Large-Scale Event Detection Using Semi-Hidden Markov Models [*]

Somboon Hongeng and Ramakant Nevatia
Institute for Robotics and Intelligent Systems
University of Southern California
Los Angeles, California 90089-0273
{hongeng, nevatia}@iris.usc.edu

Abstract

We present a new approach to recognizing events in videos. We first detect and track moving objects in the scene. Based on the shape and motion properties of these objects, we infer probabilities of primitive events frame-by-frame by using Bayesian networks. Composite events, consisting of multiple primitive events, over extended periods of time are analyzed by using a hidden, semi-Markov finite state model. This results in more reliable event segmentation compared to the use of standard HMMs in noisy video sequences at the cost of some increase in computational complexity. We describe our approach to reducing this complexity. We demonstrate the effectiveness of our algorithm using both real-world and perturbed data.

1 Introduction

Recognizing events taking place in a video stream is of key importance in many applications such as video surveillance, video indexing, video annotation and video summarization. Event recognition is difficult as there is a huge gap between video signal data and the event concepts and the mapping between the two is not one-to-one. To bridge this gap, it is common to first detect moving objects and make event interpretations based on their trajectories (and shapes).

Some events, such as an object moving towards another, can be inferred directly from the motion trajectory at each frame; we term these *primitive* events. More complex events consist of a sequence of primitive events; we call such events as being *composite* events. Composite events defined by a single agent can be ordered in time; we call these as being *single threaded* where as events involving multiple agents may be *multiple threaded*.

Many systems have been proposed to infer primitive systems. Static Bayesian networks (BNs) are suitable tools for mapping from a variety of numerical visual properties to event concepts and have been used in previous work [2, 1, 6, 3]. However, BNs are not suitable for segmenting temporal sequences of unknown component durations needed for composite event recognition. Hidden Markov models (HMMs) have become a method of choice to model and segment continuous movements into a predefined number of states [11, 10, 4, 8]. These systems also attempt to recognize an entire sequence as being one event rather than segmenting the sequence into intervals corresponding to different composite events.

One of the potential weaknesses of HMM models is that, the first-order Markov assumption that the probabilities of the transition to the next states at time $t+1$ depend only on the state at time t implies that the probability of an event state being observed for a certain interval of time declines exponentially with the length of the interval. This may be a good model for speech recognition but is not realistic for visual events where the duration of a sub-events in the same event may vary from being very short to very long (for example, the time that a person takes to walk towards an object before picking it up).

In this paper, we present an augmentation of the HMM model where the *a priori* duration of the event states are explicitly modeled and incorporated into the finite-state automaton (FSA) to better approximate the reality of visual events. Such models are known as semi-HMMs (semi Markov, not semi Hidden). Inferences in semi-HMMs are more expensive: complexity is $O(NT^2)$, where N is the number of states and T is the length of the video in frames, compared to $O(NT)$ for a standard HMM [7]. In this paper, we develop an algorithm that reduces the complexity back to $O(NT)$ by taking advantage of simplifications resulting from assuming that the duration distributions are either uniform or normal. We demonstrate advantages of the new model on some real video examples.

[*] This research was supported in part by the Advanced Research and Development Activity of the U.S. Government under contract No. MDA-908-00-C-0036.

Proceedings of the Ninth IEEE International Conference on Computer Vision (ICCV 2003) 2-Volume Set
0-7695-1950-4/03 $17.00 © 2003 IEEE

2 System Overview

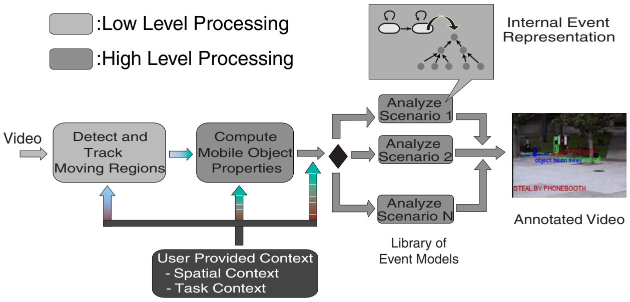


Figure 1: *Overview of the system*

Our approach is shown schematically in Figure 1. **Context** consists of associated information that is useful for object tracking and activity recognition such as a spatial map (spatial context) and prior activity expectation (task context). First, moving objects are detected and tracked and then some properties of the trajectories and shapes of moving objects are computed. These properties are used to infer the probability of potential primitive events defined in a library of scenario event models by a Bayesian network as well as classification of moving objects (for example, into humans or vehicles). The conditional probability distributions (CPDs) assigned to the links in the networks are assumed to be Gaussian, whose parameters (μ, σ) can be estimated from a training set of video samples directly due to the transparent nature of the models.

The primitive event probabilities are then used to infer probabilities of composite single thread events which are represented as FSA with semi-HMM properties. Temporal interval logic logical relations are used to compute probabilities of multiple agent, multi-thread events. In this paper, we focus on the modeling and recognition of single-thread events. Multiple-thread events can be computed from these as in our previously described approach [5].

The output of the system can be the actions performed by each actor or the global events that the actors participate in, with the most likely segmentation (i.e. the start and end times) of these events. A textual description of the video contents may be produced for various applications such as Video Annotation.

Figure 2 shows a representation of the composite event *"a car avoids the checkpoint"*, which is analyzed in results shown later. It is modeled by an FSA, consisting of an initial state S_0, and three primitive event states: *"approach checkpoint*, *"stop short before arriving"*, and *"leave"*. The arrows of the FSA indicate probabilistic transitions among event states. Primitive events are modeled by a BN of sub-events and mobile object properties defined at a lower level. Using Bayes' rule, the probabilities of primitive events are computed, which are then used for inferring the composite event.

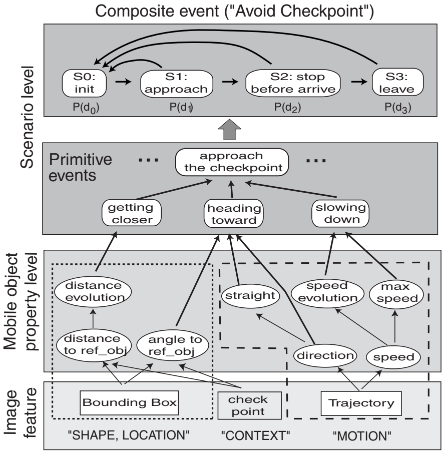


Figure 2: *Event representation of "avoid checkpoint".*

3 Composite Event Recognition

3.1 Probabilities of Composite Events

Let a multi-state composite event k, ${}^kMS$, be composed of the initial state ${}^kS_0$ and N event states ${}^kS_1, {}^kS_2, ..., {}^kS_N$. Let $O^{<1,t>}$ be the set of observations during time frames 1 to t, ${}^kS_i{}^t$ be the fact that ${}^kS_i$ occurs at frame t, and ${}^kMS_i^t$ be the fact that the sequence of states ${}^kS_1, ..., {}^kS_i$ has occurred with ${}^kS_i$ being the current state of at frame t. ${}^kMS$ is recognized at frame T by computing $P({}^kMS_N^T|O^{<1,T>})$. We drop the superscript k in the following for clarity. Now,

$$
\begin{aligned}
P(MS_N^T|O^{<1,T>}) &= \alpha_0 P(MS_N^T) P(O^{<1,T>}|MS_N^T) \cdots \text{(a)} \\
= \alpha_0 \sum_{\forall(t_1,t_2,\ldots,t_N)} & \overbrace{P(MS_0^{(t_1-1)})P(O^{<1,t_1-1>}|MS_0^{(t_1-1)})} \\
& a_{1,0}P(d_1 = t_2 - t_1)P(\widehat{O_1}|\widehat{S_1})a_{2,1}\ldots \\
& a_{N,N-1}P(d_N = T - t_N)P(\widehat{O_N}|S_N^{<t_N,T>}) \quad \cdots \text{(b)},
\end{aligned} \tag{1}
$$

where $\alpha_0 = \frac{1}{P(O^{<1,T>})}$ is a normalizing constant, t_i is the time at which the transition to S_i occurs, and $\widehat{S}_i = S_i^{<t_i,t_{i+1}-1>}$, which means that S_i occurs during t_i and t_{i+1} - 1. We write $\widehat{O_i}$ as shorthand for $O^{<t_i,t_{i+1}-1>}$.

Eq. 1 (a) is derived by Bayes' rule and can be computed as the summation of all possible event segmentation $t_1, \ldots, t_N$. Under semi-HMM assumption that $P(S_i^{t_i}|\widehat{S_{i-1}S_{i-2}...}) = P(S_i^{t_i}|\widehat{S_{i-1}})$, $P(MS_N^T)$ can be written as a product of $P(MS_0^{(t_1-1)})$ and $a_{i+1,i}P(d_i = t_{i+1} - t_i), \forall i = 1, \ldots, N$, where $P(d_i)$ is the duration probability of S_i and $a_{i+1,i}$ is the probability of the path from S_i to S_{i+1} normalized by all possible paths from S_i. Also, under the observation that $\widehat{O_i}$ is independent of $\widehat{S_j}$ given $\widehat{S_i}$, $P(O^{<1,T>}|MS_N^T)$ can be expanded into the product of $P(O^{<1,t_1-1>}|MS_0^{(t_1-1)})$ and $P(\widehat{O_i}|\widehat{S_i}), \forall i = 1, \ldots, N$.

The terms under the over-brace in eq. 1 (b) can be expanded similarly (as in eq. 2) into the product of 1) the probability that S_0 starts at t_0 and ends at t_1, and 2) the probability that the state of FSN at time t_0-1 is not S_0. The terms under the over-brace in eq. 2 indicate the cases where the sequence of event states in the FSA breaks down, which are shown by the arrows back to S_0 in figure 2.

$$\begin{aligned} &P(MS_0^{(t_1-1)})P(O^{<1,t_1-1>}|MS_0^{(t_1-1)}) \\ &\quad = \sum_{\forall(t_0<t_1-1)} P(d_0 = t_1 - t_0)P(\widehat{O_0}|\widehat{S_0}) \\ &\qquad \overbrace{\sum_{\forall(i\neq 0)} a_{0,i}P(MS_i^{t_0-1}, O^{<1,t_0-1>})} \end{aligned} \tag{2}$$

The derivation in equations 1 (b) and 2 requires the computation of $P(d_j)P(\widehat{O_j}|\widehat{S_j})$. Under the observation that $O^{t'}$ is independent of $S_j^{t''}$ given $S_j^{t'}$ (where $t' \neq t''$), this can be derived from the Bayesian probabilities as:

$$\begin{aligned} P(d_j)P(\widehat{O_j}|\widehat{S_j}) &= P(d_j) \prod_{t_j \leq t' \leq t_{j+1}-1} P(O^{t'}|S_j^{t'}) \quad \cdots \text{(a)} \\ &= \beta_{(t_j,t_{j+1}-1)} \overbrace{P(d_j) \prod_{t_j \leq t' \leq t_{j+1}-1} P(S_j^{t'}|O^{t'})} \quad \cdots \text{(b)} \\ &= \beta_{(t_j,t_{j+1}-1)} \text{Bel}(\widehat{S_j}) \quad \cdots \text{(c)}, \end{aligned} \tag{3}$$

where $\prod_{t_m \leq t' \leq t_{m+1}-1} \frac{P(O^{t'})}{P(S_j^{t'})}$ is factored out and written as $\beta_{(t_j,t_{j+1}-1)}$. For compactness, we write the terms under the over-brace as $\text{Bel}(S_j^{<t_j,t_{j+1}-1>})$.

We can now substitute equations 2 and 3 (c) into eq. 1 (b) and get eq. 4, where $\beta_{(t_j,t_{j+1}-1)}, \forall j$, are combined and factored out as $\beta = \beta_{<1,T>}$ under the assumption that the *a priori* probabilities of all primitive events are equal (i.e., $P(S_i) = P(S_j), \forall i \neq j$). We let $\overline{P(MS_i^t|O^{<1,t>})}$ be $P(MS_i^t, O^{<1,t>})$ after the factorization.

When we compare $P({}^mMS_q^T|O^{<1,T>})$ and $P({}^mMS_r^T|O^{<1,T>})$ of the event model ${}^mMS$, or the probabilities of any two composite-events $P({}^mMS_M^T|O^{<1,T>})$ and $P({}^nMS_N^T|O^{<1,T>})$, α_0 and $\beta_{<1,T>}$ will be canceled out. Therefore, we can normalize $P(MS_N^T|O^{<1,T>})$ with $\alpha_0\beta_{<1,T>}$ and get $\overline{P(MS_N^T|O^{<1,T>})}$. In the following, the term $P(MS_N^T|O^{<1,T>})$ is used interchangeably with $\overline{P(MS_N^T|O^{<1,T>})}$. We note that the derivation of eq. 4 is similar to that of HMM, except that 1) we do not make the first-order Markov assumption, and 2) instead of the maximal likelihood, our derivation is based on the maximum posteriori probability framework, into which BNs can be integrated naturally.

Recursive Computation of $\overline{P(MS_N^T|O^{<1,T>})}$

The direct computation of $\overline{P(MS_N^T|O^{<1,T>})}$ at time $t = T$ involves at least an operation of $O(T^N)$ complexity since there are T^N combination of the values of $t_1, t_2, ..., t_N$. In the case of un-segmented videos, the computation becomes more intensive since we also need to determine t_0 and all prior events. A more efficient recursive algorithm similar to the computation of the *forward probabilities* in the conventional HMMs [9] can be derived as follows. First, the terms related to state S_N are factored out to the left of $\sum_{\forall(t_1,t_2,\ldots,t_N)}$ as in eq. 5 (a). The terms after $a_{N,N-1}$ are now equivalent to $\overline{P(MS_{N-1}^{t_N-1}|O^{<1,t_N-1>})}$, which is readily available at T as it has been computed earlier at $t_N - 1 < T$. In eq. 5 (c), we write the terms after $\sum_{t_N \leq T}$ as $P(MS_{N_{t_N}}^T|O^{<1,T>})$, which is the probability of MS_N at time T, where the last state S_N starts from t_N. The complexity is now reduced to $O(NT^2)$, since, at frame t, we need to explore only t possible transitions (t_i) to $S_i, \forall i = 1, \ldots, N$, which requires only a simple update of $\text{Bel}(\widehat{S_i})$

3.2 Modeling $P(d_i)$

One way to derive $P(d_i)$ required for the computation of $\text{Bel}(\widehat{S_i})$ in eq. 3 (c) is:

$$\begin{aligned} P(d_i = d) = (1 - P(S_i^{t_i+d+1}|S_i^{<t_i,t_i+d>})) \\ P(S_i^{t_i+d}|S_i^{<t_i,t_i+d-1>}) \ldots P(S_i^{t_i+1}|S_i^{t_i}) \end{aligned} \tag{6}$$

Learning the distribution functions modeled by eq. 6 is difficult due to its high dimensionality. We can simplify eq. 6 by making the first-order Markov assumption that $P(S_i^{t_j}|S_i^{<t_i,t_j-1>}) = P(S_i^{t_j}|S_i^{t_j-1})$ and get:

$$P(d_i = d) = P(S_i^{t'}|S_i^{t'-1})^{(d-1)}(1 - P(S_i^{t'}|S_i^{t'-1})) \tag{7}$$

Now, only one parameter needs to be learned. However, $P(d_i)$ is now modeled by an exponential function, which is inappropriate for many large-scale events. For example, the probability that a person walks towards an object should not decrease exponentially with the number of frames in many real events. To overcome this difficulty, we parameterize $P(d_i)$. For specific tasks that follow a specific pattern, the parameters may be estimated directly from training data. In

$$P(MS_N^T|O^{<1,T>}) = \alpha_0\beta_{<1,T>} \sum_{\forall(t_1,t_2,\dots,t_N)} \left[\sum_{\forall(t_0<t_1-1)} \text{Bel}(\widehat{S_0}) \sum_{\forall(i\neq 0)} a_{0,i}\overline{P(MS_i^{t_0-1}|O^{<1,t_0-1>})}\right] a_{1,0}\text{Bel}(\widehat{S_1})a_{2,1}\text{Bel}(\widehat{S_2})\dots a_{N,N-1}\text{Bel}(S_N^{<t_N,T>}) \quad (4)$$

$$\begin{aligned}\overline{P(MS_N^T|O^{<1,T>})} &= \sum_{t_N\leq T}\text{Bel}(S_N^{<t_N,T>})a_{N,N-1}\sum_{\forall(t_1,t_2,\dots,t_{N-1})}[\dots]\,a_{1,0}\text{Bel}(\widehat{S_1})\dots a_{N-1,N-2}\text{Bel}(\widehat{S_{N-1}})\cdots\text{(a)}\\ &= \sum_{t_N\leq T}\text{Bel}(S_N^{<t_N,T>})a_{N,N-1}\overline{P(MS_{N-1}^{t_N-1}|O^{<1,t_N-1>})} \qquad \cdots\text{(b)}\\ &= \sum_{t_N\leq T}\overline{P(MS_{N_{t_N}}^T|O^{<1,T>})} \qquad \cdots\text{(c)}\end{aligned} \quad (5)$$

some situations the durations are relatively constant; we model these as Gaussian distributions and estimate their mean and variance. In other situations, the durations may be highly variable depending on the scene context and the execution styles of the actor. In such cases, we assume that all possible execution styles are equally likely and model $P(d_i)$ as a uniform distribution over a certain frame range. We restrain short durations (*e.g.*, 1 to 10 frames) by a sigmoid function to avoid unlikely event segments that can be caused by noise.

3.3 Segmenting Composite Events

By comparing $P(MS_i^T|O^{<1,T>}), \forall i$, we can make a decision about the most likely state in the FSA at time T. However, to segment the event MS from a continuous video stream, we need to know where the start of the event sequence is and when the transitions between event states have taken places. To keep track of the most likely start time of MS, we compute $P(MS_i^{*T}|O^{<1,T>})$ which is defined as *the probability of the most likely event sequence* $S_1,\dots,S_i$ *occurs at* T *given* $O = O^{<1,T>}$. $P(MS_i^{*T}|O^{<1,T>})$ can be computed using a similar equation to that of $P(MS_i^T|O^{<1,T>})$ (eq. 1 (b)), but replacing the summation with a max operator over $t_1,t_2,...,t_i$. By following the same derivation as that for $P(MS_i^T|O^{<1,T>})$ (i.e. equations 1(b) through 5 (c)), we can compute $P(MS_i^{*t}|O^{<1,t>}), \forall S_i$, as:

$$P(MS_i^{*t}|O^{<1,t>}) = \max_{t_i\leq t}\text{Bel}(S_i^{<t_i,t>})a_{i,i-1}P(MS_{i-1}^{*(t_i-1)}|O^{<1,t_i-1>}) \quad (8)$$

The most likely transition to S_i (or $t_{i_{best}}$) can be computed by replacing the max operator with *argmax*. As with the case of $P(MS_N^t|O^{<1,t>})$, eq. 8 are processed at each frame, starting from $i = 1$ to N.

The end of an event segment of MS can be detected in a video stream by setting a probability threshold (τ_e). Whenever $P(MS_N^t|O^{<1,t>})$ falls below τ_e, we mark that frame as the end of the current event segment of MS. There are several ways to define the start of an event segment. The most naive way is to backtrack the most likely path $t_{N_{best}}, t_{N-1_{best}}, \dots, t_{1_{best}}$ from the current frame. Another way is to find the time frame $t = t_{\text{peak}}$ with the highest the probability of $P(MS_N^t|O^{<1,t>})$ during the ends of the current segment and the previous one, and backtracking the best path at t_{peak}.

3.4 Event Recognition Algorithm

The computation of $P(MS_i^t|O^{<1,t>})$ using eq. 5(c) (replacing N with i) can be illustrated as follows. The structure of an FSA state S_i is shown in figure 3, consisting of the following three parts: 1) a list of substates $S_{i,d}$, 2) $P(d_i)$, and 3) a list of four-tuples shown in the boxes at the bottom. $S_{i,d}$ is generated to evaluate $P(MS_{i_{t_i}}^t|O^{<1,t>})$ (where $t - t_i = d$) in eq. 5(c). So, if there are k possible start time t_i of event S_i, a total of k sub-states will be generated and maintained. The structure of $S_{i,d}$ contains four parameters which are shown inside the box with the arrow. The summation of $P(MS_{i_{t_i}}^t|O^{<1,t>})$ computed by all sub-states $S_{i,d}$ is used as an update for $P(MS_i^t|O^{<1,t>})$ in the four-tuple. At $t = 0$, $P(S_0^0|O^0)$ and $P(MS_0^0|O^0)$ are initialized to 1. For all other S_i, $P(S_i^0|O^0)$ and $P(MS_i^0|O^0)$ are initialized to 0. At time t, starting from S_1 to S_N, the following steps are performed to update the list of substates $S_{i,d}$ and compute $P(MS_i^t|O^{<1,t>})$ in the four-tuple.

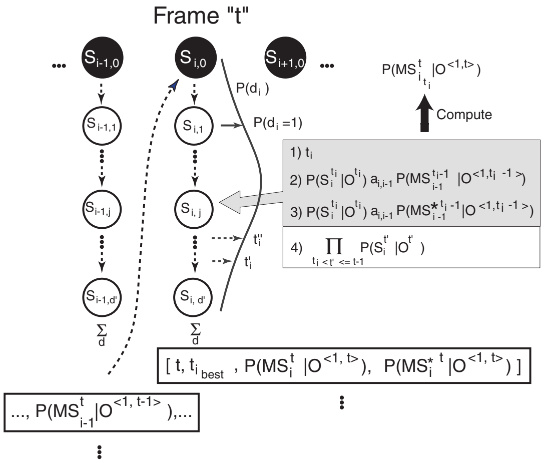

Frame "t"
...
$S_{i-1,0}$
$S_{i,0}$
$S_{i+1,0}$
...
$P(MS^{t}_{i_{t_i}}|O^{<1,t>})$
Compute
$P(d_i)$
$P(d_i=1)$
$S_{i-1,1}$
$S_{i,1}$
$S_{i-1,j}$
$S_{i,j}$
$S_{i-1,d'}$
$S_{i,d'}$
t''_i
t'_i
1) t_i
2) $P(S_i^{t_i}|O^{t_i})\, a_{i,i-1}\, P(MS^{t_i-1}_{i-1}|O^{<1,t_i-1>})$
3) $P(S_i^{t_i}|O^{t_i})\, a_{i,i-1}\, P(MS^{*\,t_i-1}_{i-1}|O^{<1,t_i-1>})$
4) $\prod_{t_i<t'<=t-1} P(S_i^{t'}|O^{t'})$
$\sum_d$
$\sum_d$
$[\, t, t_{i_{best}}, P(MS_i^t|O^{<1,t>}), P(MS_i^{*\,t}|O^{<1,t>})\,]$
$..., P(MS^t_{i-1}|O^{<1,t-1>}),...$

Figure 3: *Processing steps performed on state S_i.*

1. If the list of $S_{i,d}$ is empty, go to step 2 directly. Otherwise, for all $S_{i,d}$, increment d by one frame, i.e. $S_{i,d} \rightarrow S_{i,d+1}$. Parameters 1) to 3) of $S_{i,d}$ are kept unchanged. The fourth parameter is updated by multiplying it with $P(S_i^t|O^t)$ derived from the BN of S_i. For $S_{0,d}$, we multiply it with $1 - P(S_1^t|O^t)$ as S_0 is not represented by a BN.

2. Create a new sub-state $S_{i,1}$ and add it to the list of $S_{i,d}$. The structure of $S_{i,1}$ is initialized as follows. 1) Set t_i to t. 2) For the second parameter, multiply $P(MS_{i-1}^{t-1}|O^{<1,t-1>})$ obtained from S_{i-1} (upward dotted arrow) with $a_{i,i-1}$ and $P(S_i^t|O^t)$. 3) The third parameter is updated similarly but replacing $P(MS_{i-1}^{t-1}|O^{<1,t-1>})$ with $P(MS{*}_{i-1}^{t-1}|O^{<1,t-1>})$. 4) Initialize the fourth parameter to 1.

3. Create $(t, t_{i_{\rm best}}, P(MS_i^t|O^{<1,t>}), P(MS{*}_i^t|O^{<1,t>}))$, and add it to the list of four-tuples. $P(MS_i^t|O^{<1,t>})$ is computed, using equations 5 (b) and (c), as the summation of $P(MS^t_{i_{t_i}}|O^{<1,t>})$, derived at each sub-state $S_{i,d}$ by taking the product of parameters 2), 4) and $P(d_i = t - t_i)$. $P(MS{*}_i^t|O^{<1,t>})$ and $t_{i_{\rm best}}$ are computed similarly based on equations 8.

Complexity Reduction and Normalization

The earlier 3-step computation is bound by $O(NT^2)$. We can further reduce the complexity by eliminate the unlikely sub-states $S_{i,d}$ or discard the unlikely t_i (e.g., when $P(MS^t_{i_{t_i}}|O^{<1,t>})$ is less than a threshold, τ_{MS}). The issue here is to make certain that the discarded t_i will not become more likely again.

Suppose t'_i and t''_i are two candidates of t_i, where $t'_i < t''_i$, with the corresponding $S_{i,d'=t-t'_i}$ and $S_{i,d''=t-t''_i}$ (see figure 3). In the case that $P(d_i)$ is a uniform distribution, $S_{i,d'}$ can be safely discarded when $P(MS^t_{i_{t'_i}}|O^{<1,t>}) < \tau_{MS}$ as long as C.1) $S_{i,d''}$ is still in the list, C.2) $P(MS^t_{i_{t'_i}}|O^{<1,t>}) < P(MS^t_{i_{t''_i}}|O^{<1,t>})$, and C.3) $t - t'_i$ is longer than the spread of the sigmoid function applied to inhibit short event durations. This is because $P(MS^{t+k}_{i_{t'_i}}|O^{<1,t+k>})$ will never become more likely than $P(MS^{t+k}_{i_{t''_i}}|O^{<1,t+k>})$, where $k > 0$. For example, at time frame $t + 1$, the update of parameter 4) of $S_{i,d'}$ and $S_{i,d''}$ by step 2), effectively results in the update of $P(MS^{t+1}_{i_{t'_i}}|O^{<1,t+1>})$ and $P(MS^{t+1}_{i_{t''_i}}|O^{<1,t+1>})$ by the multiplication of $P(S_i^{t+1}|O^{t+1})$:

$$P(MS^t_{i_{t'_i}}|O^{<1,t>})P(S_i^{t+1}|O^{t+1}) \leq \\ P(MS^t_{i_{t''_i}}|O^{<1,t>})P(S_i^{t+1}|O^{t+1}). \tag{9}$$

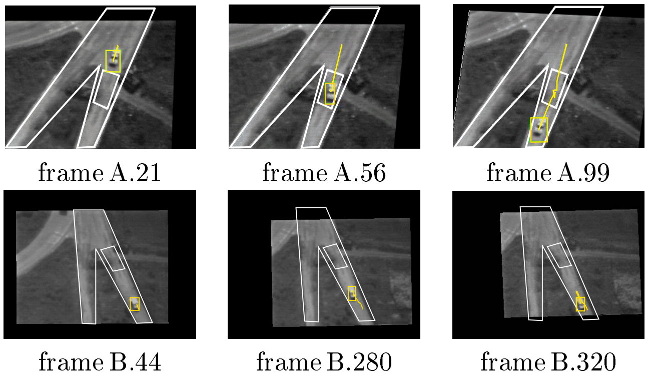
In the case that the event duration distribution is Gaussian, another condition that must be met to safely disregard $S_{i,d'}$ is that C.4) $d' = t - t'_i$ is longer than the mean of the Gaussian PDF. The reason is that, if C.4 is met, $P(d_i = t - t'_i) < P(d_i = t - t''_i)$ (and effectively the relation in eq. 9) will always hold (see figure 3).

In practice, by choosing an appropriate value of τ_{MS}, the upper bound of complexity can be set such that we need to maintain only a certain number, say k, of t_i. Alternatively, we can control the complexity by choosing an appropriate k. From our experiments, in the case of uniform $P(d_i)$, k should be longer than the spread of the sigmoid function (which is approximately 10 frames). For a Gaussian $P(d_i)$, we can choose k to be an integer approximation of the variance. In either case, k is significantly less than the length of the video, even in the case of moderately noisy video sequences. Therefore, the complexity of our event recognition algorithm is reduced to $O(NT)$.

Another practical issue is that the probabilities must be normalized periodically to prevent the underflow of the floating point variables. Our strategy is that whenever $\sum_{0 \leq i \leq N} P(MS_i^t|O^{<1,t>}) < \delta_{\rm MS}$, we normalize the fourth parameter of $S_{i,d}$ (i.e. $\prod_{t_i < t' \leq t-1} P(S_i^{t'}|O^{t'})$) of all sub-states of S_i by $\sum_{0 \leq i \leq N} P(MS_i^t|O^{<1,t>})$. That is, $P(MS_i^t|O^{<1,t>})$ of all states should sum up to 1, since the state of FSA must be one of $S_0, ..., S_N$.

4 Results

We have constructed eighteen BNs and thirteen FSAs similar to the one shown in figure 2 to represent primitive events and composite events respectively. Parameters of each network are assumed to be Gaus-



frame A.21 frame A.56 frame A.99

frame B.44 frame B.280 frame B.320

Figure 4: *Detection and tracking of moving regions for “CheckPntA”(top) and “CheckPntB”(bottom).*

sian and are estimated directly from a training data set composed of approximately 600 frames. *A priori* probabilities of all events are assumed to be equal. We tested our single-thread event recognition algorithm on videos collected in various domains (e.g. ground and airborne surveillance).

4.1 Detecting Pre-segmented Events

First, we show an example of discriminating between two similar but different events in surveillance video acquired from an Unmanned Airborne Vehicle (UAV). The images are stabilized (compensated for sensor motion) and objects are tracked. Figure 4 shows two stabilized image sequences (CheckPntA and CheckPntB) and overlaid tracking results. First video contains the event (*“a car goes through checkpoint”*; the second contains the event *“a car avoids checkpoint”*). The checkpoint is a known area, shown by the polygon in the figure. Our system also processes other events (e.g. “follow another vehicle”) in these UAV sequences. However, the two events we select are the ones within the scope of the checkpoint context. The model of *“go through checkpoint”* is similar to that of *“avoid checkpoint”* (in figure 2) but consists of three different FSA states: *“approach the checkpoint zone”*, *“move inside”* and *“leave”*. Both videos have been pre-segmented to include only the one composite event each.

Figure 5 (a) shows the probabilities of four primitive event models related to the checkpoint in “CheckPntA”. The plots (b) and (c) show the plots of normalized $P(MS_i^t|O^{<1,t>}), \forall S_i$ (from the four-tuples) of the semi-HMM models of *“go through”* and *“avoid”* respectively. *“Go through checkpoint”*(the solid line in (b)) is recognized at frame 99 ($P(MS_3^{t=99}|O^{<1,99>})$ is 0.96), while the state of the semi-HMM of *“avoid checkpoint”* remains in the initial state (in (c)). For comparison, the plots in figure 6 (a) and (b) show $P(MS_i^t|O^{<1,t>})$ of a standard HMM event model applied to “CheckPntA”. We can see that the HMM event model also detects *“go through checkpoint”* (shown by the solid line); however, an oscillation between S_0 and S_1 occurs temporarily in the HMM result of *“avoid checkpoint”*. This is because the exponential event duration model encourages quick transitions to S_1 at around frame 35, even though the direct probability of S_1 is very low.

For “CheckPntB” sequence, the probabilities of primitive events contain peak noise due to motion stabilization errors (stopped cars appear to be moving). Nevertheless, *“avoid checkpoint”* is detected at frame 285, where $P(MS_3^{t=285}|O^{<1,285>})$ is 0.99, and *“go through checkpoint”* is not recognized as the state of semi-HMM model remains in either S_0 or the *“approach”* state. Standard HMM model confuses noise for real events (graphs not shown due to lack of space).

4.2 Segmenting Events

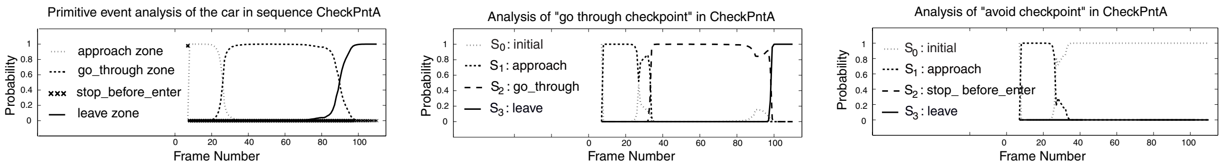
In general, we need to detect and segment events from continuous video streams. Our methods allows for this as described earlier. We show results on a synthesized sequence, called “CheckPntC” constructed by concatenating the two real sequences “CheckPntB” and “CheckPntA” shown earlier. The goal is to examine whether the event segmentation from different event models coincides with one another. Ideally, the start of one model should minimally overlaps with the end of another, and the cross-over point should be close to (i.e. minimal delay) the cross-over of Bayesian probabilities of primitive events.

Results shown in figure 7(a) indicate the pattern of *“avoid checkpoint”* being followed by *“go through”*. In figure 7(b), by backtracking from the last state of the *“go through checkpoint”* semi-HMM at frame 173, we infer that this event started at frame 72. Similarly, *“avoid the checkpoint”* is detected to end at frame 71 and begin at frame 7. This segmentation agrees with the construction of the video.

4.3 A More Challenging Example

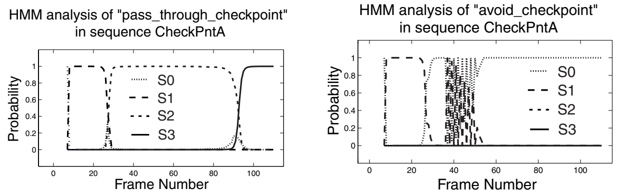
Figure 8 shows the object tracking results of *“theft at phonebooth”* sequence, where we observe that a person (obj1) drags a suitcase (obj2) to a phonebooth, then another person (obj4) comes and takes the suitcase, while the owner is using the phone. Due to the low camera angle, the ground trajectories of objects in this video can be very noisy (figure 8 (b)). For example, the track of obj4 is very noisy compared to that of obj3 because obj4 is further away from the camera than obj3. A few-pixel tracking error is projected to over a meter on the ground. We model a few actions common to such a scene (e.g. *“bring in object”*, *“attacking a person”*, *“use phone”*). Figure 8 (c) shows

0-7695-1950-4/03 $17.00 © 2003 IEEE



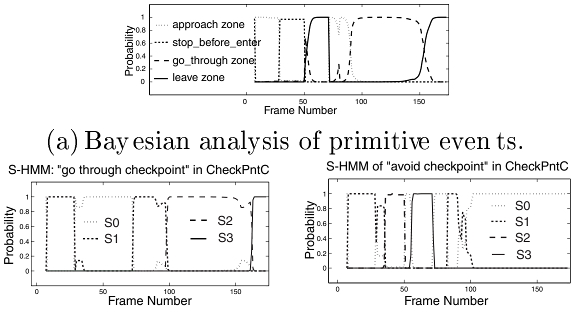
(a) Analysis of primitive events. (b) “Go through checkpoint”. (c) “Avoid checkpoint”.

Figure 5: *Event Analysis Results of “CheckPntA” sequence.*



(a)“go through” (b)“a void heckpoint”

Figure 6: *Plots of $P(MS_i^t|O^{<1,t>})$ $\forall S_i$ of the HMM models of “go through checkpoint” and “avoid che ckpoint” as they are applied on “CheckPntA”.*



(a) Bay esian analysis of primitive even ts.

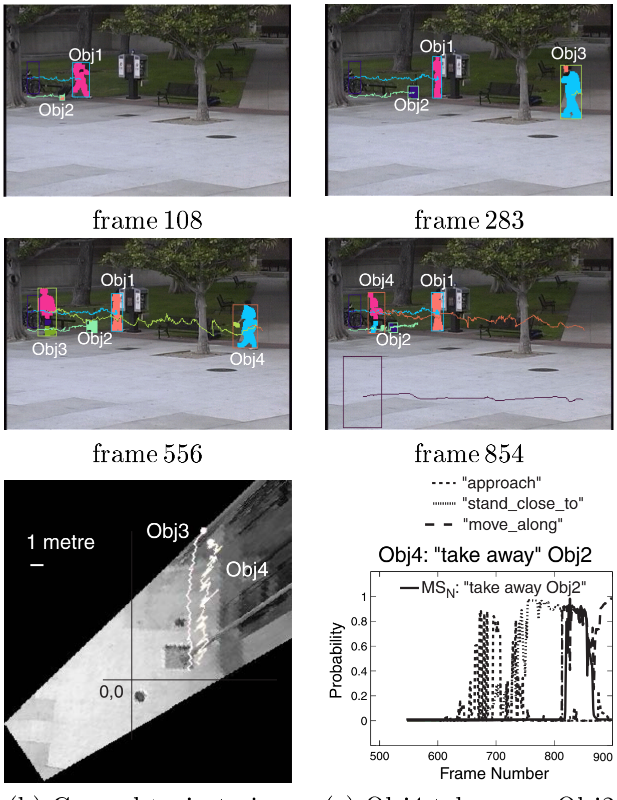
(b) (c)

Figure 7: *Se gmenting two events in the simulated sequence “CheckPntC”. Plots of $P(MS_i^t|O^{<1,t>}), \forall S_i$, for S-HMM models of “go through” and “avoid” are shown in (b) and (c) respectively.*

the detection of *“obj4 take away obj2”*. The probabilities of primitive events are shown by dotted lines and $P(MS_N^t|0^{<1,t>})$ is shown by the solid line. We notice the Bay esian probabilities of *“approach”* are v ery noisy due to the ground trajectory projection errors. The ev ent *“take away the object”* is still detected correctly at frame 826 where $P(MS_3^{826}|0^{<1,826>}) = 0.925097$. We also compare $P(^jMS_N^t|0^{<1,t>})$ of other competing ev ents(^jMS) that obj4 did not perform suc h as *“bring in the object”*, and find them to be much low er. We also correctly detect and segment all other even ts by other actors, and recognize the multi-agent global scenario *“theft at phonebooth”* which relies on accurate temporal segmentation of these events.

4.4 Performance Evaluation

We have processed other real videos con taining complex events (e.g., *“exchange an obje ct”*, *“attack and chase”*) and achiev ed the detection rate of 96.7% on



(b) Ground trajectories. (c) Obj4 take aw ay Obj2

Figure 8: *A nalysis of “Theft at PhoneBooth”.*

discriminating predefined competing composite events in volving 30 objects. How ev er, a complete ev aluation of the system performance requires very extensive data collection. Complex ev ents, such as *“theft at phonebooth”* are rare in natural observation and it is hard to an ticipate and stage all the variations that may occur and affect the results. Instead, we have experimented with simulated perturbations of trajectories extracted from some real events. In one example, w e generated 40 simulated object trajectories for two competing composite human events: *“pass by”* (S_1 :approach a person, S_2:move near and S_3:leave) and *“make contact”* (S_1 :approach, S_2:stop at and S_3:turn around and leave). Object tracks were corrupted with Gaussian noise with zero mean and the variance of the human w alking speed (6.68 cm/frame). Such Gaussian noise causes the probability of primitive even ts to be noisy. We have characterized the performance of our methods

Sequence	Frames	pe/ce/mt/ctx	fps.
CheckPntA (2)	109	38/3/0/1	43.6
CheckPntB (3)	292	38/3/0/1	16.22
ObjExchange (3)	640	83/11/3/1	0.71

Table 1: *Computation Time of Video Sequences.*

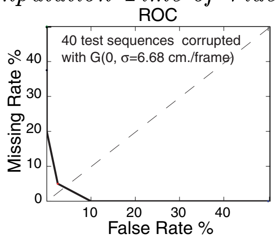


Figure 9: *ROC curves of data with various noise levels.*

by ROC curves: curves showing the trade-off between missing and false alarm rates at varying probability threshold values, as shown in figure 9. We notice that our system still maintains both the missing and the false alarm rates below 5%, indicating that our composite event recognition algorithm avoids inferring erroneous short event durations that may be caused by the noisy primitive events.

Table 1 summarizes the computation time (excluding motion detection and tracking) to process sequences using a PII-333MHz machine with 128 MB of RAM (about 1/8 of today'sprocessing power). The computation time depends on various free parameters such as the number of moving objects (in the parentheses), scene contexts and the events in the library that are of interest: *pe*, *ce*, *mt* and *ctx* are short for the number of primitive events, composite events, multi-thread events and contexts respectively. *"CheckPntA"* and *"CheckPntB"* are relatively fast to process. This is because a large number of events related to human actions do not apply. As for *"object exchange"* (defined similarly to *"theft at phonebooth"*, the number of composite and multi-thread events increases to 11 and 3. Many of these events are defined with regard to other moving objects (as reference objects) which are unbound parameters, causing an increase in the computation time (frame rate has dropped to 0.71). For example, if there are three objects in the video, there will be six possible combinations of (actor, reference) pairs for each event to be analyzed. In the cases where the number of moving objects are high (a crowd of people), some pruning of the *(actor, reference)* pairs may be necessary.

5 Conclusion

We have presented a new event modeling and recognition method using modified semi-HMMs integrated with Bayesian networks. The transparent nature of the representation permits direct estimation of parameters from training data. We have described an efficient algorithm to make inferences with these models and shown their effectiveness in presence of considerable noise in some examples. We have also experimented with a large number of simulated, noisy trajectories but are unable to include those results in the paper for lack of space. While many problems of event representation and recognition, including those of the lower level object detection and tracking, remain, we believe that the tools we have introduced for higher level inferences are generic and will apply to many complex tasks.

References

[1] P. Remagnino, T. Tan, K. Baker. Multi-agent visual surveillance of dynamic scenes. *Image and Vision Computing*, 16:529–532, 1998.

[2] J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive probabilistic networks with hidden variables. *Machine Learning*, 29:213–244, 1997.

[3] S. Hongeng, F. Brémond and R. Nevatia. Bayesian framework for video surveillance application. In *IEEE Procedings of ICPR*, Barcelona, Spain, 2000.

[4] S. Hongeng, F. Brémond and R. Nevatia. Representation and optimal recognition of human activities. In *IEEE Proceedings of Computer Vision and Pattern Recognition*, South Carolina, 2000.

[5] S. Hongeng and R. Nevatia. Multi-agent event recognition. In *IEEE Proceedings of the International Conference on Computer Vision*, volume 2, pages 84–91, Vancouver, Canada, July 2001.

[6] S. Intille and A. Bobick. Recognizing planned multi-person action. In *Computer Vision and Image Understanding*, volume 3, pages 414–445, March 2001.

[7] S. Levinson. Continuously variable duration hidden markov models for automatic speech recognition. *Compput. Speech Language*, 1:29–45, 1986.

[8] A. Galata, A. Cohn, D. Magee and D. Hogg. Modeling interaction using learnt qualitative spatio-temporal relations and variable length markov models. In *Proceedings of the European Conference on Artificial Intelligence*, pages 15–21, Lyon, 2002.

[9] L. R. Rabiner and B. H. Juang. *Fundamental of Speech Recognition*. Prentice Hall, 1993.

[10] N. Oliver, B. Rosario and A. Pentland. A bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, August 2000.

[11] J.M. Siskind and Q. Morris. A maximum-likelihood approach to visual event classification. In *Proceedings of the European Conference on Computer Vision*, volume 2, pages 347–360, 1996.