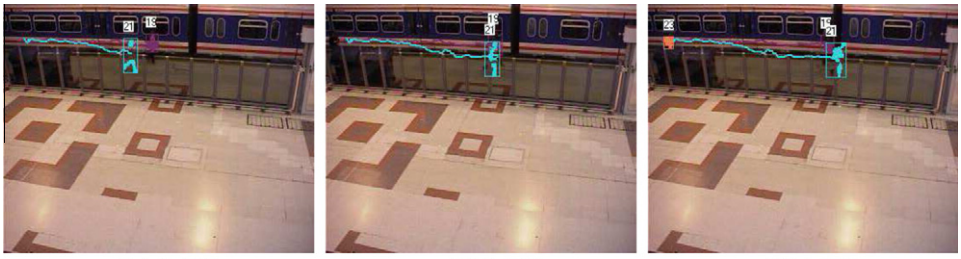
contains three people walking, sitting and overlapping each other. People are often occluded by others or by static objects. The objects sizes are smaller than in the other videos. Video 2 (Fig. 9(b)) shows a single person that frequently changes his posture (stands up, falls down, lies on the floor, sits). In Video 3 (Fig. 9(c)), although two people are present, static occlusions (by a table and two chairs) hide almost half of the objects most of the time. Video 4 (Fig. 9(d)) depicts three people largely overlapping. The challenges for this video are the splitting and merging operations required since people enter the scene occluded by a table and are therefore split in the middle. In the four videos, 100% of people are correctly detected and the edge-based method for occlusion classification is able to correctly classify 85% of non-visible points.

In order to give some quantitative results, we tested the tracking system on some videos of the PETS2006 dataset. In particular we used the seven videos of the third camera, since its point of view has a slightly changing background. A summary of the results obtained is reported in Table 2. In particular, we would like to point out the number of the people walking through the scene and the challenging problems characterizing the videos.

Fig. 10 shows some sample frames of the S1V3 Video (frames 560, 580, and 590). During this part of the video two people are walking behind the Plexiglas, crossing each other. The labels are correctly assigned and kept. During this sequence, the occlusion classification prevents the deformation of the appearance model and the people are never split into legs and torso. Other example frames are depicted in Fig. 11. In particular, Fig. 11(a) demonstrated the performance of the tracking system in very crowded scenes, while in Figs. 11(b) and (c) a splitting event takes place.

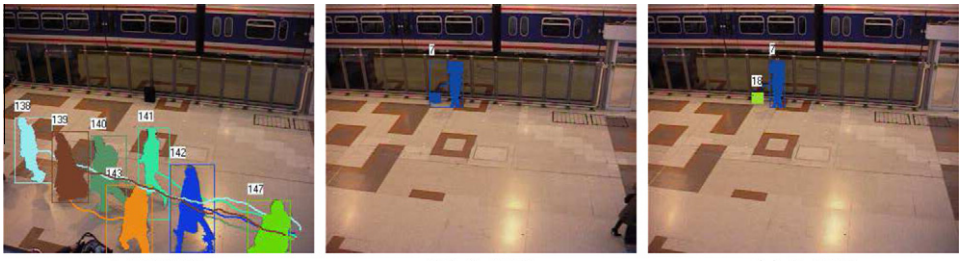
Table 2
PETS2006 dataset (third camera). The number of correctly tracked people and the solved issues are reported. In the last column the errors generated by the tracker are summarized.

Video (#frames)	Number of people	Solved issues	Errors
S1V3 (3020)	35	1 Abandoned luggage 6 Dynamic occlusions 3 Groups of people 29 Correctly tracked people	1 Identity change 1 Split head/feet 2 Groups of 2 people are Not correctly split
S2V3 (2550)	31	1 Abandoned luggage 1 Dynamic occlusions 2 Groups of people 27 Correctly tracked people	2 Groups of 2 people are Not correctly split
S3V3 (2370)	19	1 Abandoned luggage 4 Dynamic occlusions 1 Groups of people 16 Correctly tracked people	1 Identity change 1 Group not correctly Split
S4V3 (3050)	16	1 Abandoned luggage 4 Dynamic occlusions 14 Correctly tracked people	1 Identity change for a Person changing direction during a complete occlusion 1 Split head/feet
S5V3 (3400)	30	1 Abandoned luggage 6 Dynamic occlusions 2 Groups of people 27 Correctly tracked people	1 Split head/feet 1 Group of 2 people not Correctly split
S6V3 (2800)	16	1 Abandoned luggage 2 Dynamic occlusions 1 Groups of 2 people of people 14 Correctly tracked people	
S7V3 (3400)	44	1 Abandoned luggage 10 Dynamic occlusions 3 Groups of people 38 Correctly tracked people	1 Identity change 2 Split head/feet 1 Group of 3 people not correctly split 1 Identity change of the luggage



(a) fr 560 (b) fr 580 (c) fr 590

Fig. 10. Video 1; sample output sequence of the tracking system during the crossing of two people behind a Plexiglas panel.



(a) fr2540 (b) fr 1076 (c) fr 1090

Fig. 11. Some sample frames from Videos 2 and 3.

5. Conclusions

In this work we described the *AD-HOC* tracking system, a complete approach for multiple people tracking in video surveillance applications. In particular, our effort was focused on overcoming large and long-lasting occlusions by using an appearance driven tracking model. The main novelty of the system is the classification of non-visible regions into three classes, which aims at distinguishing between actual occlusions, occlusions with an object belonging to the background, and shape changes. Therefore, based on classification results, a different behavior can be adopted to keep memory of the occluded parts of each object and to recover them once

they appear again. The proposed tracking is very robust and fast; it has been adopted in several projects of indoor and outdoor people surveillance, with many people and real operating conditions.

Acknowledgement

This work is currently under development and improvement within the project THIS (JLS/2009/CIPS/AG/C1-028), with the support of the Prevention, Preparedness and Consequence Management of Terrorism and other Security-related Risks Programme European Commission - Directorate-General Justice, Freedom and Security.

References

Bobick, A.F., Davis, J.W., 2001. The recognition of human movement using temporal templates. IEEE Trans. Pattern Anal. Machine Intell. 23 (3), 257–267.

Brown, L., Connell, J., Hampapur, A., Lu, M., Senior, A., Shu, C.-F., Tian, Y., 2005. Ibm smart surveillance system (s3): A open and extensible framework for event based surveillance. In: Proc. Internat. Conf. on Advanced Video- and Signal-based Surveillance, 2005, vol. 01. pp. 318–323.

Comaniciu, D., Ramesh, V., Meer, P., 2003. Kernel-based object tracking. IEEE Trans. Pattern Anal. Machine Intell. 25 (5), 564–575.

Cucchiara, R., Grana, C., Piccardi, M., Prati, A., 2003. Detecting moving objects, ghosts and shadows in video streams. IEEE Trans. Pattern Anal. Machine Intell. 25 (10), 1337–1342.

Cucchiara, R., Grana, C., Prati, A., Vezzani, R., 2005. Probabilistic posture classification for human behaviour analysis. IEEE Trans. Systems Man Cybernet. 35 (1), 42–54.

Cucchiara, R., Grana, C., Tardini, G., Vezzani, R., 2004. Probabilistic people tracking for occlusion handling. In: Proc. Internat. Conf. on Pattern Recognition, vol. 01. pp. 132–135.

Grana, C., Borghesani, D., Cucchiara, R., 2010. Optimized block-based connected components labeling with decision trees. IEEE Trans. Image Process. 19 (6).

Greenhill, S., Venkatesh, S., West, G.A.W., 2004. Adaptive model for foreground extraction in adverse lighting conditions. In: PRICAI 2004: Trends in Artificial Intelligence, 8th Pacific Rim International Conference on Artificial Intelligence, pp. 805–811.

Haritaoglu, I., Harwood, D., Davis, L., 2000. W4: Real-time surveillance of people and their activities. IEEE Trans. Pattern Anal. Machine Intell. 22 (8), 809–830.

Isard, M., MacCormick, J., 2001. Bramble: A Bayesian multiple-blob tracker. In: Proc. IEEE Internat. Conf. on Computer Vision, pp. 34–41.

Javed, O., Shah, M., 2002. Tracking and object classification for automated surveillance. In: Proc. Eur. Conf. on Computer Vision, pp. 343–357.

Jepson, A., Fleet, D., El-Maraghi, T., 2003. Robust online appearance models for visual tracking. IEEE Trans. Pattern Anal. Machine Intell. 25 (10), 1296–1311.

Lanz, O., 2006. Approximate bayesian multibody tracking. IEEE Trans. Pattern Anal. Machine Intell. 28 (9), 1436–1449.

Liu, X., Tu, P., Rittscher, J., Perera, A., Krahnstoever, N., 2005. Detecting and counting people in surveillance applications. In: Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance, vol. 1. pp. 306–311.

McKenna, S., Jabri, S., Duric, Z., Rosenfeld, A., Wechsler, H., 2000. Tracking groups of people. Comput. Vision Image Understanding 80 (1), 42–56.

PETS, 2000–2007. Pets: Performance evaluation of tracking and surveillance. Website, <http://www.cvg.cs.rdg.ac.uk/slides/pets.html>.

Ramanan, D., Forsyth, D.A., Zisserman, A., 2007. Tracking people by learning their appearance. IEEE Trans. Pattern Anal. Machine Intell. 29 (1), 65–81.

Senior, A., Hampapur, A., Tian, Y.-L., Brown, L., Pankanti, S., Bolle, R., 2006. Appearance models for occlusion handling. Image Vision Comput. 24 (11), 1233–1243.

Stauffer, C., Grimson, W., 2000. Learning patterns of activity using real-time tracking. IEEE Trans. Pattern Anal. Machine Intell. 22 (8), 747–757.

Thome, N., Miguet, S., 2005. A robust appearance model for tracking human motions. In: Proc. IEEE Conf. on Advanced Video and Signal Based Surveillance, vol. 1. pp. 528–533.

Vezzani, R., Cucchiara, R., 2008a. Ad-hoc: Appearance driven human tracking with occlusion handling. In: First Internat. Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS'2008). Leeds, UK, pp. 9–18.

Vezzani, R., Cucchiara, R., 2008b. Visor: Video surveillance on-line repository for annotation retrieval. In: Proc. IEEE Internat. Conf. on Multimedia and Expo (IEEE ICME 2008), Hannover. URL <http://www.openvisor.org>.

Wu, B., Nevatia, R., 2006. Tracking of multiple, partially occluded humans based on static body part detection. In: Proc. IEEE Interant. Conf. on Computer Vision and Pattern Recognition, vol. 1. pp. 951–958.

Wu, Y., Yu, T., Hua, G., 2003. Tracking appearances with occlusions. In: Proc. IEEE Internat. Conf. on Computer Vision and Pattern Recognition, vol. 01. p. 789.

Yilmaz, A., Li, X., Shah, M., 2004. Contour-based object tracking with occlusion handling in video acquired using mobile cameras. IEEE Trans. Pattern Anal. Machine Intell. 26 (11), 1531–1536.

Zhao, T., Nevatia, R., 2004. Tracking multiple humans in complex situations. IEEE Trans. Pattern Anal. Machine Intell. 26 (9), 1208–1221.