Track-based and object-based occlusion for people tracking refinement in indoor surveillance

R. Cucchiara, C. Grana, G. Tardini
Dipartimento di Ingegneria Informatica - University of Modena and Reggio Emilia
Via Vignolese, 905/b
41100 Modena, Italy
+39 059 2056136

{cucchiara.rita,grana.costantino,tardini.giovanni}@unimore.it

ABSTRACT

People tracking deals with problems of shape changes, self-occlusions and track occlusions due to other interfering tracks and fixed objects that hide parts of the people shape. These problems are more critical in indoor surveillance and in particular in home automation settings, in which the need to merge information obtained form different cameras distributed around the house calls for the integration of reliable data obtained during time. Therefore, tracking algorithms should be carefully tuned to cope with occlusions and shape changes, working not only at pixel level but also at region level. In this work we provide a novel technique for object tracking, based on probabilistic masks and appearance models. Occlusions due to other tracks or due to background objects and false occlusions are discriminated. The classification of occluded regions of the track is exploited in a selective model update. The tracking system is general enough to be applied with any motion segmentation module, it can track people interacting each other and it maintains the pixel to track assignment even with large occlusions. At the same time, the model update is very reactive, so as to cope with sudden body motion and silhouette's shape changes. Due to its robustness, it has been used in different experiments of people behavior control in indoor situations.

Categories and Subject Descriptors

I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis – *motion, tracking*.

General Terms

Algorithms, Performance, Design, Experimentation, Theory.

Keywords

People tracking, Video surveillance, Occlusions, Probabilistic models.

1. INTRODUCTION

Tracking is one of the most critical step in processes of people motion capture, people behavior control and indoor video surveillance. The tracking module should be very efficient, in order not to affect the speed of the whole process and, at the same time, it should be very reactive, to adjust the model to sudden changes of silhouette's shape and very robust to occlusions due to other people or objects present in the environment.

A typical scenario is indoor people behavior control: for example, a home or office could be instrumented with a large number of video sensors, that, working together, can identify people within the home and actively track them as they move throughout the environment, providing services that make life easier such as automatic lighting, natural human-home interfaces, and surveillance for security. But to this kind of interaction the coordination system should be provided with the most information possible, in order to reduce the difficulty of the identification task, and to supply a continuous knowledge of the different tracked people positions. In typical applications no more than a single fixed camera for each room can be considered, to limit the overall installation costs. Therefore, the system should supply high level information, such as people color appearance and shape, not only for people tracking from a single point of view, but also to handle the camera hand off and the coordination between different cameras.

In this framework, people tracking must cope with problems of frequent shape changes, self occlusions, and other types of occlusions caused by moving objects *(track-based occlusions*), or fixed objects included in the background model (*object-based occlusions*). Therefore tracking cannot be provided at pixel level only, predicting the pixel motion, but must be supported by assumptions at object-level, assuring spatial coherency of points of the same shape during the time.

Accordingly, we address the problem of tracking by exploiting appearance and probabilistic models, suitably modified in order to take into account the shape variations and the possible region of occlusion. The *appearance image* of a track represents the knowledge we have of an object during tracking. For each point of the track, $AI(\mathbf{x})$ is the estimated aspect of the object, described in the RGB space (see Figure 2.b). The correspondent *probability mask* $P_M(\mathbf{x})$ defines the probability that the point belongs to the track (see Figure 2.c). Since AI and P_M are defined at point level, they are a representation of the temporal coherency of the point, giving us the information of how much

the point is a "inlier" since it has been detected and assigned to the track during the time.

Many works use $AI(\mathbf{x})$ and $P_M(\mathbf{x})$, updating them frame by frame with adaptive functions depending on the single point only. Conversely, we propose a model that is based on $AI(\mathbf{x})$ and $P_M(\mathbf{x})$ to provide tracking, but exploits spatial information in the update process. We verify spatial coherency of the tracks, depending on the type of occlusion, at a shape level by means of global measures (namely *Confidence* and *Likelihood*) and at a region level, analyzing not visible regions of the track.

2. RELATED WORKS

Two aspects are important in the analysis of tracking techniques: the knowledge representation and the models of temporal correlation. For the first point, in literature object-based and image-based approaches have been proposed. Object-based approaches use a representation of the track with a binary mask, extracted by segmenting the image, and a set of shape descriptors like silhouettes or corners, as in [2] in which a temporal graph is used to produce a dynamic template that describes the average shape of the object. Image-based approaches use in addition features extracted also from the aspect of the object in the image itself, as color histograms [12] or mixtures of Gaussians [10]. These can than be clustered to verify spatial relations as in [13] in which similar histogram's bins are merged to produce spatial coherent areas or in [16] in which the mixture of Gaussians describing the background allows also for spatial clustering based on the estimated mean and variance. A set of works uses both object and image-based paradigms, exploiting a probabilistic description of the presence of a pixel in the object, along with color history images [6,17,14].

For what concerns temporal correlation, most of the works employ the Kalman filter [11,17], but also Monte Carlo approaches as the Condensation algorithm [7], and even simpler first order approaches as in [14,6].

In literature, many works address people tracking with occlusion handling, but only few of them manage the pixel assignment during the occlusion, in order to keep the knowledge of the track while the occlusion occurs. The works [8] and [10] solve the problem of occlusions between tracks. In [8] classes of similar color defined with EM algorithm are defined to segment people, tracked frame by frame with a maximum a posteriori probability approach. In [10] pixels assignment is guided by color histograms that model the a priori probability and again a Bayes rule is used to form the posterior probability: thus a visibility index is built to provide information on the depth ordering of tracks. The authors of [1] exploit a stereo vision system to deal with the occlusions and to correctly segment each person in the scene. Furthermore, similar to others [9,14], they use a mask and an appearance template for each track to resolve the temporal tracking. In [15] the tracking system is realized with the fusion of three co-operating parts: an Active Shape Tracker, a Region Tracker and a Head Detector. The Region Tracker exploits the other two modules to solve occlusions.

3. TRACKS, VISUAL OBJECTS AND MACRO OBJECTS

The tracking we propose is totally independent from previous steps of object segmentation. Given the acquisition from a single fixed camera, let us assume to have, for each frame *t*, a set V^t of *Visual Objects*: $V^t = \{VO_1^t,\ldots,VO_n^t\}, VO_j^t = \{BB_j, M_j, I_j, c_j\}$.

Each Visual Object VO_j^t is a set of connected points detected as moving by the segmentation algorithm and described with a set of features: the bounding box BB_j, the blob mask M_j, the Visual Object's color template I_j and the centroid c_j.

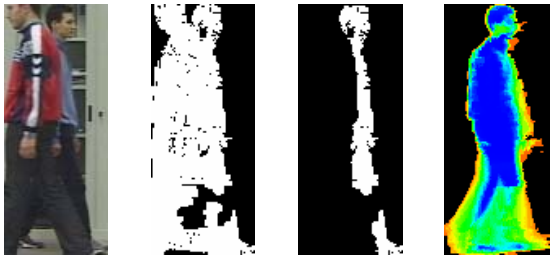
During the tracking execution, we compute a set of tracks τ^t at each frame *t*, that represents the knowledge of the objects present in the scene: $\tau^t = \{T_1^t..T_m^t\}$ with $T_k^t = \{BB_k, AI_k, PM_k, PNO_k, c_k, \mathbf{e}_k\}$, where BB_k is the bounding box; AI_k is the *Appearance Image*, i.e. the estimated aspect (in RGB space) of the track points: each value $AI_k(\mathbf{x})$ represents the "memory" of the object's point previously tracked; PM_k is the *probability mask*: each value of $PM_k(\mathbf{x})$ defines the probability that the point *x* belongs to the track T_k; PNO_k is the *probability of non occlusion* associated with the whole track, that is the probability that the track *k* is not occluded by other tracks; $\mathbf{e}_k$ is the motion vector estimated for the next frame.

Hereinafter, in order to manage a point either of the *VO* or of the Track, we will write improperly $\mathbf{x} \in VO$ or $\mathbf{x} \in T$, meaning that $\mathbf{x} \in BB$ and either the *VO*'s mask $M(\mathbf{x})$ or the probability mask $PM(\mathbf{x})$ of *T* in the point *x* is not zero.

In order to integrate in a single structure all the possible conditions of objects' interaction (merging, splitting, overlapping), the process starts with the construction of a Boolean correspondence matrix *C* between the V^t and T^{t-1} sets. The element $C_{k,j}$ is set to one if the VO_j^t can be associated to the track T_k^{t-1}. The association is established if the track (shifted into its estimated position by means of the vector $\mathbf{e}_k$) and the *VO* can be roughly overlapped, or, in other words, if they have a small distance. It is computed as a Bounding Box Distance (*BBd*) as in the following equation:

(a) (b) (c)

Figure 1. People track example. (a) Visual Object and its trajectory, (b) Appearance Image, (c) Probabilistic Mask



(a) (b) (c) (d)

Figure 2. Depth order pixel assignment.

$$BBd\left(VO_j, T_k\right) = \min_{\substack{\mathbf{x}_k \in BB_k \\ \mathbf{y}_j \in BB_j}} \left(\min\left(\left\| \mathbf{c}_j, \mathbf{x}_k + \mathbf{e}_k \right\|, \left\| \mathbf{c}_k + \mathbf{e}_k, \mathbf{y}_j \right\| \right) \right). \quad (1)$$

In the matrix C five different cases can arise:

1) a track is not associated to any *VO*: the track is missed;
2) a *VO* is not associated to any *T*: a new object is entered into the scene and a new track is generated;
3) a *T* is associated to more than one *VO*;
4) many tracks are associated to the same *VO*;
5) many tracks are associated to many *VOs*.

In the last three cases, the tracking system has to cope with problems of track split, track merge or track overlap. This work is specially oriented to solve these last cases, very frequent in indoor environments with interactions between different people and between people and objects.

To this aim, we define the concept of *Macro-Object* (*MO*) as the union of the *VOs* associated to the same tracks. Initially a *MO* is created for each *VO*, then couples of *MOs* that have at least a track in common are merged. This step is iterated until each track is associated to a single *MO* only. Thus, hereinafter, the tracking will work independently on each single *MO* and on the subset $\tilde{\tau}^t \subseteq \tau^t$ associated with that *MO*.

By adopting *MOs* instead of the segmented *VOs* in the tracking module we can get rid of the problem of managing the many-to-many correspondence case. In general, in fact, a single segmented *VO*, generated from overlapped people, has points that should be assigned to different tracks, or some disjoint *VOs* (due to segmentation errors) should be associated to the same track.

4. PEOPLE TRACKING

The tracking iterates the designed algorithm at each frame and for each pair $\left(MO, \tilde{\tau}^t\right)$. At each iteration, for each track $T_i \in \tilde{\tau}^t$, the algorithm is composed by three steps:

1) *track alignment and pixel to track assignment*: the system searches for the best pixel-level alignment between T_i and MO, and assigns each pixel of the MO to the track with the highest probability to have generated it;
2) *track evaluation*: two measures (Confidence and Likelihood) of T_i are evaluated and the parts of the tracks that are not visible in that frame are segmented (we call them *non-visible regions*) and classified as potential occluded regions;
3) *track update*: the knowledge of the track is updated according with an adaptive model that takes into account the actual RGB appearance of the single point and the results of the track evaluation phase.

A final *track set refinement* process evaluates the set of generated tracks, in order to decide if it is useful to merge or split tracks. The merging/splitting problem is very critical in indoor environment since shadows, segmentation errors, and large occlusions, occurring when a person is entering in the observed scene, generate separate *VOs* that can erroneously create separate tracks that must be merged. On the contrary, if a group of people enters the scene at once, a further analysis on the track appearance is needed, when the people will walk in different directions. Because of this, some further high level considerations, based on the motion and trajectory coherence, are employed to detect this situations and consequently merge and split the corresponding tracks.

5. TRACK ALIGNMENT

In this phase the estimated position of each track is refined with the displacement $\boldsymbol{\delta} = (\delta_x, \delta_y)$ that maximizes a fitting function P_{FIT}.

$$P_{FIT}(T_k, \boldsymbol{\delta}) = \frac{\sum_{\mathbf{x} \in MO} P_{APP}(I(\mathbf{x} - \boldsymbol{\delta}), AI_k(\mathbf{x})) \cdot PM_k(\mathbf{x})}{\sum_{\mathbf{x} \in T_k} PM_k(\mathbf{x})} \quad (2)$$

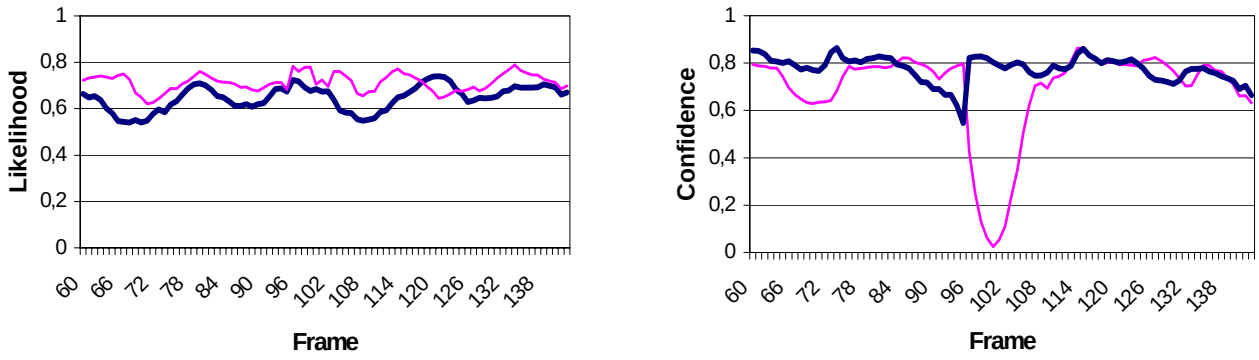
where $P_{APP}\left(RGB_i, RGB_j\right)$ measures the correspondence between the actual RGB color of the point in *MO* and the appearance model of the track. As in [14], we use a spherical Gaussian to approximate the pixel distribution around the mean $\boldsymbol{\mu}$ stored by the model

$$P_{APP}(\mathbf{RGB_i}, \boldsymbol{\mu_i}) = (2\pi\sigma^2)^{-3/2} e^{-\frac{\|\mathbf{RGB_i} - \boldsymbol{\mu_i}\|^2}{2\sigma^2}} \quad (3)$$

Here we supposed that the R,G,B variables are uncorrelated and with identical variance σ^2. This is iterated for all the tracks associated with a MO, with a order proportional to their probability of not occlusion. The $\boldsymbol{\delta}$ displacement is initialized with the value $\mathbf{e}_k$ and searched with a gradient descent approach. The alignment is given by $\boldsymbol{\delta}_{BF}(T_k) = \arg\max_{\boldsymbol{\delta}}\left(P_{FIT}(T_k, \boldsymbol{\delta})\right)$. After each fitting computation, the points of the *MO* matching a track point with high P_{APP} are removed and not considered for the following tracks' fitting. Figure 2 shows a single *MO* (two overlapped people) corresponding to two tracks: Figure 2.a is the image, Figure 2.b is the segmented MO after merging of VOs, and Figure 2.c shows the *MO*'s points remained after the assignment of the front most track, on which the rearmost track is fitted; Figure 2.d is the probability mask of this track.

6. PIXEL TO TRACK ASSIGNMENT

All points of *MO* must be assigned to a track. If a *MO* is in correspondence with a single track, the assignment is straightforward. Instead, in presence of track occlusions, when



(a) **(b)**

Figure 3. Values of Likelihood (a) and Confidence (b) for the sequence in Figure 1.

two or more tracks contend points of the same *MO*, we exploit *a posteriori* probability to solve the assignment:

$$P(T_k|\mathbf{x}) = \frac{P(\mathbf{x}|T_k)P(T_k)}{P(\mathbf{x})}. \quad (4)$$

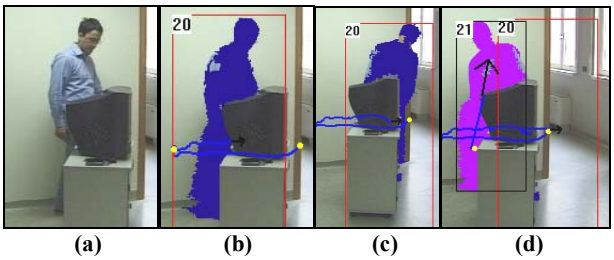
The conditional probability is the product of two terms: $P(\mathbf{x}|T_k) = P_{APP}(I(\mathbf{x}-\boldsymbol{\delta}_{\mathbf{BF}}), AI_k(\mathbf{x})) \cdot PM_k(\mathbf{x})$. It takes into account the difference between the colors of the actual pixel and of the track appearance (as in Equation 3), weighted by the probability that the point belongs to the track $PM_k(\mathbf{x})$. In order to cope with track-based occlusions, the $P(T_k)$ is suitably modeled as the *a priori* probability of seeing T_k, defined as a probability of not occlusion (see section 8 for details). Each point will be assigned to the track that maximizes $P(T_k|\mathbf{x})$ and the set of points assigned to the track T_k is named A_k.

7. TRACK EVALUATION

To evaluate track and occlusion characteristics, the spatial information at two levels is exploited: at object-level, a value of reliability of the fit measure is computed from the fitting value P_{FIT}, at region-level, possible occlusions are classified.

7.1 Likelihood and Confidence

To cope with large occlusions we refined the model by rewriting Equation 2 as:



(a) **(b)** **(c)** **(d)**

Figure 4. Example of erroneous track freezing. (a) Original image; (b) an occlusion causes the *Confidence* value to go very low; (c) the track is still maintained; (d) the track is lost.

$$P_{FIT}(T_k, \boldsymbol{\delta}_{BF}) = Likelihood \cdot Confidence =$$

$$\frac{\sum_{\mathbf{x}\in MO} P_{APP}(I(\mathbf{x}-\boldsymbol{\delta}_{\mathbf{BF}}), AI_k(\mathbf{x})) \cdot PM_k(\mathbf{x})}{\sum_{\mathbf{x}\in MO} PM_k(\mathbf{x})} \cdot \frac{\sum_{\mathbf{x}\in MO} PM_k(\mathbf{x})}{\sum_{\mathbf{x}\in T_k} PM_k(\mathbf{x})} \quad (5)$$

The first term is a measure of how similar are the corresponding pixels of the *MO* and of the track; the second term is the percentage of track points, weighted with their probability, that are visible on the current frame and belonging to the *MO*. Accordingly, when the product of *Likelihood* and *Confidence* is low, the track is considered totally occluded (and since $\vec{\delta}_{BF}$ is not reliable, the estimated $\mathbf{e}_k$ is used as the most reliable displacement). Instead, immediately after an occlusion we want to react without waiting for the best fit value to return to higher values: therefore, if the previous product is low, but the *Confidence* value is growing with respect to the previous frame, the estimated position is updated anyway. Figure 3 shows the variation of *Likelihood* and *Confidence* for the two tracks in sequence of Figure 2. As the rearmost track become occluded, its confidence value decreases, while the likelihood has only little changes due to shape and color variations, so the displacement is computed using the estimation given by $\mathbf{e}_k$ only. The lowest point in *Confidence* value represents the moment of maximum occlusion. After frame 102, the rearmost track has a high *Likelihood* (0.76) and a low, but growing, *Confidence* (0.32) and thus we accept the position refinement given by $\vec{\delta}_{BF}$.

7.2 Occlusion classification

Due to occlusions or shape changes, some points of the tracks remain without any correspondence with a *MO* point. Other proposed techniques that exploit probabilistic appearance models without coping with occlusions explicitly, use only the set of assigned points (A_k) to guide the update process [14]; the mask probability at each point $\mathbf{x} \in \{A_k\}$ is reinforced, while at each point $\mathbf{x} \in \{T_k - A_k\}$ decreases.

In our work, the adaptive update function is enriched by the knowledge of occlusion regions. When this situation happens, the

(a) (b) (c)

Fig. 5. Loss of part of the model because of an occlusion. (a) Input frame; (b) Current segmentation; (c) Appearance model and Probability mask.

Confidence value associated to the track goes lower, because a part of it is no more visible. In order not to lose the memory of the object appearance, the simplest solution could be the use of a threshold on the *Confidence* value to "freeze" the track update. In this approach two possible problems may be encountered:

- if the value is decreasing but it is still higher than the threshold the model is still updated and the probability values in correspondence with the occluded region begin to decrease; if the occlusion is quite long, they disappear. In this case we have an information loss (See the problem in Figure 5.c where the appearance of legs is lost).
- If the value goes under the threshold the model is not updated anymore. In this way, the hidden part is perfectly remembered, but any change in the track appearance is not taken into account. A problem arises when, while is freezed, it changes the visible part appearance as, for instance, when a person changes its direction, abruptly. In Figure 4, it is possible to see a person that rotates on his axis and bends on a side. In this case, since the occlusion area is quite extended, the *Confidence* value is low, so the model cannot adapt to the shape variations; the moment that it changes its direction (Figure 4.c), the track is lost again.

We can deduct that the system works correctly only in cases in which the occlusion lasts for a short time, enough not to lose zones of the track, and if substantial changes in the object appearance do not happen. The choice of track freezing has some difficulties in all the situations in which sudden variations of the track appear.

Given these considerations, the introduction of an higher level reasoning is necessary in order to discriminate between occlusions and other shape changes. The set of non visible points $NV_k^t = \left\{T_k^t - A_k\right\}$ are the candidate points for occlusion regions: in general, they are the points of the tracks that are not visible anymore at the frame *t*. After a labeling step, a set of *not visible regions* (of connected points of NV_k^t) is created, neglecting sparse points or too small regions. Non visible regions can be classified in three classes:

1) *track-based occlusions* R_{TO}: due to overlap of another track, closer to the camera; therefore the pixels of this region were assigned to the other track;

2) *background object-based occlusions* R_{BOO}: due to (still) objects, positioned ahead of the track;

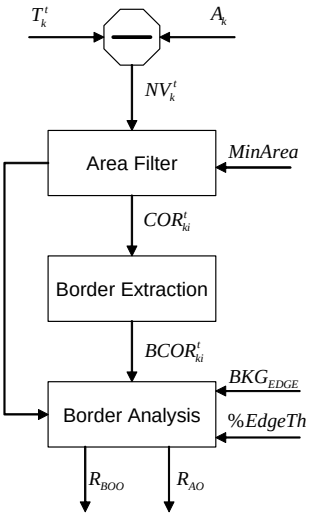


Figure 6. Occlusion region classification algorithm.

3) *apparent occlusions* R_{AO}: regions not visible because of shape changes, silhouette's motion, or self-occlusions.

The presence of occlusions can be detected with the *Confidence* value of Equation 5 decreasing above an alerting value, since in case of occlusion the track's shape changes considerably. In points of actual occlusions (classes 1 and 2), their track model should not be updated since we do not want to lose the memory of the people appearance. Instead, if the *Confidence* decreases due to a sudden shape motion (apparent occlusion), not updating the track would create an error. The solution is a *selective update* according to the region classification. The R_{TO} regions have already been distinguished in the assignment phase: they are composed by the points shared between track T_k and other tracks T_i but not assigned to T_k. In order to distinguish between cases 1) and 2) the background objects should be known. Even if we do not have an exact 3D model for each object in the scene, most of the segmentation algorithms from fixed camera make a background model available. It can be computed at each frame in background suppression segmentation techniques, or can be estimated only when needed. An approximated technique based on an edge analysis of this background image is proposed.

The algorithm is depicted in Figure 6: from the whole set of not visible points, we only keep those with a not negligible value of the probability mask in order to get rid of the noise due to motion. The remaining set of points is segmented into connected regions. Then, for each region, the area weighted with the probability values is calculated, and too small regions are pruned out (*MinArea* parameter in Figure 6). The remaining regions are the Candidate Occlusion Regions (COR_{ki}^t), that must be discriminated into background object occlusions and apparent occlusions.

The borders of the COR_{ki}^t are extracted, and called $BCOR_{ki}^t$. At the same time, the edges of the background model are made available by a simple Sobel edge detector. The pixels of $BCOR_{ki}^t$ corresponding to edge pixels of the background are classified as *bounding pixels*, while the others are said *not bounding pixels*. If

Table 1. System performances.

Video	#pe	#fr	#C	FP	FN
V1	2	1596	2346	0	0
V2	2	1753	1947	188	0
V3	3	1331	687	58	19
V4	3	1270	2698	25	329
PETS 2002 TR2	9	1471	1372	70	29
PETS 2002 TR3	10	1295	882	343	70
PETS 2002 TE1	5	653	538	0	115
PETS 2002 TE2	9	1753	1345	120	288

the percentage of bounding pixels is sufficiently high (typically 40% of the region borders), we can infer that an object is hiding a part of the track, and the region is labeled as R_{BOO}, otherwise as R_{AO}. In Figure 7 an example is shown: a part of a person is occluded; analyzing the border of the candidate occlusion region (Figure 7.c), we find that the majority of bounding pixels is located in correspondence of background edges. Thus, this not visible region is classified as R_{BOO} and the correspondent probabilistic and appearance model is "frozen", that is neither reinforced neither weakened (the dark part in Figure 7.d).

8. SELECTIVE TRACK UPDATE

As the final step, the probability mask, the appearance mask and the probability of not occlusion are updated with adaptive functions. In particular, $\forall \mathbf{x} \in T^t$

$$PM^t(\mathbf{x}) = \begin{cases} \lambda PM^{t-1}(\mathbf{x}) + (1-\lambda) & \mathbf{x} \in A_K \\ PM^{t-1}(\mathbf{x}) & (\mathbf{x} \in R_{TO}) \vee (\mathbf{x} \in R_{BBO}) \\ \lambda PM^{t-1}(\mathbf{x}) & otherwise \end{cases} \quad (6)$$

$$AI^t(\mathbf{x}) = \begin{cases} \lambda AI^{t-1}(\mathbf{x}) + (1-\lambda) I^t(\mathbf{x}) & \mathbf{x} \in A_K \\ AI^{t-1}(\mathbf{x}) & otherwise \end{cases} \quad (7)$$

When the track is generated $P_M{}^t(x)$ is initialized to an intermediate value (0.4 when λ=0.9) while the appearance image is initialized to the image $I^t(\mathbf{x})$.

Defining $Po^t_{i\to k}$ as the probability that track T_i occludes T_k, the non-occlusion probability, $P(T_k) \equiv PNO^t(T_k)$ used in the Bayes rule of Equation 4, is computed as a value proportional to the number $a_{i\to k}$ of shared points assigned to T_i and not to T_k. In particular:

$$PNO^t(T_k) = 1 - \max_{i=1..m}(Po^t_{i\to k}), \quad (8)$$

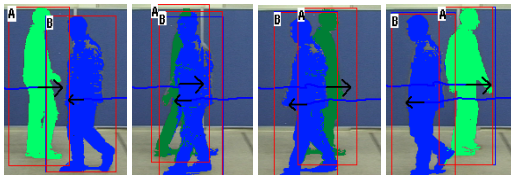
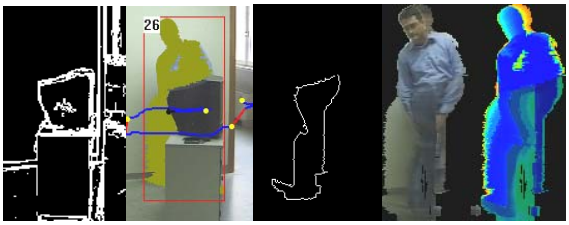


Figure 8. Correct track-based occlusion resolution.



(a) (b) (c) (d)

Figure 7. Edge pixel classification and selective update. (a) Background edges, (b) track, (c) border of non visible region, (d) appearance image and probabilistic mask.

calling $\beta_{i\to k} = \frac{a_{i\to k} + a_{k\to i}}{\|A_i\|}$, $Po^t_{i\to k}$ update model is:

$$Po^t_{i\to k} = \begin{cases} 0 & \beta_{i\to k} < \vartheta_{occl} \\ (1-\beta_{i\to k}) Po^{t-1}_{i\to k} & a_{i\to k} = 0 \\ (1-\beta_{i\to k}) Po^{t-1}_{i\to k} + \beta_{i\to k} e^{-\frac{a_{k\to i}}{a_{i\to k}}} & a_{i\to k} \neq 0 \end{cases} \quad (9)$$

Finally, the motion vector $\mathbf{e}_k$ is estimated according to a constant speed assumption, but enforced by a segmented trajectory schema. Starting from a reference initial position, a certain number of successive motion vectors are linearly interpolated by finding the least squares solution. The solution vector is the motion estimation. In order to check if the interpolation describes correctly the last vectors in the observation window, we evaluate the ratio between the two eigenvalues of the principal direction computation and also if the angle or modulus has changed much from the first value. If the solution fails these checks, a new reference position is created and a new direction can be searched. In this way, an adaptive finite window is used to infer the future motion of the object, able to cope with change of direction in a robust way. This technique has the advantage of being able to handle also non linearities in the measured motion, with respect to classic estimation techniques.

9. RESULTS AND CONCLUSIONS

The system has been devised for a project of Indoor Surveillance to control the people behavior in the house and detect dangerous situations, as people falling and lying on the floor motionless for a long time. The initial description of the video surveillance system was described in [3]. In [4] a reliable people posture classification technique is presented. To cope with a precise frame by frame people behavior control, a complete tracking module with occlusion handling capabilities was needed.

This complex but complete process has been tested over days of indoor video surveillance in two rooms equipped with fixed camera, with some actors and indoor furniture. Moreover, it has been tested over the videos of PETS 2002, in which people walk and interact behind a shop window. Figures 2 and 4 are examples of frames of videos V2 and V3 respectively. Figure 8 shows how the track based occlusion in V1 are correctly managed. Table 1 shows the performance of the system over eight sequences. The values in column *#pe* is the number of people present in the scene, *#fr* is the number of frames, *#C* is the number of correct

assignments at track level and *FP* and *FN* are the number of false positives and false negatives respectively measured against a manual ground-truth. The former are cases in which two or more tracks are assigned to the same person, while the latter are the number of times in which no tracks are assigned to a person. In the video TR3 the high error rate in FP is due to the fact two people enter together in the scene and the system has not the possibility to see them as separate objects. Important experiments are V3 and V4 experiments, where large occlusions due to furniture and track overlaps occur: in these videos a percentage of about 88% of correct assignment is reached.

The tracking approach is not too computationally intensive. In our experiment, the indoor video surveillance is able to process about fifteen frames per second on a standard PC including an initial visual object segmentation module with background suppression, the shadow removal module [5] and a further people posture classification process [4]. The edge-based method is able, on average, to correctly classify the 85% of non visible regions. This approach could be further refined but it is enough precise to allow a good reactivity to silhouette's shape change and, at the same time, a good memory of the appearance model also when a person remains occluded by static objects for a long time.

Therefore the proposed tracking module is a general scheme that exploits probabilistic function and appearance model to keep the knowledge of tracked objects even if they are partially hidden. The robustness and the reactivity is based on a selective update process, that manages differently visible pixels, pixels occluded by static or moving regions and pixels that are not visible anymore, due to shape changes self-occlusions or sudden silhouette's motion.

10. ACKNOWLEDGEMENTS

The project is funded by the European Network of Excellence DELOS of the VI Framework Program.

11. REFERENCES

[1] D. Beymer, K. Konolige, "Real-time tracking of multiple people using continuous detection", Int. Conf. on Computer Vision, 1999.

[2] I. Cohen, G. Medioni. "Detecting and Tracking Moving Objects in Video Surveillance" Proc. of the IEEE CVPR 99, Fort Collins, June 1999.

[3] R. Cucchiara, C. Grana, A. Prati, R. Vezzani, "Computer Vision Techniques for PDA Accessibility of In-House Video Surveillance" in Proceedings of ACM Multimedia 2003 - First ACM International Workshop on Video Surveillance, Berkeley (CA), USA, pp. 87-97, Nov. 2-8, 2003

[4] R. Cucchiara, C. Grana, A. Prati, R. Vezzani, "Probabilistic Posture Classification for Human Behaviour Analysis" in press on IEEE SMC Transactions, Part A: Systems and Humans, special issue on Ambient Intelligence, 2004

[5] R. Cucchiara, C. Grana, M. Piccardi, A. Prati, "Detecting Moving Objects, Ghosts and Shadows in Video Streams" in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, n. 10, pp. 1337-1342, 2003

[6] Haritaoglu, D. Harwood, and L. S. Davis, "W-4: Real-time surveillance of people and their activities," IEEE Trans. PAMI (22) 8, pp. 809-830, 2000

[7] M. Isard and A. Blake. A smoothing filter for condensation. In Proc. ECCV, volume 1, pages 767--781, 1998.

[8] S. Khan, M. Shah, "Tracking People in Presence of Occlusion", Asian Conf. on Computer Vision, Taiwan, Jan 2000.

[9] A. J. Lipton, et al. "Moving target classification and tracking from real-time video" IEEE Image Understanding Workshop, 1998, pp. 129-136.

[10] S.J. McKenna, et. al. "Tracking interacting people", IEEE Int. Conf. on Automatic Face and Gesture Recognition, France, Mar 2000, pp. 348-353.

[11] H.T. Nguyen, and A. W.M. Smeulders. Template tracking using color invariant pixel features. In Proc. ICIP'02, Vol 1, pp. 569 - 573, Rochester, 2002.

[12] P. Pérez, C. Hue, J. Vermaak and M. Gangnet. Color-based probabilistic tracking. ECCV'2002, Copenhagen, Denmark, June 2002

[13] Hyung-Ki Roh, Seonghoon Kang, Seong-Whan Lee: Multiple People Tracking Using an Appearance Model Based on Temporal Color. ICPR 2000: 4643-4646

[14] A. Senior, et al. "Tracking people with probabilistic appearance models", Int. Workshop on Perf. Eval. of Tracking and Surveillance Systems, 2002.

[15] N.T. Siebel, S. Maybank, "Fusion of Multiple Tracking Algorithms for Robust People Tracking", 7th European Conf. on Computer Vision, Denmark, May 2002, vol. IV, pp. 373-387.

[16] C. Stauffer, W.Eric, L. Grimson – "Learning Pattern of Activity Using Real-Time Tracking" –IEEETrans on PAMI (.22)8, August 2000

[17] T. Zhao, R. Nevatia and F. Lv, Segmentation and Tracking of Multiple Humans in Complex Situations, CVPR Kauai, Hawaii, Dec., 2001.