On-line Tracking Groups of Pedestrians with Bayesian Networks *

Pedro M. Jorge
ISEL / ISR
pmj@isel.ipl.pt

Jorge S. Marques
IST / ISR
jsm@isr.ist.utl.pt

Arnaldo J. Abrantes
ISEL
aja@isel.ipl.pt

Abstract

A video tracker should be able to track multiple objects in the presence of occlusions. This is a difficult task since there is not enough information during the occlusion time intervals. This paper proposes a tracking system which solves these difficulties, allowing a long term tracking of multiple interacting objects. First active regions are tracked using simple image analysis techniques. Then, a Bayesian network is used to label/recognize all the detected trajectories, taking into account the interaction among multiple objects. Experimental results are provided to assess the proposed algorithm with PETS video sequences.

1. Introduction

Video surveillance systems aim to detect, track and classify human activities from video sequences captured by single or multiple cameras. Several systems have been recently proposed to perform all or some of these tasks (e.g., see [13, 16, 6, 11, 14]).

The problem becomes difficult when there is an overlap of several objects in the image or the occlusion of some of the objects to be tracked. In such cases it is not possible to track each moving object all the time and inference strategies must be devised in order to recover tracking when enough information becomes available. Fig. 1 shows the superposition of multiple objects with partial occlusion of some of them and their separation into isolated active regions.

Several methods have been used to recover from object superposition and occlusion as well as detection errors (misdetection and false alarms). Some of them are modified versions of the methods used in the tracking of point targets in clutter e.g., nearest neighbor tracker [4], the JPDAF [2], the multiple hypothesis tree or particle filtering [3, 8]. The two problems (target tracking and video objects tracking) are very different however and they should be tackled with different techniques.

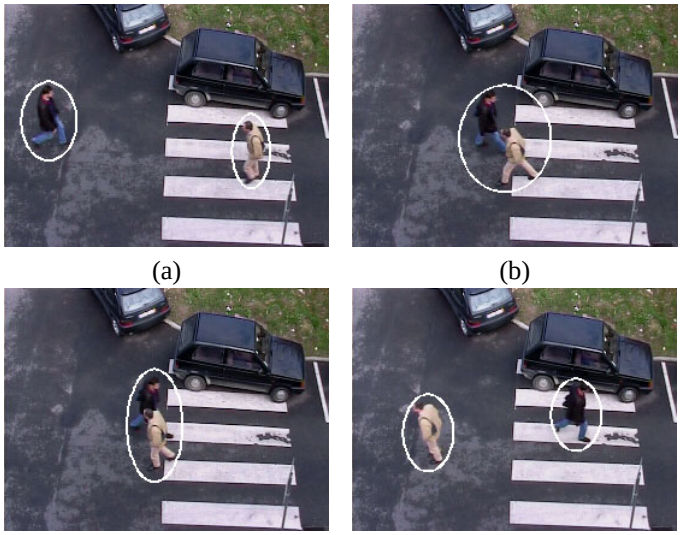


Figure 1. Occlusion example: merge & split

This paper describes a new method which has been developed by the authors which formulates object tracking in video sequences as a labeling problem. It is often simple to detect and track moving objects in video sequences when they are isolated. This can be efficiently done using simple image analysis techniques (e.g., background subtraction). When the object is occluded by other objects or by the background it is usually not possible to separately track. All we can expect to achieve most of the time is to track the group of objects. However, when the object becomes isolated again we should be able to recognize it and recover the track. How can we perform these tasks using all the available information (e.g., information about the interaction among multiple objects, visual characteristics of the objects to be tracked, physical laws)?

This paper described a solution based on Bayesian networks which addresses all these problems. Object tracking is decomposed in two steps: tracking of active regions and

*This work was supported by FEDER and FCT under project LTT (POSI 37844/01).

labeling/recognition of detected trajectories. The labeling task is formulated as an inference problem which is solved by resorting to the use of Bayesian networks which provide useful models for objects interaction and occlusion.

This paper is organized as follow. Section 2 presents an overview of the Bayesian Network tracker. The low level processing is described in section 3 and the generation of the Bayesian network is presented in section 4. Section 5 deals with computation and implementation aspects of the proposed tracker. Section 6 described experimental results and section 7 presents the conclusions.

2. Bayesian Network Tracker

The Bayesian network (BN) tracker consists of two steps. The first step tries to track all the active regions in the video stream. These regions are either isolated objects or groups of objects. The output of the first step is a set of trajectories (see [1, 10] for details).

When the objects overlap in he image domain or when they are occluded, the methods used in first step are not able to reliably associate active regions detected in consecutive frames and the trajectories are broken. A labeling operation is then performed in the second step in order to recognize trajectories of the same object.

Furthermore, we wish to perform a consistent track of object groups i.e., we want to know if a given region is a group, to estimate the group trajectory and to know which objects are in the group.

The labeling operation is performed using a Bayesian network. The Bayesian network plays several roles. It models the interaction among the trajectories of different objects and with the background. Second it provides a consistent labeling which accounts for known restrictions (e.g., in object occlusions, group merging and splitting). Finally, it allows to update the labeling decisions every time new information is available. Fig. 2 shows the output of the two steps for the example of Fig. 1

Let $s_k, k = 1, ..., N$ be the set of segments detected by the low level operations of step 1 (see Fig. 2a). In order to interpret this data, a label x_k is assigned to each segment s_k. Each label identifies all the objects in the segment i.e., if the segment corresponds to a single object, the label is the object identifier. If the segment corresponds to a group, the label is a set of identifiers of all the objects inside the group. The key issue is how to estimate the labels from the information available in the video stream?

Three information sources should be explored. First, labels should be compatible with physical restrictions (e.g., the same object can not be in two places at the same time, the objects velocities are bounded). Second there is prior information which should be used e.g., if the trajectories of two isolated meet a given point and a new trajectory is created, then the new trajectory is probably a group with the two previous objects. Finally, visual features can be easily extracted from the video stream (e.g., color histogram) which aid to recognize the objects especially in the case of isolated objects.

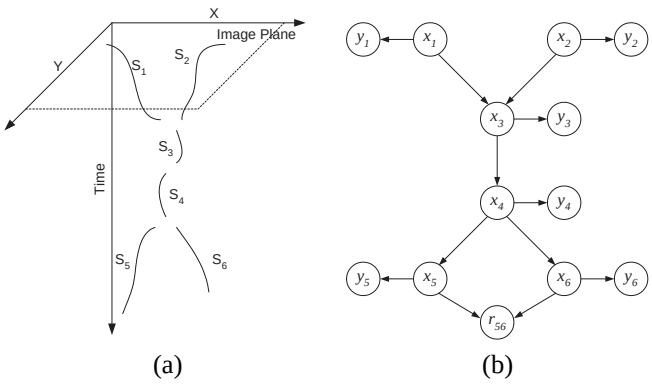


Figure 2. BN tracker: a) object trajectories b) Bayesian network.

A Bayesian network is used to represent the joint distribution of the labels $x = (x_1, ..., x_N)$ and visual features $y = (y_1, ..., y_N)$ detected in the video stream. Additional variables r denoted as restriction variables are also used to guarantee that the physical restrictions are verified (details are given in section 4). Fig. 2.b shows the Bayesian network associated with the example of Fig. 2a. The labeling problem is solved if we manage to obtain the most probable configuration given the observations,

$$\hat{x} = \arg\max_{x} p(x/y, r) \tag{1}$$

where x is the label configuration, y the visual features and r the restriction variables. Each variable corresponds to a node of the BN. Object interaction (trajectory geometry) is encoded in the network topology. Two nodes x_i, x_j are connected if the j-th segment starts after the end of the i-th segment. Additional restrictions are used to reduce the number of connections as discussed in Section 4.

Three issues have to be considered in order to specify a Bayesian network for a tracking problem: i) computation of the network architecture: nodes and links; ii) choice of the admissible labels L_i associated to each hidden node; iii) the conditional distribution of each variable given its parents.

The last two items depend on the type of application. Different solutions must be adopted if one wants to track isolated objects or groups of objects. Group tracking leads to more complex networks since each segment represents multiple objects. These topics are addressed in the next sections. Section 3 describes low level processing and section 4 describes the network architecture.

Since the network represents all the trajectories detected during the operation, the number of nodes increases with time without bound. As mentioned before, this approach can only be used for off-line analysis of short video sequences with few tens of objects. Section 5 describes the extension of this method for on-line operation.

3. Low Level processing

The algorithm described in this paper was used for long term tracking of groups of pedestrians in the presence of occlusions. The video sequence is first pre-processed to detect the active regions in every new frame. A background subtraction method is used to perform this task followed by morphological operations to remove small regions [14].

Then region linking is performed to associate corresponding regions in consecutive frames. A simple method is used in this step: two regions are associated if each of them selects the other as the best candidate for matching [15]. The output of this step is a set of strokes in the spatial/temporal domain describing the evolution of the region centroids during the observation interval.

Every time there is a conflict between two neighboring regions in the image domain the low level matcher is not able to perform a reliable association of the regions and the corresponding strokes end. A similar effect is observed when a region is occluded by the background. Both cases lead to discontinuities and the creation of new strokes.

The role of the Bayesian network is to perform a consistent labeling of the strokes detected in the image i.e., to associate strokes using high level information when the simple heuristic methods fail. Every time a stroke begins a new node is created and the inference procedure is applied to determine the most probable label configuration as well as the associated uncertainty.

4. Network Architecture

The network architecture is specified by a graph, i.e., a set of nodes and corresponding links. Three types of nodes are used in this paper: the hidden nodes x_i representing the label of the i-th segment, the observation nodes y_i which represent the features extracted from the i-th segment and binary restriction nodes r_{ij} which are used to avoid labeling conflicts. The restriction node r_{ij} is created only if x_i and x_j share a common parent. A link is created from a hidden node x_i to x_j if x_j can inherit the label of x_i. Physical constrains are used to determine if two nodes are linked (e.g., the second segment must start after the end of the first and the average speed during the occlusion gap is smaller than the maximum velocity specified by the user). Furthermore, we assume that the number of parents as well as the number of hidden children of each node is limited to 2. Therefore, seven basic structures must be considered (see Fig. 3). These structures show the restriction nodes r_{ij} but the visible nodes y_i are omitted for the sake of simplicity. When the number of parents or children is higher than two, the network is pruned using link elimination techniques. Simple criteria are used to perform this task. We prefer the connections which correspond to small spatial gaps.

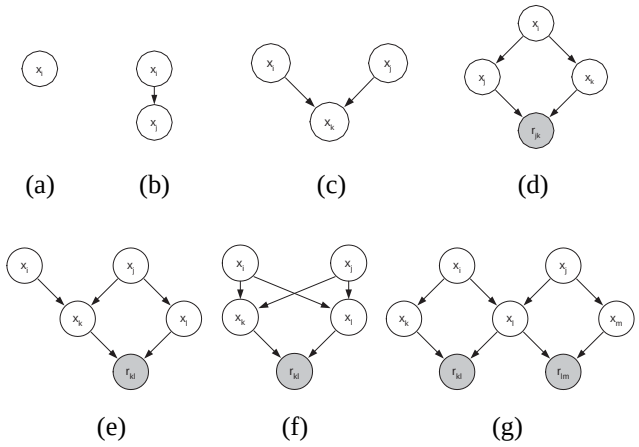


Figure 3. Basic structures (grey circles represent restriction nodes).

4.1. Tracking Isolated Objects

A stroke s_i is either the continuation of a previous stroke or it is a new object. The set of admissible labels L_i is then the union of the admissible labels L_j of all previous strokes which can be assigned to s_i plus a new label corresponding to the appearance of a new object in the field of view. Therefore,

$$L_i = \left[\bigcup_{j \in I_i} L_j \right] \cup \{l_{new}\} \tag{2}$$

where I_i denotes the set of indices of parents of x_i. See Table 1 which shows the labels associated to the hidden nodes of the Bayesian network of Fig. 2. The Bayesian network becomes defined once we know the graph and the conditional distributions $p(x_i|p_i)$ for all the nodes, where p_i are the parents of x_i. As mentioned before, seven cases have to be considered (see Fig. 3). The distribution $p(x_i|p_i)$ for each of these cases are defined following a few rules. It is assumed that the probability of assigning a new label to x_i is a constant P_{new} defined by the user. Therefore,

$$p(x_i = l_{new} | x_j = k) = P_{new} \tag{3}$$

All the other cases are treated on the basis of a uniform probability assignment. For example in the case of Fig. 3c,

k	L_k
1	1
2	2
3	1 2 3
4	1 2 3 4
5	1 2 3 4 5
6	1 2 3 4 6

Table 1. Admissible labels (isolated objects).

k	L_k
1	1
2	2
3	1 2 (1,2) 3
4	1 2 (1,2) 3 4
5	1 2 (1,2) 3 4 5
6	1 2 (1,2) 3 4 6

Table 2. Admissible labels (groups of objects).

x_i inherits the label of each parent with equal probability

$$p(x_i|x_p, x_q) = (1 - P_{new})/2 \tag{4}$$

for $x_i = x_p$ or $x_i = x_q$. Every time two nodes x_i, x_j have a common parent, a binary node r_{ij} is included to avoid conflicts i.e., to avoid assigning common labels to both nodes. The conditional probability table of the restriction node is defined by

$$\begin{array}{c} p(r_{ij} = 1/x_i \cap x_j = \emptyset) = 1 \\ p(r_{ij} = 0/x_i \cap x_j \neq \emptyset) = 0 \end{array} \tag{5}$$

It is assumed that $r_{ij} = 0$ if there is a labeling conflict i.e., if the children nodes x_i, x_j have a common label; $r_{ij} = 1$ otherwise. To avoid conflicts we assume that r_{ij} is observed and equal to 1. Inference methods are used to compute the most probable configuration (label assignment) as well as the probability of the admissible labels associated with each node. This task is performed using the Bayes Net Matlab toolbox [12]. Each stroke detected in the image is characterized by a vector of measurements y_j . In this paper y_j is a set of dominant colors. The dominant colors are computed applying the LBG algorithm to the pixels of the active region being tracked in each segment. A probabilistic model of the active colors is used to provide soft evidence about each node [9]. Each label is also characterized by a set of dominant colors. This information is computed as follows. The first time a new label is created and associated to a segment, a set of dominant colors is assigned to the label. The probability of label $x_j \in L_j$ given the observation y_j is defined by

$$P(x_j/y_j) = \binom{N}{n} P^n (1 - P)^{N-n} \tag{6}$$

where n is the number of matched colors, N is the total number of colors ($N = 5$ in this paper) and P is the matching probability for one color.

4.2. Group Model

This section addresses group modeling. Three cases have to be considered: group occlusions, merging and splitting. Fig. 2 shows a simple example in which two persons meet, walk together for a while and separate. This example shows three basic mechanisms: group merging, occlusion and group splitting. These mechanisms allow us to model more complex situations in which a large number of objects interact forming groups. After detecting the segments using image processing operations each segment is characterized by a group label x_i. A group label is a sequence of labels of the objects present in the group. A Bayesian network is then built using the seven basic structures of Fig. 3. Let us now consider the computation of the admissible labels. The set of admissible labels L_k of the k-th node is recursively computed from the sets of admissible labels of its parents L_i, L_j , starting from the root nodes. This operation depends on the type of connections as follows:

occlusion

$$L_k = L_i \cup l_{new} \tag{7}$$

merging

$$\begin{array}{c} L_k = L_i \cup L_j \cup L_{merge} \cup L_{new} \\ L_{merge} = \{a \cup b : a \subset L_i, b \subset L_j, a \cap b = \emptyset\} \end{array} \tag{8}$$

splitting

$$L_k = L_j = \mathcal{P}(L_i) \cup l_{new} \tag{9}$$

where $\mathcal{P}(L_i)$ is the partition of the set L_i, excluding the empty set. In all these examples, l_{new} stands for a new label, corresponding to a new track. Table 2 shows the set of admissible labels for the example of Fig. 2. Labels 1,2 correspond to the objects detected in the first frame and labels 3-6 correspond to new objects which may have appeared. Conditional probability distributions must be defined for all the network nodes, assuming that the parents labels are known. Simple expressions for these distributions are used based on four parameters chosen by the user:

- P_{occl} - occlusion probability
- P_{merge} - merging probability
- P_{split} - splitting probability
- P_{new} - probability of a new track

These parameters are free except in the case of the occlusion (Fig. 3b). In this case, the conditional probability of x_k given x_i in given by

$$P(x_k/x_i) = \begin{cases} 1 - P_{new} & x_k = x_i \\ P_{new} & x_k = l_{new} \end{cases} \quad (10)$$

The computation of all conditional distributions for the basic structures are detailed in [10].

The probabilistic models for the observations is the same used in the previous section (see (6))

Since the network represents all the trajectories detected during the operation, the number of nodes increases with time without bound. As mentioned before, this approach can only be used for off-line analysis of short video sequences with few tens of objects. The following section describes the extension of this method for on-line operation.

5. On-line Operation

A tracking system should provide labeling results in real time, with a small delay. Therefore it is not possible to analise the video sequence in a batch mode i.e., performing inference after detecting the object trajectories. Furthermore, the model complexity must be bounded since it is not possible to deal with very large networks in practice.

To avoid these difficulties two strategies are proposed in the paper: periodic inference and network simplification. The first strategy consists of incrementally building the network and performing the inference every T seconds. If we denote by $x_0^{kT}, y_0^{kT}, r_0^{kT}$ the variables of the video signal in the interval $[0, kT[$, then the inference problem is given by

$$\hat{x}_0^{kT} = \arg\max_{x_0^{kT}} p(x_0^{kT}/y_0^{kT}, r_0^{kT}) \quad (11)$$

The network grows as before but the labeling delay is reduced to less than T seconds. The solution of (11) can be obtained by several methods e.g., by the junction tree algorithm. The Bayes net toolbox was used in this paper [12].

In practice we wish to have an instantaneous labeling of all the objects i.e., we do not wish to wait T seconds for a new global inference. To obtain on-line labeling a suboptimal approach can be devised which combines the optimal decision obtained at the instant kT with the new information. Let x_i be a hidden node associated to a trajectory active in the interval $[kT, t[$. Using the Bayes law

$$\begin{aligned} P(x_i/y_0^t, r_0^t) &= P(x_i/y_0^{kT}, y_{kT}^t, r_0^{kT}, r_{kT}^t) \\ &= \alpha P(y_{kT}^t, r_{kT}^t/x_i) P(x_i/y_0^{kT}, r_0^{kT}) \end{aligned} \quad (12)$$

where $P(x_i/y_0^{kT}, y_0^{kT})$ is a prior, computed before in the inference step at time kT and $P(y_{kT}^t, r_{kT}^t/x_i)$ represents new information. The choice of the best label x_i is performed by selecting the highest *a posteriori* probability $P(x_i/y_0^t, r_0^t)$. When x_i is a new variable which was created in the interval $[kT, t[$, then we assume that the prior $P(x_i/y_0^{kT}, y_0^{kT})$ is uniform: no label is preferred based on past information.

The previous strategy converts the batch algorithm into an on-line algorithm i.e., it solves the first problem. However, the network size increases as before. To overcome this difficulty, a simplification is needed. The main idea used in this work is to bound the memory of the system.

Old (hidden and visible) nodes influence the labeling assignment of current nodes. However this influence decreases and tends to zero as time goes by: recent variables are more important than old ones. So, we need to use techniques to forget the past. In this paper, we allow a maximum of N nodes and freeze all the other nodes by assigning them the most probable label obtained in previous inferences. In this way, the complexity of the network remains bounded and can be adapted to the computational resources available for tracking. Several strategies can be used to select the nodes to be frozen (dead nodes). A simple approach is used in this paper: we eliminate the oldest nodes and keep the N most recent. A comparison of this strategy with other using synthetic and real data will be presented elsewhere.

6. Experimental Results

Experimental tests were performed with video surveillance sequences using the implemented on-line tracker described in this paper. The tests were performed with PETS sequences (PETS2001 dataset1 training [5] and PETS2004 "*Meet Split 3rdGuy*" [7]) used as benchmarks in video surveillance, as well as other video sequences obtained in an university campus.

Figure 4 shows the performance of the tracker in the PETS2004 "*Meet Split 3rdGuy*" sequence at 25 fps. This is a difficult example, useful to illustrate the performance of the tracker in the presence of occlusions, group merging and splitting. Fig. 4a shows the evolution of all active regions detected in the video stream. This figure displays one of the coordinates of the mass center (column) as a function of time. Every time there is an occlusion or when two or more objects overlap it is no longer possible to associate the new active regions with the ones detected in the previous frame. The trajectories are interrupted in such cases. Fig. 4b shows the labeling results obtained with the on-line algorithm described in the paper. The BN tracker manages to disambiguate most of the occlusions well (only the yellow stroke is misclassified).

Figure 5 shows examples of the tracker performance in group merging and splitting for PETS 2004 sequence. This sequence has three moving objects (3,4,6) and three static objects. The tracker manages to correctly track the three

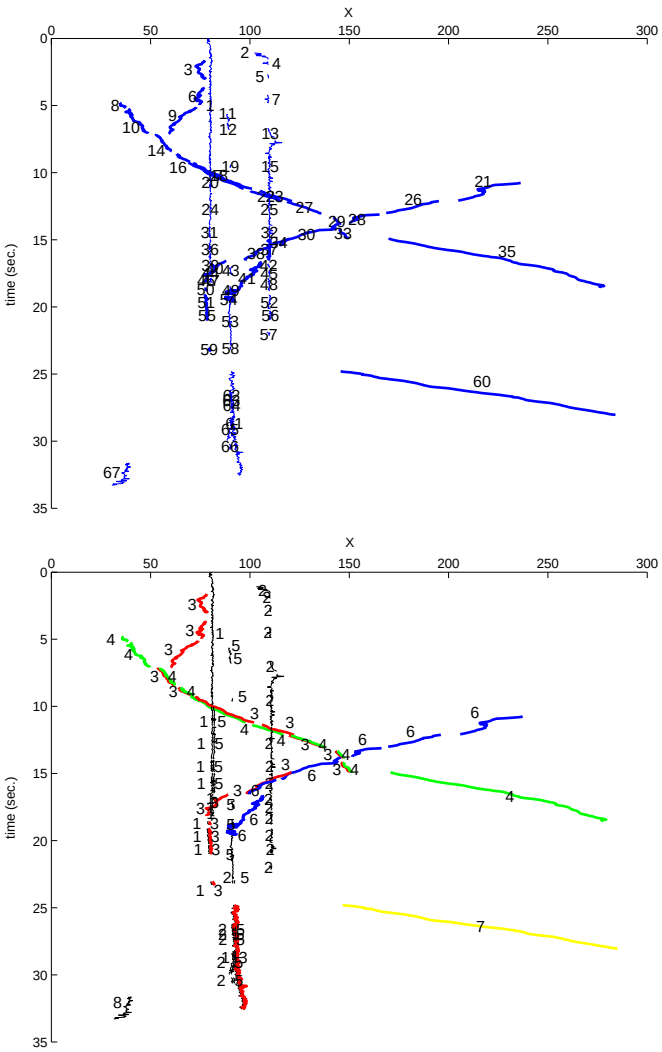


Figure 4. Example (PETS2004 test sequence) : a) detected strokes; b) most probable labeling obtained with the on-line algorithm.

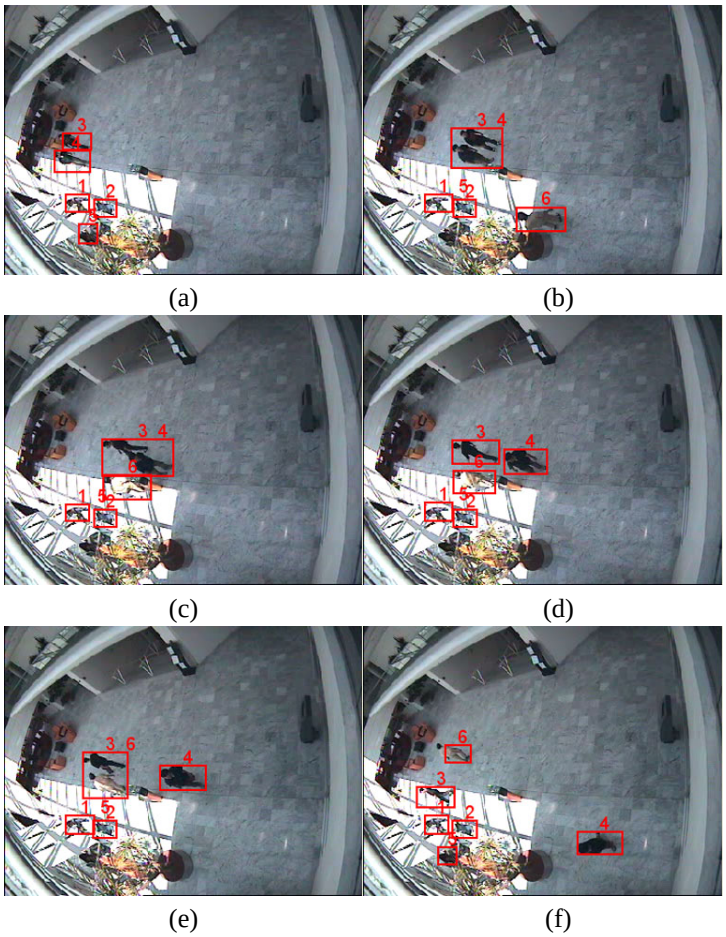


Figure 5. Labeling examples (PETS2004 sequence) after group formation (b,e) and splitting (d,f).

moving objects most of the time as shown in Fig. 5. Three persons walk in separately (Fig. 5a), they merge in groups of two (Figs. 5b,c,e) and they split after a while (Figs. 5d,f). All these events are correctly interpreted by the tracker. Namely, the correct label is assigned after the two splits of Figs. 5d,f.

The tracker has some difficulty to deal with the static objects (labels 1,2,5) since they are not correctly detected by the low level algorithms (background subtraction). These objects remain in the same place during the whole sequence. They are therefore considered as background. However, there are small movements which are detected and appear in Figs. 4, 5.

The Bayesian network is automatically buils during the tracking operation. Figure 6 shows the Bayesian network architecture at the instant $t = 12$ sec. Although the number of nodes grows quickly with time, only the most recent ones are updated by the inference algorithm, therefore keeping the computational burden under control. The gray nodes were classified as frozen by the prunning algorithm and their labels and are not allowed to change.

The BN tracker was also applied to other video sequences as well. Figures 7 and 8 show two examples which illustrate the performance of the tracker in group merging and splitting in other video sequences (PETS2001 and campus sequences). Both occlusions are correctly solved e.e., a correct labeling is produced by the tracker once the persons appear isolated again.

Table I shows statistics which characterize the complexity of the three video sequences and the performance of the tracker. It displays the number of objects in the video se-

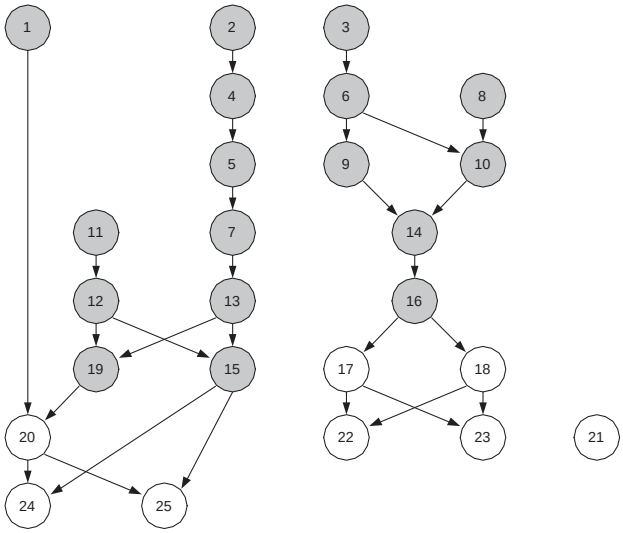


Figure 6. Bayesian network at time instant $t = 12$ sec. (gray nodes are frozen, white nodes are active).

Seq.	NO	NG	NT	LE	D	CT
CAMPUS	7	3	20	0	22.9	2.1
PETS2001	8	5	34	3	120	12.8
PETS2004	7	4	67	5	36	26.6

Table 3. Performance of the BN tracker: Seq. - sequence name; NO - number of objects; NG - number of groups; NT - number of tracks; LE - labeling errors; D - duration (sec.); CT - computational time (sec.).

quence (NO), the number of groups (NG), the number of tracks detected by the low level processing (NT), the number of labeling errors (LE), the duration of the sequence (D) in sec and the computational time (CT). It is concluded from this table that most of the occlusions are well disambiguated by the proposed algorithm (LE $\ll$ NT) and the computational time is smaller than the duration of the sequences[1].

7. Conclusions

This paper presents a system for long term tracking of multiple objects in the presence of occlusions and group merging and splitting. The system tries to follow all moving objects present in the scene by performing a low level detection of trajectories followed by a labeling procedure

[1] these tests were performed with Murphy toolbox for Matlab [12], running on a P4 at 2.8 GHz

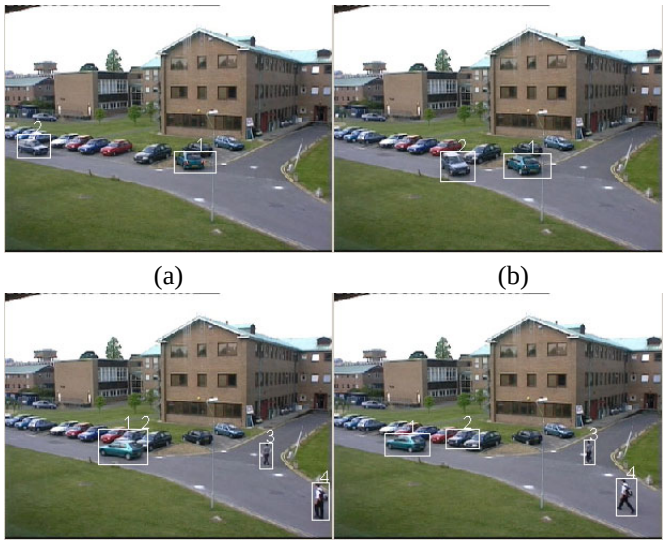


Figure 7. Labeling examples (PETS2001 sequence) after c) group formation and d) splitting.

which attempts to assign consistent labels to all the trajectories associated to the same object. The interaction among the objects is modeled using a Bayesian network which is automatically built during the surveillance task. This allows to formulate the labeling problem as an inference task which integrates all the available information extracted from the video stream and updates the interpretation of the detected tracks every time new information is available. This is a useful feature to solve ambiguous situations such as group splitting and occlusions in which long term memory is needed.

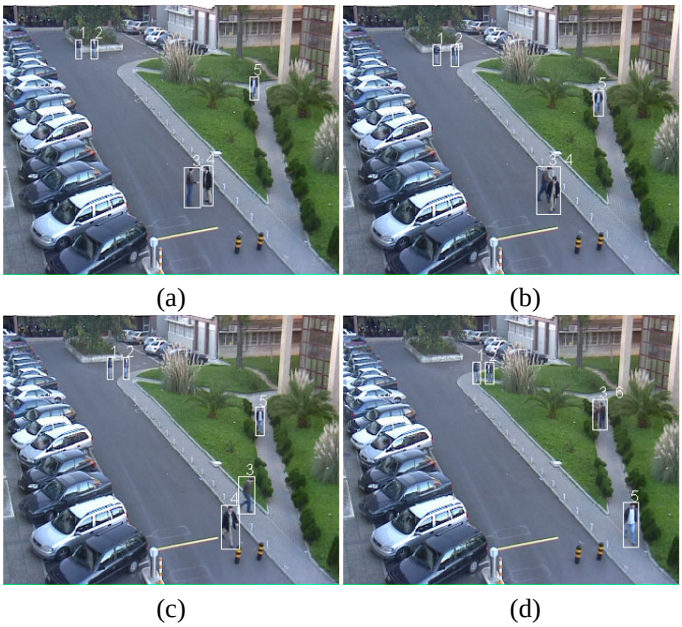
To allow an on line operation of the tracker, inference is periodically performed and pruning techniques are used to avoid a combinatorial explosion of the Bayesian network complexity.

Experimental tests with video sequences were carried out to assess the performance of the system. It is shown that the proposed tracker is able to disambiguate many difficult situations in which there is a strong overlap among different objects.

References

[1] A. Abrantes, J. Marques, and J. Lemos. Long term tracking using bayesian networks. *IEEE ICIP*, III:609–612, September 2002.

[2] Y. Bar-Shalom and T. Fortmann. *Tracking and Data Association*. Academic Press, 1998.



(a) (b)

(c) (d)

Figure 8. Labeling examples (CAMPUS sequence): after b) group formation and c) splitting.

[3] I. Cox and S. Hingorani. An efficient implementation of reid's multiple hypothesis traking algorithm and its evaluation for the propose of visaul traking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(2):138–150, Feb. 1996.

[4] R. Deriche and O. Faugeras. Tracking line segments. *Image and Vision Computing*, 8(4):261–270, 1990.

[5] ftp://pets2001.cs.rdg.ac.uk.

[6] I. Haritaoglu, D. Harwood, and L. Davis. W4: Real-time surveillance of people and their activities. *IEEE Trans. on PAMI*, (22):809–830, 2000.

[7] http://www.dai.ed.ac.uk/homes/rbf/CAVIAR/ CAVIAR project/IST 2001 37540.

[8] M. Isard and A. Blak. Condensation - conditional density propagation for visual tracking. *IEEE Inter. Journal of Computer Vision*, 29(1):5–28, 1998.

[9] F. Jensen. *Bayesian Networks and Decision Graphs*. Springer, 2001.

[10] J. Marques, P. Jorge, A. Abrantes, and J. Lemos. Tracking groups of pedestrians in video sequences. *IEEE WoMOT*, June 2003.

[11] S. McKenna, S. Jabri, Z. Duric, A. Rosenfeld, and H. Wechsler. Tracking groups of people. *Journal of CVIU*, (80):42–56, 2000.

[12] K. Murphy. The bayes net toolbox for matlab. *Computing Science and Statistics*, 33, 2001.

[13] C. Regazzoni and P. Varshney. Multi-sensor surveillance systems. *IEEE ICIP*, pages 497–500, 2002.

[14] C. Stauffer and W. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on PAMI*, 8(22):747–757, 2000.

[15] S. Ullman and R. Basri. Recognition by linear combination of models. *IEEE Trans. on PAMI*, (13):992–1006, 1991.

[16] C. Wren, A. Azabayejani, T. Darrel, and A. Pentland. Pfinder: Real time tracking of the human body. *IEEE Trans. on PAMI*, (19):780–785, 1997.