Joint Recognition of Complex Events and Track Matching

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

Abstract

We present a novel method for jointly performing recognition of complex events and linking fragmented tracks into coherent, long-duration tracks. Many event recognition methods require highly accurate tracking, and may fail when tracks corresponding to event actors are fragmented or partially missing. However, these conditions occur frequently from occlusions, traffic and tracking errors. Recently, methods have been proposed for linking track fragments from multiple objects under these difficult conditions. Here, we develop a method for solving these two problems jointly. A hypothesized event model, represented as a Dynamic Bayes Net, supplies data-driven constraints on the likelihood of proposed track fragment matches. These event-guided constraints are combined with appearance and kinematic constraints used in the previous track linking formulation. The result is the most likely track linking solution given the event model, and the highest event score given all of the track fragments. The event model with the highest score is determined to have occurred, if the score exceeds a threshold. Results demonstrated on a busy scene of airplane servicing activities, where many non-event movers and long fragmented tracks are present, show the promise of the approach to solving the joint problem.

1. Introduction

Many algorithms for recognizing events or activities depend on tracking event actors correctly [7, 16, 8, 13, 5]. But assumptions of robust tracking are problematic under real-world conditions such as busy scenes, full and partial occlusions, shadows, and so on. Frequently, objects involved in an event are tracked intermittently, with no preservation of identity across track fragments corresponding to the same object. Recent work attempts to address these tracking issues by linking track fragments together in a multi-object framework [9] or by background layers [18].

We propose a novel algorithm that solves event recognition and track linking jointly. Rather than assuming complete tracks for event actors, we dynamically determine the global track linking solution over the entire scene and an arbitrary temporal window that maximizes the likelihood of a hypothesized event occurrence. This optimization is performed online as objects are tracked, resulting in both the best estimate of which event is occurring (if any), and how all track fragments in the scene should be linked given that event. The tracking formulation follows the track matching paradigm in [9] enhanced with constraints from the event model. Events are modeled using Dynamic Bayesian Networks, with observed nodes corresponding to spatio-temporal, semantic relations between event actors and elements as in [7].

The primary advantage of our method is that events can be recognized despite highly fragmented tracking due to long occlusions, in scenes with many non-involved movers, under different scene viewpoints and/or configurations. An example scene and event is shown in Figure 1. In our experiments, event actors are not tracked for hundreds of frames at 30Hz. Non-actors move through the same areas as the actors, often at the same time. Both of these problems are addressed by the coupling of event recognition and track linking.

Furthermore, we focus on complex events, in the sense that events are defined by the interactions of multiple objects, through multiple states. In these cases the analysis of a single track is insufficient to characterize the event. In order to recognize the same complex event in different scene configurations, we model the invariant, underlying semantics of the event through discrete, semantic state models constructed on relational semantic primitives, rather than direct measurements such as position and orientation that vary from scene to scene.

A complementary advantage is that tracking is also improved. Instead of being an independent process, tracking

0-7695-2597-0/06 $20.00 © 2006 IEEE

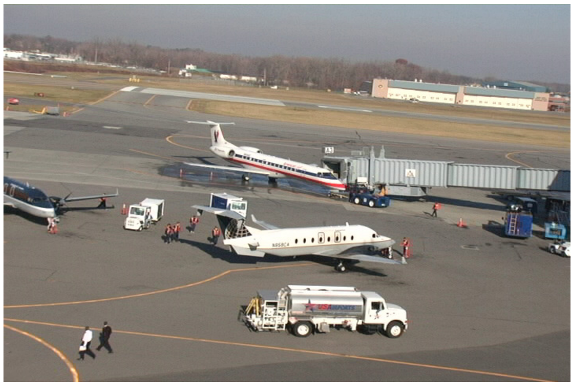


Figure 1. An airplane refueling event. The scene contains many moving objects, significant occlusions behind the aircraft and trucks, objects in close proximity, and erratic motion patterns.

is embedded in the context of the hypothesized event. This context provides resolution of ambiguity that is very difficult or impossible otherwise, particularly in busy scenes. Consider a situation where appearance information is not sufficient to discriminate between a set of individuals. Multiple people are tracked initially, then some become occluded, then some reappear one at a time. How do we associate the reappearing tracks to those we saw before? In simple cases, kinematic prediction might be sufficient. In complex cases, where people are occluded for long periods but short spatial distances, their behavior both before and after the occlusion may be an important clue for identity association. Our event models provide expectations on those behaviors, which are then used to constrain track association.

One potential concern about this approach is that when a hypothesized event is not occurring, event-guided track linking could "hallucinate" the event by creating incorrect links to fit the event model. Although this is possible, for complex events it is very unlikely. As the number of model states and actors increases, the probability of linking incidental track fragments into plausible actor tracks rapidly decreases. We have not observed these problems in our experiments, although our experimental validation is not yet extensive.

There are many approaches to event, activity and behavior recognition. Our focus is on detecting modeled events, for which little or no training data may be available. Hence our methods are appropriate for applications where interesting events are rare and can be specified in semantic terms. When sufficient training data is available, statistical methods can be used to learn normal behavior and detect deviations [16, 6, 13], although explaining abnormal behavior may be performed through explicit models. Of particular relevance is the work of Gong and Xiang [4, 17], because they recognize complex activities without explicit tracking. Instead, complex behavior patterns of interacting objects are learned directly from moving object detections in order to avoid difficulties from track fragmentation. This work is a compelling example of what can be done without tracking, although ultimately tracking is required for scaling to busy scenes, viewpoint and configuration changes, and complete scene understanding.

Results are demonstrated on a very challenging scenario, refueling airplanes. The primary actors in the event are occluded or otherwise not tracked for thousands of frames. Non-involved people and vehicles move through the scene during refueling, and appearance is a weak linking cue for most of the people. We show that refueling events can be distinguished from other activity under these conditions, and that tracking performance is improved by event-guided linking vs. context-free linking.

Although we directly address track fragmentation and long tracking gaps, we do not yet handle other tracking issues such as merged objects, completely missing actor tracks, and partially erroneous tracks that correspond to different objects at different times. The last issue is virtually eliminated by the track linking framework, which allows the threshold for track termination to be set at a very sensitive level.

The next section summarizes the track linking formulation from [9]. The third section describes our event modeling approach, followed by the main contribution of the paper in combining track linking and event recognition. Results are then discussed followed by conclusions.

2. Track Linking

The fundamental challenge addressed by track linking is to maintain object identity through significant track fragmentation caused by long occlusions, shadows, pose changes, nearby moving objects and other tracking problems. These conditions can cause the track of an object to be lost. Another track on the same object may then be created, and we wish to link these "tracklets" together into a single track for the object. When multiple moving objects are close together, this problem becomes much more difficult, particularly in the presence of false tracks from motion segmentation errors.

Track linking integrates a set of relatively independent modules to address each of these problems. On each frame, a moving object detector detects all movers in the video. For each mover that is not already being tracked, a single object tracker is initiated to form a tracklet. When a tracklet's lock score drops below a threshold, the tracklet is terminated. Periodically, the track matching module makes a global assignment of tracklets into tracks, considering all tracklets within a specified time window – new tracklets, previously lost tracklets (called *suspended* tracklets), and previously linked tracklets (their link associations may be changed).

Ideally, the output is a single linked track per object.

For moving object detection, we use either frame differencing or GMM background modeling [15]. For single-object tracking, we use mean-shift [3]. In [9] scene understanding was used to assist in occlusion reasoning. Although we expect it would help in event recognition, we have not yet considered it in our work.

The track matching module jointly associates all suspended tracklets with all active tracklets or no tracklet if no suitable match is found. Denoting the set of suspended (source) tracklets as $S = \{S_1, S_2, S_3, ..., S_m\}$ and the active (destination) tracklets as $D = \{D_1, D_2, D_3, ..., D_n\}$, then a matching configuration is $C_k \subseteq S \times D'$, where D' is D extended with a null track. The goal is to find

$$C_{\text{opt}} = \arg\max_{C_k} \mathrm{P}(C_k \,|\, \text{tracklet data}). \qquad (1)$$

Following [14], an association matrix M is formed where the elements t_{ij} represent the probability that source i corresponds to destination j and t_{i0} corresponds to the null match,

$$M = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1n} & t_{10} \\ t_{21} & t_{22} & \cdots & t_{2n} & t_{20} \\ \vdots & \vdots & & \vdots & \vdots \\ t_{m1} & t_{m2} & \cdots & t_{mn} & t_{m0} \end{bmatrix}. \qquad (2)$$

Assuming that tracklet pairings are conditionally independent with the constraint that no two source tracklets can match to the same destination tracklet, the optimal assignment is

$$C_{\text{opt}} = \arg\max_{C_k} \prod_{(i,j)\in C_k} t_{ij} \qquad (3)$$

which is computed using the Hungarian algorithm [10].

To compute the probability t_{ij} that a source tracklet corresponds to a destination tracklet, the probabilities of kinematics $\tilde{x}$ and appearance $\tilde{a}$ are combined,

$$t_{ij} \propto p(\tilde{x}|S_i \rightarrow D_j)p(\tilde{a}|S_i \rightarrow D_j).$$

Kinematic prediction follows a constant-state assumption, and appearance matching is performed on color histograms accumulated over each tracklet.

Here, we extend the previous formulation to include event constraints as described below.

3. 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 the quantities defining those states. This type of model has proven to be effective in previous work by various groups [7, 12], particularly for generalization over a wide range of scene conditions without model modification [2].

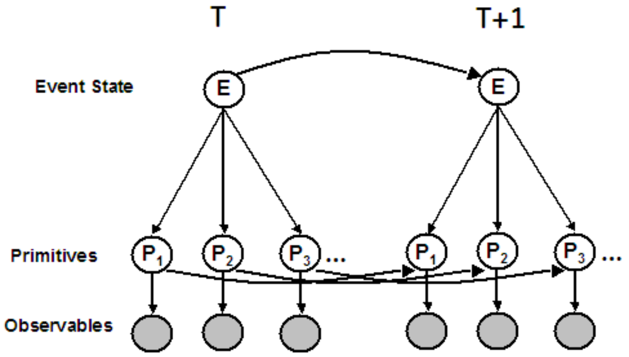


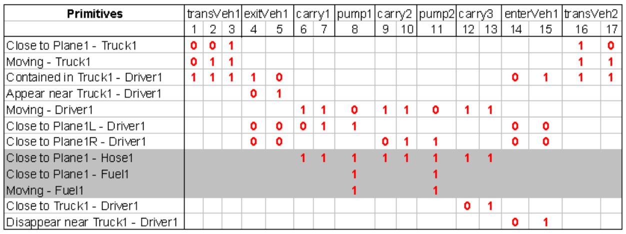
Figure 2. The DBN event model. Event state is maintained in node E, the semantics of each event state are represented by the Primitive layer, and the observation density of each primitive is captured in the Observables layer.

An event model consists of a fixed set of actors, A_i, and a dynamical model U 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.

To account for uncertainty in observable data such as object position and velocity, we use a Dynamic Bayes Net (DBN). DBN's generalize HMM's, relaxing some of the limitations of the latter by allowing temporal dependence between arbitrary nodes to model persistence. Our DBN model is shown in Figure 2. 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 (for which a direct observation is possible). Our model resembles the multi-observation HMM previously proposed [1] 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.

One of the significant advantages 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. Our DBN is based on these relations rather than on raw observables such as position. In previous



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 3. 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 in our video, and are not used in the model.

work, we have demonstrated that this intuitive enhancement can provide surprising robustness with respect to changes in viewpoint and scene configuration in recognizing complex multi-stage events [2]. The 17 states here are actually logically grouped into 9 subevents, each of which can be reused to compose other event models. Our semantic primitives here are constructed with relational predicates such as *CloseTo*, *ContainedIn*, *AppearNear*, *DisappearNear* and unary predicates 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 3. The values in this table were specified initially by hand through observing an example sequence (T10). Note that the primitive *ContainedIn(Truck1, Driver1)* actually cannot be observed in our example video (see row 3 in Figure 3) 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.

The event model contains a number of parameters, which can be learned or manually specified. We used a combination, as we desire to train on at most one sequence and generalize to others. 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. After manual initialization of the parameters, a training step was performed by a standard expectation maximization method for DBN's [11], using the given example sequence (T10) to update all parameters.

A major limitation of this baseline approach is that it assumes complete tracks for the event actors, so that the observed semantic relations are always present. In the next section we describe how we combine event recognition and track linking to resolve this problem.

4. Jointly Recognizing Events and Linking Tracks

We formulate our method for event recognition under fragmented tracking, as a joint solution to event recognition and track linking. Given a set of event models and a video sequence, our goal is to find all instances of events that have occurred. In the video, a number of tracklets **T**=$\{T_j\}$ are observed over time. Each actor $A_i \in \mathbf{A}$ in event instance E corresponds to a temporal sequence of zero or more tracklets, but any tracklet T_j may not correspond to any actor (a *clutter* tracklet). For simplicity, we assume that each tracklet may be assigned to at most one actor (*i.e.* we defer the problem of handling merged tracks), and that each actor is involved in only one event.

Then, the problem is to find a labeling L of **T**, where each tracklet T_j is assigned to one actor A_i or to a clutter track, such that the event likelihood is maximized. L can be written

$$L = \{(l_1, T_1), (l_2, T_2), ..., (l_n, T_n)\} \tag{4}$$

where $n = ||\mathbf{T}||$ is the number of tracklets. Defining $m = ||\mathbf{A}||$ as the number of actors, then $l_j = i$ for $i \leq m$ indicates that T_j is assigned to A_i. If $l_j > m$, then T_j is assigned to a clutter track $G_i \in \mathbf{G}$.

The goal is to find the labeling that maximizes the probability of the event,

$$L_{\text{opt}} = \arg\max_{L} \mathrm{P}(E|\mathbf{T}, L). \tag{5}$$

A complete solution to this requires a search over all labeling possibilities, which is combinatorial.

However, we can cast the problem as a track matching problem, guided by the event model, with a few additional constraints to make it tractable. In addition, we extend the problem to include track matching for all tracks (*i.e.* including clutter tracks), which adds little computational cost and should increase robustness when clutter and actor tracks are close together.

Continuing the notation of track linking, our combined event-track matching configuration is $C_E \subseteq (\mathbf{A} \cup \mathbf{G}) \times \mathbf{T}$. This can be expressed as a $(m + ||\mathbf{G}||) \times (n + 1)$ cost matrix M_E analogous to Eq.2, except the first m rows are designated actor tracks.

Once the assignment problem involving M_E is solved using the Hungarian algorithm as before, *i.e.* the lowest cost path is found with exactly one entry in each row and column (except for the null column), then L can be extracted directly, as the selected column j in each row i provides the value $l_j = i$. But how does this help us find the labeling that maximizes $\mathrm{P}(E|\mathbf{T}, L)$?

0-7695-2597-0/06 $20.00 © 2006 IEEE

To incorporate event constraints into M_E, we extend the computation of the cost t_{ij} of matching source tracklet S_i to destination tracklet D_j, given by Eq. 2, to

$$t_{ij} \propto \mathrm{P}(\tilde{x}|S_i \rightarrow D_j)\mathrm{P}(\tilde{a}|S_i \rightarrow D_j)\mathrm{P}(E|S_i \rightarrow D_j)^{\alpha}. \tag{6}$$

$\mathrm{P}(E|S_i \rightarrow D_j)$ is the probability of the event model given the hypothesized match, and α is a weighting factor (a multiplicative factor in the log-likelihood space) that governs the influence of the event model on the solution. It is computed by assuming the match, spatially and temporally interpolating across the gap between the tracklets, evaluating the DBN through the (temporal) end of D_j, then taking the average log-likelihood. Since this computation can be expensive, we first filter the t_{ij} entries by kinematics and appearance.

The assignment solution with the cost matrix M_E takes into account the event model and its quantified preferences for track matches. Since this is incorporated directly into track matching, clutter tracks that may interfere with the event are considered together with actor tracks, and the solution should optimize both event likelihood and track matching given the event.

We can now summarize the approach. We perform a brute force search over both initial role assignments and event models. Specifically, the system is initialized by selecting an event model, assigning a tracklet to each of its actors, and then performing concurrent event-guided linking and DBN evaluation. Linking is performed whenever an actor tracklet ends, while DBN evaluation occurs at each frame and is performed using the forward-backward algorithm [11]. An event is detected when its average log-likelihood score exceeds a threshold.

However, we shall note that the M_E formulation violates the pairwise conditional independence assumption required to solve M_E efficiently. Because of $\mathrm{P}(E|S_i \rightarrow D_j)$, the score of one match may depend on other matches since all actor tracks contribute to the event model. We resolve this by restricting M_E to have exactly one actor tracklet at a time. Since M_E is evaluated whenever an actor tracklet ends, this implies that at most one actor tracklet can end in any frame. We have not encountered exceptions to this condition, but a simple workaround would be to extend one of the actor tracklets by a frame or two. Similarly, we cannot allow the previous assignment of a tracklet to an actor track to be revised (this is not true for clutter tracks, which can be re-matched at any later point; see [9]).

Addressing these problems, and improving upon the brute-force search, which does not scale well with the number of event models and actors, is deferred to future work.

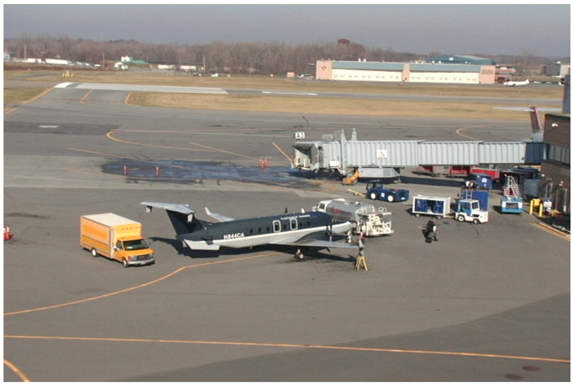


Figure 4. A second example of airplane refueling (T10). This scene has the same challenges as the one in Figure 1 (T09). The two scenes differ considerably in viewpoint, relative object positions, temporal spacing of actor occlusions and clutter tracks.

5. Experiments and Results

Our experiments were conducted on video sequences showing refueling of commercial aircraft as shown in Figures 4 and 1. In these videos, which have significantly different viewpoints, many non-involved movers are present and they move near the airplanes, refueling trucks and people involved in the event. We demonstrate successful recognition of refueling activities under these conditions, using a DBN event model with 17 states and 9 semantic primitive relations. After parameters were tuned based on T10, no adjustments were made for T09. First, we validate the approach on manual tracks with increasing levels of fragmentation. Then computed tracklets are used, and recognition scores are compared to non-events to show model discrimination, and to independently linked tracks to show the benefit of event-guided linking.

In both scenes, the duration of the refueling events is about 16,000 frames. In each event, a fuel truck drives to the front of the plane. The driver gets out, retrieves a hose and a ladder from the back of the truck, and walks to one wingtip. He ascends the ladder, inserts the hose into the wing, and pauses for many seconds. He then removes the hose, descends the ladder, walks to the other wingtip with the hose (passing by the truck), and repeats the refueling. He then returns to the truck, pulls in the hose, gets in the truck and drives off.

This sequence of sub-events is modeled by 17 DBN states defined by changes in the observation likelihoods of the semantic primitives. In our model, the roles are the driver/refueler, the refueling truck, the two wingtips, and the plane itself. The semantic primitives are computed using the tracks of these role objects as described below. We manually selected the plane and wingtip locations, but these could be automatically identified, particularly if the plane is

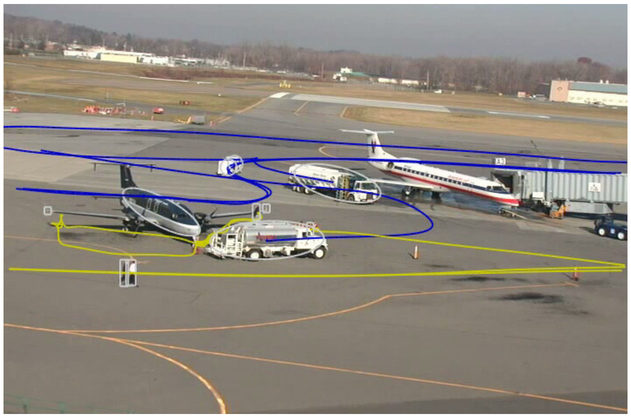


Figure 5. Manually-created tracks overlaid on one frame of the T09 video. Note the proximity of various objects, and the complex nature of the trajectories in the event.

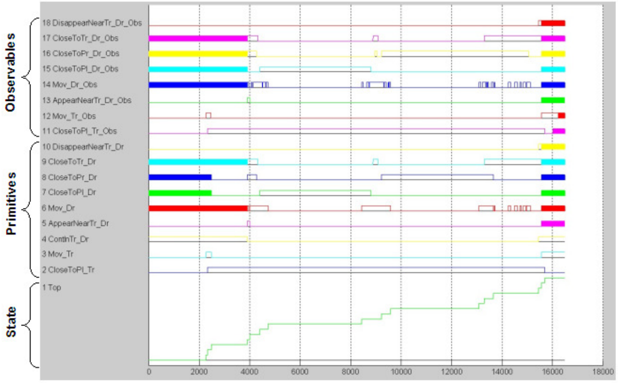


Figure 6. Recognition of event model over 16,000 frames, using manual tracks. The average log-likelihood score is -2.1.

observed taxiing into its parked position (our videos do not include this).

5.1. Validation using Manual Tracks

To investigate the performance of our system under ideal conditions, and for track evaluation, we manually created ground-truth tracks for all moving objects in the two refueling scenes (see Figure 5 for one example). This ground truth includes the user's guess at object positions when they are occluded.

The highest event score should be achieved when the correct role assignments are made on these ideal tracks. No track linking is required for this experiment. Figure 6 shows this case, and details the evolution of the observed and inferred values of the DBN nodes. The green curve at the bottom shows the state transitions (vertical edges) through the model. These transitions closely correspond to changes in the observed and primitive nodes, according to the expected values shown in Figure 3. Note that the primitive nodes roughly follow the observed nodes after accounting for noise in the data. For example, Primitive node 6 made a smoothed interpretation of the Observable node 14 between frame 8000 to frame 10000 when the driver moved from one side of the plane to the other.

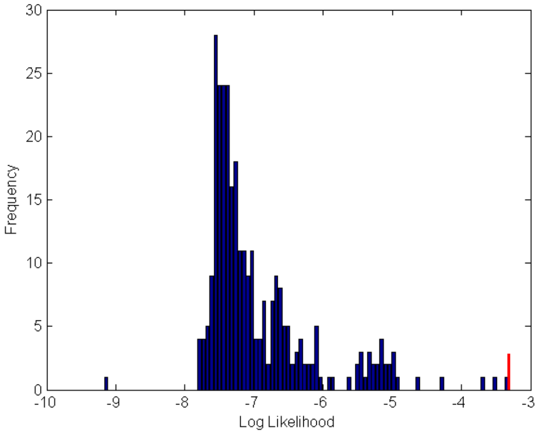


Figure 7. Histogram of log-likelihood scores calculated on observation sequences derived from the T09 video. 303 different observation sequences were generated by assigning different actor roles to object tracks. The 303 role assignments include 300 random ones, and 3 hand-picked ones that were meant to be confusable. The red bar corresponds to the correct role assignment and was found to be well separated from the others.

For the model to be effective, the score of the true event must be distinguished from scores of the model when role assignments are incorrect. This is shown in Figure 7, again for ideal tracks. Only a small fraction of the non-event scores were close to the true event score. Although the model was tuned on T10, no parameter adjustments were made for T09.

Since we are particularly interested in how event recognition performs under high levels of fragmentation, we systematically fragmented the manual tracks by deleting h frames, then keeping k frames, deleting h again, and so on. Tracks were then linked by associating them to ground truth.

The results are shown in Figure 8. As the tracking gap length increases from 0 to 900 frames, recognition performance on T10 remains relatively consistent. Performance on T09 is expectedly lower than T10, but remains reasonable until the gap length exceeds 400 frames. As shown in Figure 7, most non-event scores are below -5 on T09.

5.2. Results on Computed Tracks

With a good understanding of the behavior of our model on ideal tracks, we now turn our attention to the much more difficult problem of recognition on computed tracks. On

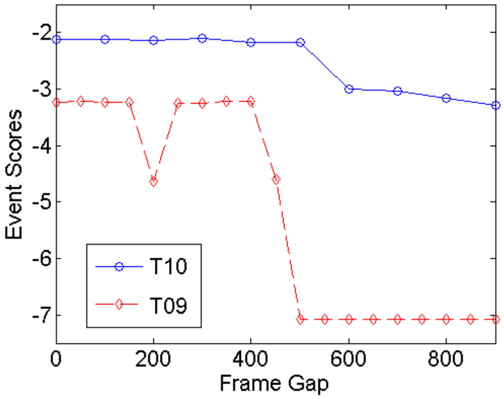


Figure 8. Results of recognizing the refueling event on manual tracks, with increasing levels of fragmentation. The horizontal axis is the length of track gaps, and the vertical axis is the event average log-likelihood. T10 is the sequence used for training. Reasonable recognition performance is maintained on sequence T09 over track gaps up to 400 frames.

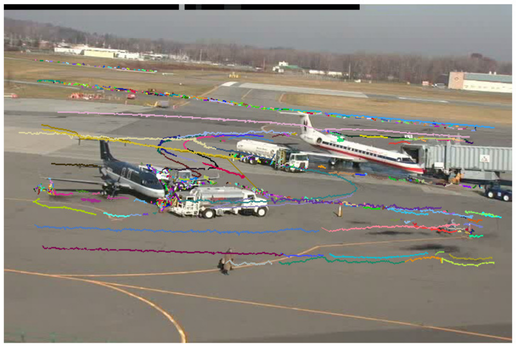


Figure 9. Tracklets produced by mean-shift tracking with automatic initialization on sequence T09.

both of these videos, the tracker creates very high fragmentation, with gaps of thousands of frames on some objects because of occlusion, lack of motion and tracking loss. Results of tracking (without linking) are shown in Figure 9 for T09. The driver is fragmented into dozens of tracklets, as are many of the other movers. This tracking was performed with high sensitivity and a low track termination threshold, which is suitable for subsequent linking.

Independent track linking, without the event model, should still improve the situation, and hence it serves as an interesting result in its own right and a baseline for comparing event-guided linking. Attempting to recognize the event without any sort of linking proved to be hopeless, as none of the actor tracks last for a reasonable fraction of the event duration (even with parameters tuned for this as opposed to linking). With actor track fragments correctly assigned, the resulting model scores were all below -7, *i.e.* no recognition.

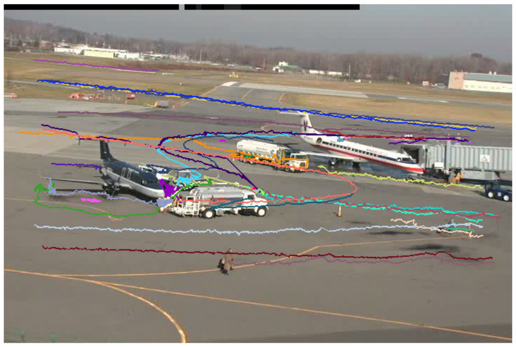


Figure 10. Results of independent track linking, *i.e.* no use of the event model, on sequence T09. The event recognition scores are poor because the actor tracks are still partially fragmented. For example, the driver/refueler is fragmented into the purple, green and violet tracks near the front and right wing of the plane.

Results of independent track linking are shown in Figure 10 on T09. Event recognition results were computed using these tracks, with interpolation across the gaps. Track linking is partially correct, but it divides the primary actor (driver) into three tracks. To examine whether recognition is still possible, we ran separate tests for each of the three driver tracks, with the truck assigned to its ground truth track. This raises the score somewhat artificially, but provides a harder comparison for event-guided linking.

The resulting scores are -5.2, -4.7 and -4.2 for the purple, green and violet tracks respectively in Figure 10. Comparing to the scores in Figure 7, these numbers indicate partial recognition – which is quite reasonable, as the driver track is partially correct (and the other actor tracks are completely correct, although much less complicated).

If the three tracks are manually linked into a single track comprising most of the driver's trajectory, the score increases to -3.0. This score is particularly significant, because it is an upper bound given the driver tracklets. Recognition using event-guided linking cannot do better given the same tracklets.

Finally, we discuss preliminary results on joint event recognition and linking. For computational efficiency, these were computed on tracks partially linked by independent track linking. This is similar to the hypothesis filtering step before evaluating $\mathrm{P}(E|S_i \rightarrow D_j)$, in the sense that it allows conservative linking without event guidance. More aggressive track linking would have resulted in incorrect track matches that could not be undone. On the T09 video (not used for learning), event-guided linking is able to correctly piece together the complete tracks of the driver and the fuel truck. For example, it correctly links the purple, green and violet tracklets corresponding to the driver in Figure 10. The resulting event score indicates reasonable recognition performance.

Although our experimental validation is limited at this point, these preliminary results indicate that the method has significant potential to address very challenging problems in tracking and event recognition.

6. Conclusions

We have addressed the challenging problem of recognizing complex events involving multiple interacting objects, in busy scenes with long occlusions. The primary contribution is the combination of track linking with event recognition in a joint formulation that optimizes both simultaneously. This integration enables robust event recognition despite occlusions of hundreds or thousands of frames on the actor tracks, in the presence of clutter tracks. We also demonstrate robustness with respect to changes in viewpoint and scene conditions through the use of semantic modeling in a DBN framework. More evaluation experiments with additional video sequences are underway in order to characterize the performance of the approach more completely. Overall, the joint formulation has a great potential in solving tracking and event recognition problems in real-world conditions.

7. Acknowledgments

We want to thank Zhaohui Sun and Wensheng Hu for providing implementation assistance for some of the experiments conducted for this paper. 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] 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.

[2] 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.

[3] 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.

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

[5] 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.

[6] R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, and G. Coleman. Detection and explanation of anomalous activities: Representing activities as bags of event n-grams. In *Proc. CVPR*, June 2005.

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

[8] 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.

[9] 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.

[10] J. Munkres. Algorithms for assignment and transportation problems. *J. SIAM*, 5:32–38, 1957.

[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] Y. Shi, Y. Huang, D. Minnen, A. Bobick, and I. Essa. Propagation networks for recognition of partially ordered sequential action. In *Proc. CVPR*, volume 2, pages 862–869, 2004.

[14] C. Stauffer. Estimating tracking sources and sinks. In *Proceedings of the IEEE Workshop on Event Mining in Video*, 2003.

[15] C. Stauffer and E. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. CVPR*, pages 246–252, 1999.

[16] 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.

[17] T. Xiang and S. Gong. Video behaviour profiling and abnormality detection without manual labelling. In *Proc. ICCV*, volume 2, pages 1238–1245, 2005.

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