ON-LINE OBJECT TRACKING WITH BAYESIAN NETWORKS

Pedro M. Jorge *Arnaldo J. Abrantes* *Jorge S. Marques*

ISEL / ISR *ISEL* *IST / ISR*

ABSTRACT

A tracking system based on Bayesian networks was recently proposed. This system deals with difficult situations (e.g., occlusions, group formation and splitting) trying to recover the object identity provided it appears isolated again. This requires an off line processing of the video sequence which prevents its use in real time applications such as video surveillance. This paper describes a modified version of the BN tracker, tailored for on-line tracking of moving objects. This is achieved by gradually forgetting the influence of past information on the current decisions avoiding a combinatorial explosion and keeping the network complexity within reasonable bounds.

1. INTRODUCTION

Several algorithms have been proposed for object tracking in video sequences [11, 2, 10, 4, 7, 9]. Many of them rely on the association of active regions detected in consecutive frames. This is an easy task most of the time. Since the frame rate is high, compared with the object velocity in the image plane, region association can be performed by simple heuristic rules in most cases. An important exception concerns the occlusion of objects by the background or by groups of objects. In this case, it is not possible to track the objects of interest during the occlusion interval and higher level techniques must be adopted to identify the object when it becomes isolated again.

The use of Bayesian networks was recently proposed as a tool to perform long term tracking of moving objects [1, 6]. Object tracking is decomposed in two steps: 1) simple algorithms are used to track non occluded objects and 2) a data conflict module is used to deal with difficult situations (e.g., occlusions, group merging and splitting). The data conflict module performs a labeling operation, i.e., it assigns a label to each detected trajectory. Trajectories of the same object should receive the same label. The labeling operation is performed using a Bayesian network (BN). 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 oclusions, group merging and splitting). Finally, it allows to update the labeling decisions every time new information is available.

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

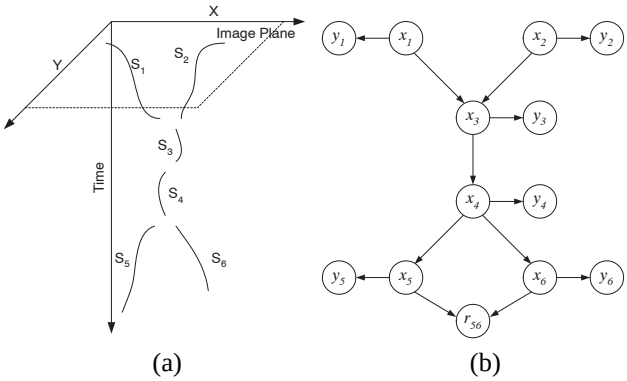

Fig. 1. BN tracker: a) object trajectories b) Bayesian network.

The tracker proposed in [1, 6] works off-line. The object trajectories are first detected in the whole video sequence and they are then labeled using global Bayesian network model. This procedure is used for off-line interpretation of small video sequences but can not be applied for on-line tracking of moving objects since the network complexity grows without bound.

This paper overcomes both difficulties. An on-line version of the Bayesian network tracker is proposed which allows an adaptive interpretation of the data. This is achieved by gradually forgetting the influence of past information on the current decisions avoiding the combinatorial explosion and keeping the network complexity within reasonable bounds.

2. BAYESIAN NETWORK TRACKER

The BN tracker is based in two steps. The first step computes the trajectories s_i of all the objects in the video stream, provided that they are isolated. Every time the object is occluded by other objects or by the background the trajectory is broken and new trajectories are created (see Fig. 1a). The second step assigns a label x_i to each trajectory, representing the identity of the object being tracked. The label is retrieved using the visual properties of each object (e.g., color, shape) as well as the physical restrictions about its

motion. All these variables are described by a probabilistic model: the Bayesian network (see Fig. 1b). The BN has three types of nodes: nodes associated with trajectory features y_i (e.g., color histogram) computed from the video signal, label nodes x_i and restriction nodes r_{ij} which are used to create dependencies among children in the case of group splits. The network defines the joint probability distribution of all x_i, y_i, r_{ij} variables as a product of factors (node conditional distributions) [5].

To specify the network, we have to define the architecture (nodes / links), the set of admissible labels for each node and the conditional distribution of each node given its parents. All of them are automatically computed from the detected trajectories using simple rules. Links define causal dependencies between pairs of variables (e.g., the label of a splitting group has a direct influence on the labels of the sub groups). The children of each node inherit the parents admissible labels with addtional elements corresponding to group labels. The conditional distributions are defined using simple heuristic rules defined by the user. See [1, 6] for details.

The best labeling is obtained by the MAP method

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

where $x = \{x_i\}$ are the label variables, $y = \{y_i\}$ the observations and $r = \{r_{ij}\}$ are binary restriction nodes which are set to 1.

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.

3. 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

$$\hat{x}_0^{kT} = \arg\max_{x_0^{kT}} p(x_0^{kT}/y_0^{kT}, r_0^{kT}) \tag{2}$$

The network grows as before but the labeling delay is reduced to less than T seconds. The solution of (2) can be obtained by several methods e.g., by the junction tree algorithm. The Bayes net toolbox was used in this paper [8].

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

$$\begin{aligned} P(x_i/y_0^t, r_0^t) &= P(x_i/y_0^{kT}, y_{kT}^t, r_0^{kT}, r_{kT}^t) \\ &= \alpha P(y_{kT}^t, r_{kT}^t/x_i) P(x_i/y_0^{kT}, r_0^{kT}) \end{aligned} \tag{3}$$

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.

4. EXPERIMENTAL RESULTS

Experimental tests were performed with video surveillance sequences using the batch algorithm and the on-line tracker described in this paper. The tests were performed with PETS 2001 sequences, used as a benchmark in video surveillance, as well as other video sequences obtained in a university campus [3]. Inference was performed every 15 seconds in the on-line algorithm and a maximum number of ancestor nodes N=4. The performance of the on-line algorithm was always identical to the performance of the batch algorithm in all the tests. Long sequences were only processed by the on-line algorithm since the batch version gets stuck with the increase of complexity after the first few minutes.

Figure 2 shows the performance of the tracker in the PETS data set 1 (training) sampled at 25 fps during the first

Seq.	NO	NG	NT	LE	L	CT
PETS 1	8	5	34	3	120	12.8
CAMPUS	7	3	20	0	22.9	2.1

Table 1. Performance of the BN tracker: Seq. - sequence name; NO - number of objects; NG - number of groups; NT - number of tracks; LE - labeling errors; L - length (sec.); CT - computational time (sec.).

120 sec. This sequence is useful to illustrate the performance of the tracker in the presence of occlusions, group merging and splitting. Fig. 2a 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. 2b shows the labeling results obtained with the on-line algorithm described in the paper. The algorithm manages to disambiguate most of the occlusions well. Only 3 labeling errors are observed in a total of 34 trajectories: label 6 and the switch between labels 3 and 8 after the split for t=110s (see Fig. 2b). The output of the on-line algorithm was compared with the batch results during the first 113.6 s. The same labeling was obtained in both cases with important computational savings (CPU times[1]: 258 s (batch), 10 s (on-line)). In this example, the computation time of the labeling algorithm is 15% of the sequence duration while the batch algorithm performs off-line and it is not able to process the whole sequence.

Figure 3 shows the evolution of the Bayesian network, for the PETS sequence, at three instants: although the number of nodes grows quickly with time, only the most recent ones are active and updated by the inference algorithm, therefore keeping the computational burden under control.

Figures 4 and 5 show two examples which illustrate the performance of the tracker in group merging and splitting. A correct labeling is produced in both cases.

Table I shows the statistics which characterize the complexity of the video sequence and the performance of the tracker namely: number of objects, number of groups, number of tracks, labeling errors length and computational time. It can be observed that most of the occlusions are well disambiguated by the proposed algorithm and the compuational time is low (15% of the sequence duration).

5. CONCLUSIONS

This paper describes an on-line version of the Bayesian network tracker assuming as starting point the algorithm proposed in [6]. Video objects are tracked using a two step

[1]these tests were performed with Murphy toolbox for Matlab [8], running on a P4 at 2.8 GHz

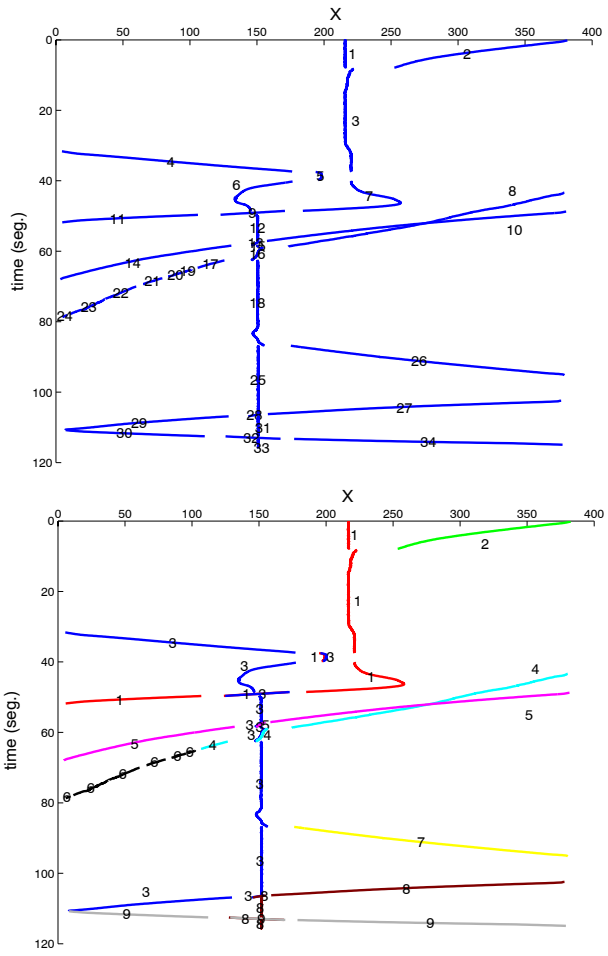

Fig. 2. Example (PETS: test sequence 1): a) detected strokes; b) most probable labeling obtained with the on-line algorithm.

approach: object trajectory detection and trajectory labeling. Object trajectories are detected by simple low level operations and trajectory labeling is performed by statistical inference using a Bayesian network model to represent the interactions among trajectory labels.

To allow an on line operation of the tracker, inference is periodically performed every 15 s and pruning techniques are used to bound the size of the Bayesian network avoiding an exponential increase of computational complexity. Complexity reduction is achieved as follows. Instead of trying to represent the joint probability distribution of all the trajectories detected by the system, we only try to model the joint probability distribution of the most recent trajectories, forgetting the influence of past uncertainty on current labels.

The performance of the system was evaluated using PETS 2001 sequences. It is shown that no degradation of quality is observed in these sequences while the computation time was reduced by an order of magnitude. The computation time associated with the labeling operation is about 15% of

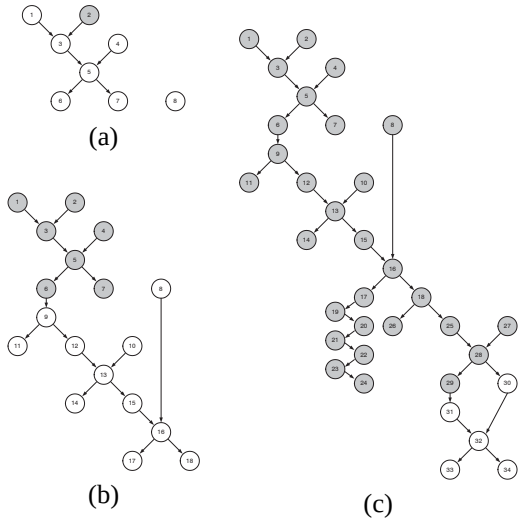

Fig. 3. Network evolution: Bayesian network at three time instants (gray nodes are frozen, white nodes are active).

the duration of the sequence. Furthermore, it is now possible to process sequences of unlimited length with the proposed algorithm.

6. REFERENCES

[1] A. Abrantes, J. Marques and J. Lemos, "Long Term Tracking Using Bayesian Networks", *IEEE ICIP*, vol. III, pp. 609-612, Rochester, September 2002.

[2] F. Bremond and M. Thonnat, "Tracking Multiple Non-rigid Objects in Video Sequences", *IEEE Trans. on CSVT*, 8, pp. 585-591, 1998.

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

[4] I. Haritaoglu, D. Harwood and L. Davis, "W4: Real-Time Surveillance of People and Their Activities", *IEEE Trans. on PAMI*, 22, pp. 809-830, 2000.

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

[6] J. Marques, P. Jorge, A. Abrantes and J. Lemos, "Tracking Groups of Pedestrians in Video Sequences", *IEEE WoMOT*, Madison USA, June 2003.

[7] S. McKenna S. Jabri, Z. Duric, A. Rosenfeld and H. Wechsler, "Tracking Groups of People", *Journal of CVIU*, 80, pp. 42-56, 2000.

[8] K. Murphy, The Bayes Net Toolbox for Matlab, *Computing Science and Statistics*, 33, 2001.

[9] C. Regazzoni and P. Varshney, "Multi-Sensor Surveillance Systems", *IEEE ICIP*, pp. 497-500, 2002.

[10] C. Stauffer and W. Grimson, "Learning Patterns of Activity Using Real-Time Tracking", *IEEE Trans. on PAMI*, 22, 8, pp. 747-757, 2000.

[11] C. Wren, A. Azabayejani, T. Darrel and A. Pentland, "Pfinder: Real Time Tracking of the Human Body", *IEEE Trans. on PAMI*, 19, pp. 780-785, 1997.

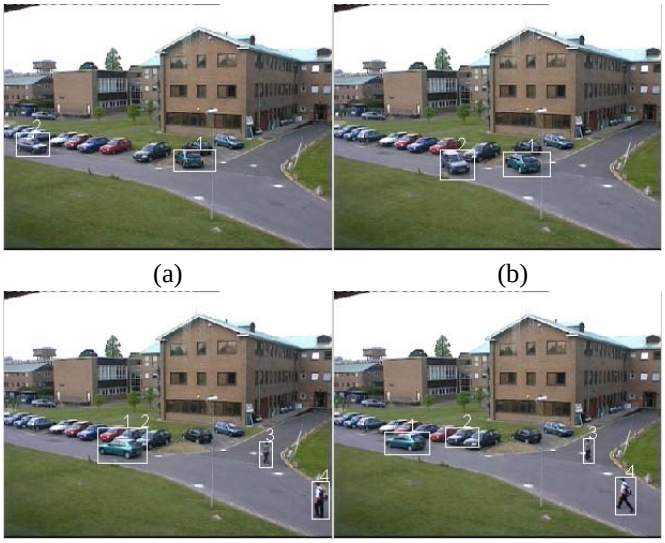

Fig. 4. Labeling examples (PETS sequence) after a) group formation and b) splitting.

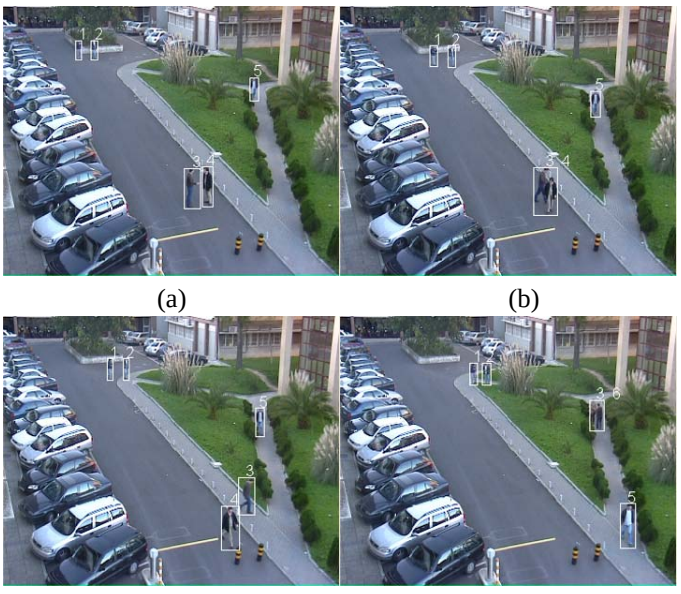

Fig. 5. Labeling examples (CAMPUS sequence): after a) group formation and b) splitting.