Event Recognition with Fragmented Object Tracks

Michael T. Chan, Anthony Hoogs, Zhaohui Sun, John Schmiederer,
Rahul Bhotika and Gianfranco Doretto
GE Global Research
One Research Circle
Niskayuna, NY 12309, USA
{chanm,hoogs,sunzh,schmiede,bhotika,doretto}@research.ge.com

Abstract

Complete and accurate video tracking is very difficult to achieve in practice due to long occlusions, traffic clutter, shadows and appearance changes. In this paper, we study the feasibility of event recognition when object tracks are fragmented. By changing the lock score threshold controlling track termination, different levels of track fragmentation are generated. The effect on event recognition is revealed by examining the event model match score as a function of lock score threshold. Using a Dynamic Bayesian Network to model events, it is shown that event recognition actually improves with greater track fragmentation, assuming fragmented tracks for the same object are linked together. The improvement continues up to a point when it is more likely to be offset by other errors such as those caused by frequent object reinitialization. The study is conducted on busy scenes of airplane servicing activities where long tracking gaps occur intermittently.

1. Introduction

Recognizing events in video often depends on accurately tracking objects involved in an event from the beginning to the end [8, 13, 9, 7]. However, achieving sufficient tracking performance can be very difficult in scenes with multiple moving objects, occlusions, shadows, low resolution, and other complications. Recently, methods have been proposed to track objects under these conditions by linking track fragments in a multi-object framework [10] or by background layers [14].

The goals of this paper are to determine the feasibility of event recognition when the input tracks are fragmented, and which level of track fragmentation is best for event recognition. For our purposes, a fragmented track is defined as a set of tracks with no temporal overlap corresponding to the same object. Fragmentation occurs when an object is tracked intermittently, creating a set of *tracklets* for one object. The tracklets may be grouped together by an algorithm such as [10]. The problem for event recognition is that fragmented tracks have gaps in space and/or time, during which no observations of the object are available. To achieve reasonable performance under real-world conditions, event recognition must discriminate between different event types and non-events, despite tracking gaps.

Our approach is to use Dynamic Bayesian Networks (DBN's) to represent events, and to interpolate over tracking gaps in both space and time. DBN's provide robustness w.r.t. the noisy data created by interpolation and tracking in general, and have been used for event or activity recognition [6, 12]. We achieve additional robustness by using semantic spatio-temporal primitives similar to [8, 4] without directly modeling object trajectories. Each event class is modeled as a DBN, and recognition is performed by testing all possible assignments of tracks to event actor roles, for each event model. (Other constraints can be introduced to avoid testing all role assignments, but it is out of the scope of the current paper.) Furthermore, our model represents complex events defined by interactions between multiple objects, and therefore multiple tracks, each of which may contain gaps.

To examine the behavior of this model as a function of tracking quality, we obtain recognition results on different levels of tracking performance [1] generated by varying the tracking lock score threshold controlling track termination; tracks are terminated when they fall below this threshold. All tracks are automatically initialized, so when an object is lost, but continues in motion, it is likely that a new track will be initiated for the same object. Consequently, as the termination threshold increases, track fragmentation increases. Tracklets are then linked into complete tracks using manually-generated ground-truth tracks to avoid dependence on a particular linking algorithm for the purpose of this evaluation.

The study was conducted on video showing complex airplane refueling activities. The video has multiple movers, object appearance changes and occlusions lasting hundreds

Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

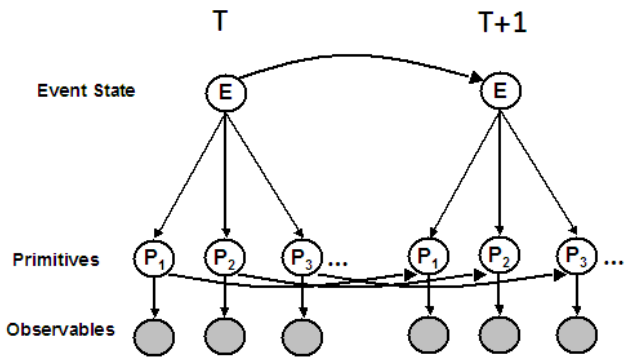

Figure 1. The DBN model. The event state is maintained in node E, the semantics of each event state are represented by the Primitives layer, and the observation density of each primitive is captured in the Observables layer.

of frames. Using mean-shift tracking [5] and linear interpolation across tracking gaps, we compute the event recognition score as a function of termination threshold.

Interestingly, the data reveals that event recognition performance improves as the termination threshold (and hence fragmentation) increases. At first glance, this seems surprising; as tracking performance degrades, event recognition performance actually improves. The explanation is that aggressive tracking (with a low termination threshold) leads to irreversible tracking errors that are particularly punitive for event recognition.

Finally, the data shows the robustness of our DBN method across a wide range of track fragmentation levels and long observation gaps. The next section details our event recognition method, followed by a description of the experiment and results.

2. Event Recognition

We are interested in the representation of complex events, where the semantics of the event are explicitly captured in a set of discrete states and in quantities defining those states. This type of model has proven to be effective in previous work [8, 12], particularly for generalization over a wide range of scene conditions without model modification [4].

An event model consists of a fixed set of actors, A_i, and a dynamical model D expressing how the actors interact over time. U is represented as a set of discrete states $U_1, U_2, ..., U_N$. For now, we assume these states are strictly ordered. The actors may be moving or stationary. There is a finite set $\{E_i\}$ of event models.

We use a Dynamic Bayesian Network (DBN) to represent the dynamical model U. DBN's generalize HMM's and relax some of the structural limitations of the HMM by allowing multiple and conditionally-dependent hidden nodes [11]. Furthermore, temporal dependence between arbitrary hidden nodes can also be included to model persistence. To account for uncertainty in observable data such as object position and velocity, multiple observation nodes can be included.

Our DBN model is shown in Figure 1. At each time slice (video frame), a new instance of the network is created and temporally linked to the previous one according to the temporal dependencies. The root node $E \in \{U_1, U_2, ..., U_N\}$ maintains the state estimate of the current time slice given the current and all previous observations.

The Observables layer contains observed nodes, one for each semantic primitive in the model as described below. The Primitives layer acts as a noise buffer between the observables and the (hidden) state, so that $\mathrm{P}(P|E)$, the prior conditional distribution of a primitive given an event state, is not directly tied to $\mathrm{P}(O|P)$, which models the distribution of a noisy observable given a primitive. Our model resembles the multi-observation HMM previously proposed [2] with the addition of the layer of primitives nodes. In our experiments, we have found that the model works well enough without the temporal links between nodes in the Primitives layer.

A significant advantage of our approach is that we apply semantic modeling as early in the data processing chain as possible, through the use of spatio-temporal semantic relations represented in the Primitives layer. Our DBN is based on these relations rather than on raw observables such as position. In previous work, we have demonstrated that this straightforward enhancement can provide surprising robustness w.r.t. to changes in viewpoint and scene configuration [4]. Our semantic primitives include relational predicates such as *CloseTo*, *ContainedIn* and *AppearNear*, *DisappearNear* and unary ones such as *Moving*.

The semantic primitives provide the link between actors and the event model. Instead of directly representing the actors in the DBN, the semantic primitives take the actors as arguments. This allows the event to be defined as a sequence of expected values of semantic primitives, as shown in Figure 2. The values in this table were specified initially by hand through observing one video sequence (T10 – see Fig. 3(a)). Note that the primitive *ContainedIn(Truck1, Driver1)* actually cannot be observed in our example video (see row 3 in Figure 2) and does not actually have a corresponding Observable node. Nevertheless, it's a primitive that is modeled and can be inferred from a video sequence.

A uniform distribution is inserted by default for nodes corresponding to entries in the table that are left unspecified. That is to say, the state of the unspecified primitives given the specific event state is assumed to be true or false with equal probability without any additional data.

With training data, the parameters can be updated via the

Primitives	transVeh1			exitVeh1		carry1		pump1	carry2		pump2	carry3		enterVeh1		transVeh2	
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
Close to Plane1 - Truck1	0	0	1													1	0
Moving - Truck1	0	1	1													1	1
Contained in Truck1 - Driver1	1	1	1	1	0									0	1	1	1
Appear near Truck1 - Driver1				0	1												
Moving - Driver1						1	1	0	1	1	0	1	1				
Close to Plane1L - Driver1				0	0	0	1	1						0	0		
Close to Plane1R - Driver1				0	0				0	1	1			0	0		
Close to Plane1 - Hose1						1	1	1	1	1	1	1	1				
Close to Plane1 - Fuel1								1			1						
Moving - Fuel1								1			1						
Close to Truck1 - Driver1												0	1				
Disappear near Truck1 - Driver1														0	1		

Figure 2. The dynamics of the event are defined by changes in the expected values of semantic primitives. The refueling model uses 9 binary primitives (rows) and 17 states (columns). The value in cell (i, j) indicates the expected value of primitive i in state j. An empty cell indicates that the model is indifferent to the value of primitive i for state j. The primitives that are shaded gray cannot be observed and are not used in the model.

standard expectation maximization method [11]. Here we use one video clip (T10) consisting of about 16000 frames. The additional training step made the model more discriminative w.r.t. other incidental events in the scene, as discussed below.

3. Experimental Results

Our goal here is to study the effect of fragmented tracks on model-based event recognition. Using video sequences showing refueling of an commercial aircraft, the event model is trained on the T10 sequence shown in Figure 3(a), and applied to the sequence T09 shown in Figure 3(b). The event involves a particular sequence of interactions between the driver, fuel truck, and airplane. Different levels of track fragmentation are introduced by varying the tracking termination threshold τ (see Figures 3(c) and (d)).

Captured from significantly different viewpoints, the T10 and T09 sequences, each with a duration of about 16,000 frames, contain complex refueling scenes with many moving objects, significant occlusions behind the aircraft and trucks, objects in close proximity, and erratic motion patterns. The common theme is the refueling event involving the airplanes, refueling trucks and truck drivers.

The goal is to recognize this sequence of actions (the refueling event) from the interactions between the fragmented tracks of the actors (airplane, truck and driver). There are tracking gaps of thousands of frames on some objects because of occlusion, tracking loss, and when moving objects come to a stop at times. The question is how this severe track fragmentation affects recognition performance.

The DBN for the refueling event has 17 states, defined by the expected changes in the values of the semantic primitives. The semantic primitives are observed based on computation performed on the tracks of the role objects. The model is trained on the sequence T10 based on the manually created ground truth tracks, and applied to the sequence T09.

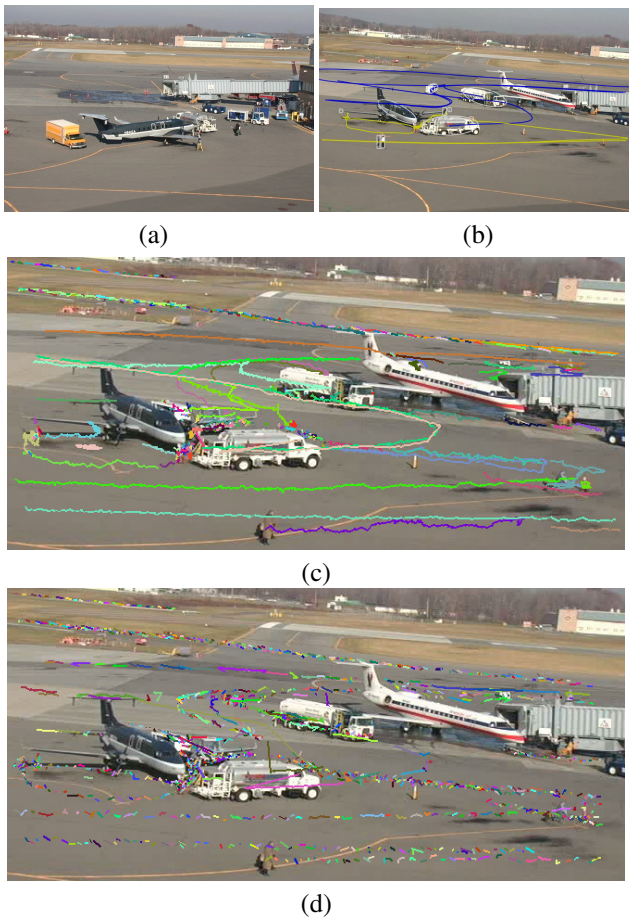

Figure 3. (a) Sequence T10 showing an airplane refueling event. (b) Sequence T09 of another refueling event, overlaid with manually created ground truth tracks. (c) and (d): T09 overlaid with tracklets from the mean-shift tracker with track termination thresholds at $\tau = 0.75$ and $\tau = 0.95$ respectively.

Moving objects are automatically detected in T09 using frame differencing and tracked by a mean-shift tracker. After a tracklet t_i is terminated (i.e., its lock score drops below τ), a new tracklet is created once the object is detected again. The tracklets corresponding to object O_j are linked together into a fragmented track $T_j = \{t_i\}$ by spatially and temporally associating them to the ground truth track G_j for O_j. After interpolating over the gaps in the fragmented tracks, they are then input to the event model E by assigning one track to each event actor A_k. Evaluating the model through the duration of the tracks yields a single log-likelihood $R = \log \mathrm{P}(E|\{A_k \leftarrow T_j\})$ indicating how well the assigned tracks $\{A_k \leftarrow T_j\}$ fit the event model.

For the model to be effective, the score of the true event must be distinguished from scores of the model when role

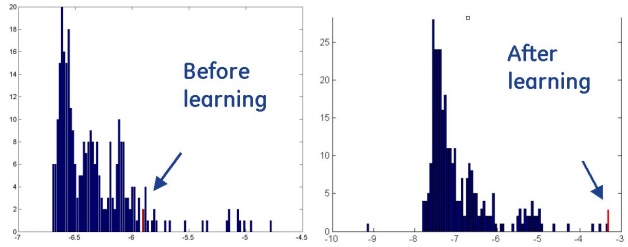

Figure 4. Histograms of event log-likelihood scores on the T09 sequence. Scores from random role assignments to the object tracks are in blue, and the correct role assignment is in red. The left histogram used manual parameter settings, and the right used settings learned on the T10 sequence. The correct assignment score is well separated from the others after learning (red bar on the far right).

assignments are incorrect. This is shown in the right plot in Figure 4 for fragmented tracks with $\tau = 0.75$ ($\tau \in [0, 1]$). A total of 303 different observation sequences were generated by assigning different actor roles to object tracks, including 300 random ones, and 3 hand-picked confusable ones. The event score highlighted by the red bar corresponds to the correct role assignment and was found to be well separated from the others. Only a small fraction of the non-event scores were close to the true event score, despite tracking gaps of up to 2000 frames on the true event.

The plot to the left in Figure 4 shows the corresponding result where the DBN model parameters were initially set manually on a best effort basis until the correct state transitions were achieved on the training sequence T10. However, the true event score is not well separated from other event scores. This demonstrates the effectiveness of the learning step in making the model more discriminative despite the change in scene viewpoint from T10 to T09.

The event recognition result on T09 with the correct role assignment is detailed in Figure 5. Rows 11 to 18 and rows 2 to 10 are the observables and primitives for the event model over time, respectively. Row 1 (bottom) shows the state transition over time, where each vertical line is a state transition and each horizontal line is the duration in one state. Random values were assigned to nodes in some time interval when no observations can be computed.

Next we examine event recognition performance R as a function of track fragmentation by varying τ. Fragmentation F is measured for each ground-truth track as N/L, where N is the number of tracklets corresponding to a ground-truth track and L is its duration (number of frames). At low values of τ, the tracker has low specificity and low fragmentation; it may remain on an object through partial occlusions and appearance changes, but may switch away from the object, yielding long but partially erroneous tracklets. At high values of τ, the tracker breaks more frequently,

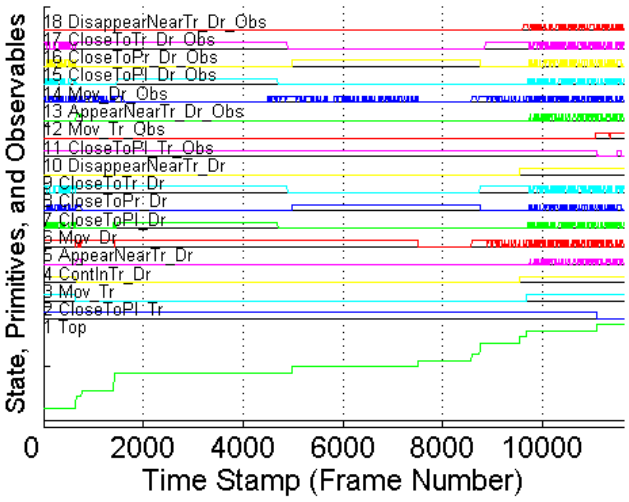

Figure 5. Recognition of the event transitions of the sequence T09.

generating many short tracklets with high specificity and high fragmentation, but fewer switching errors. Examples of the tracklets on T09 are shown in Figures 3 (c) and (d) for $\tau = 0.75$ and $\tau = 0.95$.

The relationship between τ and the mean of F over all 18 ground-truth tracks in T09 is shown in the blue plot in Figure 6, referenced to the left vertical scale. As expected, F increases with τ; the relationship is highly super-linear. Fragmentation for the driver (green) and truck (cyan) are also shown. The same figure illustrates the relationship between event score R and τ (red plot, right vertical scale). R is plotted using correct actor assignments while varying τ from 0.05 to 0.95. According to the data in Figure 4, $R > -4$ indicates reasonable recognition of the event. Therefore, we can see that recognition occurs consistently when $0.70 < \tau < 0.90$. (The score also happens to be high at $\tau = 0.55$ somewhat by chance.)

Somewhat surprisingly, R generally increases with τ and fragmentation. As increased fragmentation is generally viewed as a decrease in tracking performance [1], this implies that weaker tracking may be better for event recognition, at least when fragmentation is present and when track linking is reasonably accurate. The reason is that different types of tracking errors have different effects on event recognition. Switching errors, in which a track switches from one object to another, are particularly catastrophic for event recognition based on tracks since a switched track no longer follows the expected behavior for the event. As τ decreases, switching errors become more likely, leading to poor R. Conversely, detection errors are not as catastrophic, as long as the actor objects are tracked intermittently and opportunities to correct them via better linking exist. In the experiments presented here, tracklet linking were guided by

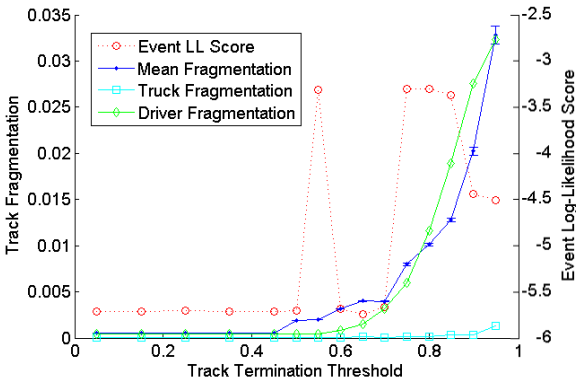

Figure 6. Fragmentation and event log-likelihood score R as functions of termination threshold τ. Fragmentation for the driver and all objects have very similar dependence on τ; the truck has a similar profile with much smaller scale. R increases dramatically when the driver fragmentation increases slightly at $\tau = 0.75$.

ground-truth tracks. The use of ground-truth does not actually lead to perfectly linked tracks because of inaccuracy in objection detection and reinitialization. Nevertheless, it avoids dependence of the results on a particular linking algorithm. We have also observed similar effects using the fully automatic linking from [10].

When fragmentation is extreme, i.e. $\tau > .85$ (see Figure 3(d)), R begin to decrease significantly. Although we have not studied this in detail, we expect that to be caused by a large increase in noise of track orientation and speed, and other errors as a result of frequent object reinitialization; track linking gets harder here even with ground truth guidance. The overall result also suggests that for realistic video where track fragmentation is likely, it would be beneficial to tune tracking algorithms jointly with event recognition algorithms to achieve higher overall recognition accuracy.

4. Conclusions

We have proposed a method for event recognition in the presence of large tracking gaps, and studied its performance as a function of track fragmentation. Although preliminary, the data reveals that optimizing for tracking performance does not necessarily lead to optimal event recognition. Instead, event recognition is best when tracking conservatively avoids switching errors. In related work [3], we exploit this finding to jointly perform track linking and event recognition simultaneously.

5 Acknowledgments

This report was prepared by GE GRC as an account of work sponsored by Lockheed Martin Corporation. Information contained in this report constitutes technical information which is the property of Lockheed Martin Corporation. Neither GE nor Lockheed Martin Corporation, nor any person acting on behalf of either; a. Makes any warranty or representation, expressed or implied, with respect to the use of any information contained in this report, or that the use of any information, apparatus, method, or process disclosed in this report may not infringe privately owned rights; or b. Assume any liabilities with respect to the use of, or for damages resulting from the use of, any information, apparatus, method or process disclosed in this report.

References

[1] J. Black, T. Ellis, and P. Rosin. A novel method for video tracking performance evaluation. In *PETS*, pages 125–132, 2003.

[2] M. Brand and V. Kettnaker. Discovery and segmentation of activities in video. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):844–851, August 2000.

[3] M. Chan, A. Hoogs, R. Bhotika, A. Perera, J. Schmiederer, and G. Doretto. Joint recognition of complex events and track matching. In *Proc. CVPR*, 2006.

[4] M. Chan, A. Hoogs, J. Schmiederer, and M. Petersen. Detecting rare events in video using semantic primitives with HMM. In *Proc. ICPR*, volume 4, pages 150–154, 2004.

[5] D. Comaniciu, V. Ramesh, and P. Meer. Real-time tracking of non-Rigid objects using mean shift. In *Proc. CVPR*, volume 2, pages 142–149, 2000.

[6] S. Gong and T. Xiang. Recognition of group activities using dynamic probabilistic networks. In *Proc. ICCV*, pages 742–749, 2003.

[7] A. Hakeem, Y. Sheikh, and M. Shah. CASE^E: a hierarchical event representation for the analysis of video. In *Proc. Natl. Conf. on AI*, pages 263–268, 2004.

[8] S. Hongeng, R. Nevatia, and F. Bremond. Video-based event recognition: activity representation and probabilistic recognition methods. *CVIU*, 96(2):129–162, November 2004.

[9] S. Intille and A. Bobick. A framework for recognizing multi-agent action from visual evidence. In *Proc. Natl. Conf. on AI*, pages 518–525, 1999.

[10] R. Kaucic, A. G. A. Perera, G. Brooksby, J. Kaufhold, and A. Hoogs. A unified framework for tracking through occlusions and across sensor gaps. In *Proc. CVPR*, pages 990–997, 2005.

[11] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference, and Learning*. PhD thesis, University of California Berkeley, 2002.

[12] S. Park and J. K. Aggarwal. Event semantics in two-person interactions. In *Proc. ICPR*, volume 4, pages 227–230, 2004.

[13] C. Stauffer and E. Grimson. Learning patterns of activity using real-Time tracking. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):747–757, Aug 2000.

[14] Y. Zhou and H. Tao. A background layer model for object tracking through occlusion. In *Proc. ICCV*, pages 1079–1085, 2003.